Journal of Machine Learning Research 27 (2026) 1-104 Submitted 9/25; Revised 7/26; Published 7/26

Unveiling the Statistical Foundations of Chain-of-Thought Prompting Methods

Xinyang Hu XINYANG.HU@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, CT 06520-8230, USA

Fengzhuo Zhang FZZHANG@U.NUS.EDU
Department of Electrical and Computer Engineering
National University of Singapore
21 Lower Kent Ridge Rd, Singapore 119077

Siyu Chen SIYU.CHEN.SC3226@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, CT 06520-8230, USA

Zhuoran Yang ZHUORAN.YANG@YALE.EDU
Department of Statistics and Data Science
Yale University
New Haven, CT 06520-8230, USA

Editor: Qiang Liu

Abstract

Chain-of-Thought (CoT) prompting and its variants have gained significant attention as effective methods for solving multi-step reasoning tasks with pretrained large language models (LLMs). However, their theoretical underpinnings remain insufficiently explored. We analyze CoT prompting from a statistical perspective, offering insights into why "pretrained LLMs + CoT prompting" performs well. Additionally, we examine the role of the transformer architecture and the inclusion of intermediate reasoning steps in enhancing performance. We introduce a multi-step latent variable model to capture the reasoning process. In this model, we show that the estimator induced by CoT prompting approximates a Bayesian estimator that solves the reasoning task by inferring the posterior distribution from examples in the prompt. We prove that the statistical error of the CoT estimator consists of (i) a prompting error, which is incurred in inferring the desired task from the prompt, and (ii) a pretraining error, which is the statistical error of the pretrained LLM. We further prove that the prompting error decreases exponentially as the number of examples in the prompt increases. For the pretrained LLM, we construct a transformer model class that explicitly approximates the target distribution and establish the generalization error under the Pac-Bayes framework.

Keywords: Large Language Models, Chain-of-Thought, In-Context Learning, Statistical Learning Theory, Bayesian Model Averaging

1. Introduction

Autoregressive Large Language Models (LLMs), based on the transformer architecture (Vaswani et al., 2017), have transformed Natural Language Processing (NLP) with their ability to understand language and follow instructions. Training involves two phases: *pre-training*, where models learn from large text corpora via unsupervised learning (Ahmad et al., 2021; Zoph et al., 2020), and *post-training*, which includes supervised fine-tuning (Wei et al., 2021) and reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022). Once trained, LLMs interact with users while their parameters remain fixed.

Human users interact with LLMs through *prompting*, where text is generated based on user-provided prompts. *Prompt engineering* (Sahoo et al., 2024) involves crafting effective prompts to elicit specific behaviors from LLMs and marks a *paradigm shift* from traditional statistical learning. In this approach, LLMs "learn" from prompts via their fixed, pre-trained parameters, without additional task-specific data.

A popular prompting heuristic is In-Context Learning (ICL) (Brown et al., 2020; Dong et al., 2022), where LLMs learn concepts from a few examples in the prompt. By providing input-output examples and asking for a new input's output, LLMs can generalize the relationship to generate desired results. This method has inspired advanced techniques for complex tasks (Wei et al., 2022; Zhou et al., 2022; Kim et al., 2022; Zhang et al., 2022b; Rubin et al., 2021; Sorensen et al., 2022; Creswell et al., 2022; Yao et al., 2023; Wang et al., 2022).

A key example is Chain-of-Thought (CoT) prompting (Wei et al., 2022), which extends ICL to multi-step reasoning by including intermediate steps in the examples. Enhanced CoT variants improve reasoning path selection through majority voting or tree search (Creswell et al., 2022; He et al., 2024; Yao et al., 2023; Wang et al., 2022).

Our Contributions

While CoT prompting has proven effective for multi-step reasoning tasks like arithmetic, commonsense, and symbolic reasoning, its theoretical basis remains unclear. This work aims to rigorously understand why "pretrained LLM + CoT prompting" excels at these problems and how the *transformer architecture* and *intermediate reasoning steps* contribute. Specifically, we seek to answer: **(a)** What statistical estimators are constructed by CoT and its variants? **(b)** What are the statistical properties of these estimators? **(c)** How does the transformer architecture enable learning these estimators? **(d)** Does CoT always outperform vanilla ICL?

To address Question **(a)**, we introduce a multi-step latent variable model that represents the data-generating process for multi-step reasoning. This model involves a sequence of $H+1$ variables $\{Z_0, \ldots, Z_H\}$ generated based on a latent variable θ^*, where H is the number of reasoning steps. Here, Z_0 and Z_H are the input and output, respectively, and $\{Z_h\}_{h=1}^{H-1}$ are intermediate steps. The parameter θ^* captures the task and is a random variable in the set Θ. Under this model, CoT prompting involves providing the LLM with n examples of such sequences and asking it to generate Z_H for a new input $Z_0 = z_0^{\texttt{test}}$. We assume the LLM was pretrained to predict the next reasoning step using data generated with θ^* from a prior distribution. We prove that CoT prompting methods produce Bayesian Model Averaging (BMA) estimators (Hoeting et al., 1999) by showing that the LLM learns a

posterior distribution over θ^* from the examples and generates Z_H by aggregating this posterior.

For Question **(b)**, we analyze the statistical error of the CoT estimator by decomposing it into (i) pretraining error and (ii) prompting error. The pretraining error arises from training the LLM to predict the next reasoning step and is bounded by approximation and generalization errors. We control the approximation error by constructing transformers that approximate the population distribution, and use a *Pac-Bayes* generalization error bound (McAllester, 1998; Alquier, 2021) to establish the pretraining error bound. The prompting error reflects the statistical error of the BMA estimator from finite examples. Combining these analyses provides a comprehensive answer to Question **(b)**.

To answer Question **(c)**, we demonstrate that the attention mechanism in the transformer architecture allows the LLM to approximate BMA effectively. This means that prompting a pretrained LLM generates an output distribution that closely matches the BMA estimator. Additionally, the transformer architecture's role in analyzing approximation error is derived from pretraining error analysis. For Question **(d)**, we focus on the case where $H = 1$, which simplifies to vanilla ICL. Our theory shows that CoT is at least as effective as vanilla ICL on average, but this advantage may not hold for specific tasks or prompts. Specifically, CoT can underperform if intermediate reasoning steps are not sufficiently informative. This is validated through experiments on a synthetic task.

In summary, this paper provides a thorough theoretical understanding of CoT and related prompting methods, offering insights that bridge the gap between theory and practice and setting the stage for further research in prompt engineering.

2. Related Works

Our work contributes to the theoretical understanding of prompting methods, with a focus on CoT prompting and its variants. It also relates to research exploring ICL and CoT from both empirical and theoretical perspectives. For additional related works, see Appendix A.

Existing Research on Understanding CoT. The vanilla CoT prompting method is proposed in Wei et al. (2022) for solving multi-step reasoning problems using LLMs. Our work aims to understand the capability and behavior of CoT. The following works provide an interpretation of CoT from both experimental and theoretical perspectives. Saparov and He (2022); Shi et al. (2022); Paul et al. (2023); Wang et al. (2023a); Tang et al. (2023); Madaan and Yazdanbakhsh (2022) offer practical insight by exploring the performance and capability of CoT reasoning empirically. On the theoretical side, Merrill and Sabharwal (2023); Feng et al. (2023); Li et al. (2023b); Prystawski et al. (2024) explore the reason behind the improvement in reasoning induced by CoT. Wu et al. (2023b); Tutunov et al. (2023); Hou et al. (2023); Wang et al. (2023b) investigate the ability demonstrated by CoT through examining the internal mechanism of the transformer architecture. The current understanding of CoT is still limited and requires further investigation. In this work, we adopt a statistical point of view to establish a refined characterization of the statistical properties of CoT in both pretraining and prompting stages.

A Bayesian Interpretation of ICL. A line of work lies in the Bayesian interpretation of the ICL paradigm (Jiang, 2023; Wang et al., 2023c; Xie et al., 2021; Wies et al., 2023; Zhang et al., 2023a; He et al., 2024). Under the Bayesian framework, Xie et al. (2021) use

Hidden Markov Model (HMM) (Rabiner and Juang, 1986) to model the token generation process and assume access to the true language distribution. However, the HMM assumption is restrictive, and the perfect pretraining assumption does not incorporate the pretraining phase into the story. To this end, Wies et al. (2023) relax these two assumptions by adopting a general i.i.d. data model and analyzing a pretrained model that well approximates the true distribution given any token sequence, which is also unrealistic. These works do not mention the relationship between transformer architecture, pretraining process, and the Bayesian interpretation of ICL.

Among these works, our work is most related to Zhang et al. (2023a) and He et al. (2024). In particular, Zhang et al. (2023a) adopt a latent variable model that generalizes the HMM model in Xie et al. (2021), and show that ICL can be explained as a BMA estimator under this model. They also establish the statistical error of the BMA estimator and connect it to the attention mechanism. He et al. (2024) further extend this BMA framework for studying LLM-based decision-making problems, where an LLM is used as a policy. They bring about the equivalence between the LLM-based policy trained by predicting the next action given the history and a Bayesian version of imitation learning. This ability of Large Language Models (LLM) allows the decision maker to take optimal actions in each timestep when the pretraining data contains the optimal actions provided by the oracle. Meanwhile, the work of Falck et al. (2024) challenges the Bayesian hypothesis of ICL by showing that commercial LLMs such as GPT 3.5 violate the martingale property on synthetic experiments, which is a prerequisite for Bayesian predictors. Intuitively, their argument is that if permuting in-context examples causes significant variations in output predictions, the model is not Bayesian. However, their findings do not refute our results. We assume the LLM is pretrained based on data generated from the latent variable model, which might not be satisfied by the commercial LLMs. We refer readers to Appendix A for a more detailed review of the empirical and theoretical work on ICL.

A Detailed Comparison With Zhang et al. (2023a). Our work extends the ideas presented in Zhang et al. (2023a) by broadening their Bayesian framework to encompass CoT prompting and its variants. While Zhang et al. (2023a) focuses on single-step in-context learning and imitation learning, our approach models the CoT process via a multi-step latent variable model, capturing the sequential generation of intermediate reasoning steps. Compared with Zhang et al. (2023a), our contributions are distinct in several key aspects. First, we analyze the predictive error for a specific test query z_0^{test}, whereas Zhang et al. (2023a) focus on the average error across all in-context samples. Second, we demonstrate that the predictive error decays exponentially with the number of in-context examples, and identify the principal factors influencing this decay by scrutinizing the structure of the task space Θ. Furthermore, compared with Zhang et al. (2023a), our more generalized framework necessitates novel proofs since the approaches in Zhang et al. (2023a) do not directly apply. Our analysis presents several technical novelties:

(i) In Section 4.3, we develop a more generalized proof than that in Zhang et al. (2023a), explicitly addressing dependencies among intermediate steps. In Section 4.2, we also handle additional out-of-distribution queries that were not considered in Zhang et al. (2023a).

(ii) The analysis in Section 5 on prompting error is entirely new. Furthermore, we extend our results to analyze advanced CoT-related methods in Section 7, an area that remains unexplored in prior work.

(iii) In Section 6.2, to analyze the pretraining error, we extend the standard PAC-Bayesian generalization analysis to accommodate our non-standard data structure. Moreover, in Section 6.3, we construct a family of transformer models that explicitly take the multi-step structure into account to bound the approximation error.

3. Background

This section introduces the background knowledge about transformer-based large language models and CoT prompting. We defer more background knowledge to Appendix B.

Pretraining Autoregressive LLMs. Most commercial LLMs, such as GPT-4 (Achiam et al., 2023), Claude (Anthropic, 2023), Llama (Touvron et al., 2023), and Gemini (Team et al., 2023), are autoregressive. These models, denoted as $\mathbb{P}_{\text{LLM}}$, predict future tokens based on previous ones (the prompt). Given a prompt $S_t = (x_1, ..., x_t) \in \mathcal{X}^t$, the model generates the next token x_{t+1} by sampling from $\mathbb{P}_{\text{LLM}}(\cdot \mid S_t)$, appends x_{t+1} to form S_{t+1}, and repeats until the sentence is complete. During pretraining, the model learns to predict the next token by maximizing the log-likelihood $\sum_{t=1}^{T} \log \mathbb{P}_{\text{LLM}}(x_t \mid S_{t-1})$, using large-scale datasets with billions to trillions of tokens from sources like Wikipedia, news, and books. The aim is to develop a model that generates coherent text and captures natural language structure.

Prompting a Pretrained LLM and In-Context learning. Users interact with LLMs by providing a "prompt". The autoregressive model maps the prompt to a probability distribution over possible outputs, and thus prompting is equivalent to sampling an output sequence, $\texttt{output} \sim \mathbb{P}_{\text{LLM}}(\cdot \mid \texttt{prompt})$.

In-context learning (ICL) enables LLMs to learn from prompts without updating their parameters (Dong et al., 2022). In basic ICL, the model receives input-output pairs, called "examples" or "demonstrations," along with a new input query. The LLM learns the pattern in these examples to generate the correct output for the query. Given T examples $\{(x_i, y_i)\}_{i=1}^{T}$ where $y_i = f_*(x_i)$, the query x_q, and the prompt $\texttt{prompt} = (x_1, y_1, \ldots, x_T, y_T, x_q)$, the LLM should produce $y_q \sim \mathbb{P}_{\text{LLM}}(\cdot \mid \texttt{prompt})$ such that $y_q = f_*(x_q)$. For example, given "grass is green, apple is red, sky is," the LLM should output "blue," learning the pattern from the examples without parameter adjustments.

Chain of Thought. Complex input-output relationships, especially in multi-step reasoning tasks, are hard for LLMs to learn from direct input-output pairs. CoT (Chain-of-Thought) is a prompting technique that addresses multi-step reasoning by including additional intermediate reasoning steps in the prompt (Wei et al., 2022). This approach breaks down complex problems into simpler subtasks, which the LLM can handle through vanilla ICL. A CoT prompt with H steps is denoted as $\texttt{prompt}_{\text{CoT}}(n) = (z^1_{0:H}, \ldots, z^n_{0:H}, z^{\text{test}}_0)$, where each $z^i_{0:H}$ includes intermediate steps $(z^i_0, \ldots, z^i_H)$ and the final input z^{test}_0. We use $s^i = z^i_{0:H}$ to denote the i-th demonstration. In this paper, H is fixed. Vanilla ICL prompts are recovered by setting $H = 1$, which omits intermediate steps. Detailed examples will be provided in Section 4.1.

For simplicity, we assume each reasoning step z belongs to a finite set $\mathcal{L} \subseteq \mathcal{X}^*$, with each element represented by an embedding vector in $\mathbb{R}^{d_Z}$. We denote the set of sequences of reasoning steps as $\mathcal{L}^*$, where $\texttt{prompt}_{\text{CoT}}(n) \in \mathcal{L}^*$.

4. A Latent Variable View of Multi-Step Reasoning

In this section, we interpret CoT prompting as a Bayesian estimator in a multi-step latent variable model. Section 4.1 introduces this model to capture the reasoning process, with a generalization to the non-i.i.d. case in Appendix C. In Section 4.2, we analyze CoT prompting from a statistical viewpoint. Section 4.3 shows that CoT prompting is equivalent to a BMA estimator, addressing Question **(a)**, and that the softmax attention mechanism parameterizes the BMA algorithm, partly answering Question **(c)**.

4.1 A Multi-Step Latent Variable Model

We introduce a multi-step latent variable model to capture the multi-step reasoning process of CoT, which serves as the data-generating model for studying CoT.

CoT Prompting Paradigm. We define the CoT prompt $\texttt{prompt}_{\text{CoT}}(n) = (\Upsilon_n, z_0^{\text{test}})$, where $\Upsilon_n = \{z_{0:H}^i\}_{i=1}^n$ represents n demonstration examples and z_0^{test} is the test query. To generate this prompt, we first specify a latent concept vector $\theta^* \in \Theta$, where Θ is the set of all latent concepts. θ^* represents the task (e.g., color description, math calculation) and defines the task-specific joint distribution $\mathbb{P}(\cdot \,|\, \theta^*)$ of the examples and test query in the prompt. The LLM recursively generates intermediate steps $(z_1^{\text{test}}, \ldots, z_{H-1}^{\text{test}})$ and final output z_H^{test} using

$$z_{h+1}^{\text{test}} \sim \mathbb{P}_{\text{LLM}}(\cdot \,|\, \texttt{prompt}_{\text{CoT}}(n), z_1^{\text{test}}, \ldots, z_h^{\text{test}}), \qquad \forall h \in [H-1]. \tag{1}$$

To evaluate CoT performance, we compare the generated distribution of z_H^{test} in (1) with the ground truth distribution $\mathbb{P}(z_H^{\text{test}} \mid \texttt{prompt}_{\text{CoT}}(n), \theta^*)$, which represents the true task-specific distribution of the final answer based on the prompt. We illustrate the CoT paradigm with a concrete example as follows.

Consider the task $\theta^* =$ "`calculate twice the area code of the given country.`" In the CoT prompt (with $n = 2$ and $H = 2$), the first example has input $z_0^1 =$"`The US = ?`", intermediate step $z_1^1 =$"`The US has area code 1`", and final output $z_2^1 =$"`so the answer is 2`". The query $z_0^{\text{test}} =$"`Japan = ?`" requires the answer "162." When tested, ChatGPT correctly outputs "`Japan has area code 81, so the answer is 162.`" In contrast, a vanilla ICL prompt provides only inputs and final outputs, but ChatGPT[1] fails to find the right answer here. This example shows how CoT prompts improve LLM accuracy over vanilla ICL. See Figure 1 for a visual illustration of CoT and vanilla ICL prompts.

The Multi-Step Latent Variable Model. To analyze CoT from a statistical perspective, we need to specify the pre-mentioned task-specific distribution $\mathbb{P}(\cdot \,|\, \theta^*)$, which serves as the data-generating distribution for the CoT prompt. We assume that the concept θ^* is a random variable sampled from a prior $\pi \in \mathcal{P}(\Theta)$ and the examples $\{s^i\}_{i=1}^n$ are i.i.d. sequences conditioning on $\theta^* \in \Theta$. For any $\theta \in \Theta$, when $\theta^* = \theta$, within the reasoning chain

1. Both the CoT and vanilla ICL prompts are tested on ChatGPT (GPT-3.5-turbo-16k) with the temperature set to zero. See Section K.1 for the details.

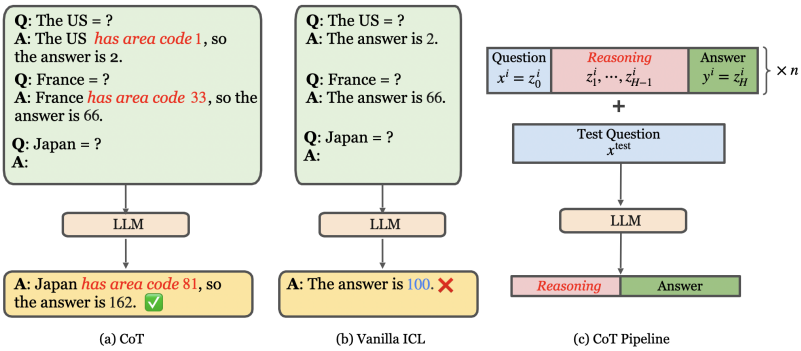


Figure 1: An illustration of CoT and vanilla ICL. Figure (a) shows the CoT prompt and the corresponding output of ChatGPT (GPT-3.5-turbo-16k). The intermediate reasoning is shown in red. The output of ChatGPT follows the pattern in the prompt, which consists of a reasoning step, followed by the desired answer. Figure (b) shows the result of using the corresponding vanilla ICL prompt, which includes of two input-output pairs. In this case, ChatGPT fails to provide the correct answer. Figure (c) illustrates a general pipeline of CoT prompting with n demonstration examples. Each example includes an input question, $H-1$ intermediate reasoning steps, and the final answer.

$i \in [n]$, we sample $z^i_{0:H}$ according to the following stochastic dynamical system with joint distribution $\mathbb{P}(s^i \,|\, \theta^* = \theta)$ given by

$$\mathbb{P}(s^i \,|\, \theta^* = \theta): \qquad z^i_0 = f_\theta\big(\zeta^i\big), \quad z^i_h = F_\theta\big(z^i_0, \cdots, z^i_{h-1}, \epsilon^i_h\big), \quad \forall 1 \le h \le H. \tag{2}$$

Here $\{\zeta^i, \{\epsilon^i_h\}_{h\in[H]}\}_{i\in[n]}$ are i.i.d. noise variables, and f_θ and F_θ are two functions parameterized by $\theta \in \Theta$. The same is true for the test sample $z^{\text{test}}_{0:H}$ and this distribution will serve as the target distribution for LLM to learn in context during the prompting stage. Specifically, f_θ generates the first query z^i_0 based on the task $\theta^* = \theta$, and F_θ models the evolution of the "reasoning process" $\{z^i_h\}_{h\in[H]}$. Specifically, each z^i_h depends on all of the previous reasoning steps as well as the latent variable θ^*. The rationale behind this model is that the generation of these reasoning steps is autoregressive and the distribution of the whole sequence is specific to the task θ^*. The random variables $\{\epsilon^i_h\}_{h\in[H]}$ allow the reasoning process to be stochastic. See Figure 2 for an illustration of this model.

Finally, note that setting $H = 1$, we obtain a latent variable model for vanilla ICL, which is studied in Wang et al. (2023c). Furthermore, our general model can be made more concrete by defining θ^* as a sequence of latent variables $\{\theta^*_h\}_{h\in[H]}$ characterizing the distribution of $z^i_{0:H}$, where the latent variables also have an autoregressive structure. Such a model is studied in Jiang (2023). A limitation of our model in (2) is that the n demonstration examples $\{s^i\}_{i=1}^n$ in $\texttt{prompt}_{\text{CoT}}(n)$ are assumed to be i.i.d. In practice, the demonstration examples might be composed in a dependent manner which is beyond the i.i.d. assumption. We will introduce a more general model in Appendix C, which (i) includes latent variables

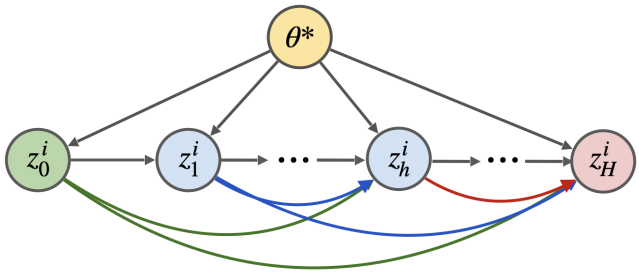


Figure 2: An illustration of the multi-step latent-variable model defined in (2). According to this graphical model, for any $h \geq 1$, each step z_h^i of i-th example depends on the previous steps $\{z_\ell^i\}_{\ell<h}$ and the hidden concept θ^*.

for each reasoning process that are governed by a latent dynamical system, and (ii) allows the generation of the demonstration examples to be dependent.

4.2 Pretrained LLM + CoT Prompting

The previous section proposes a latent variable model that captures the multi-step reasoning process of CoT. Based on this model, we will formulate the estimator constructed by CoT prompting on a pretrained autoregressive LLM from a statistical perspective.

Pretraining LLM. We assume the LLM is pretrained on data generated according to the model in (2). Specifically, the model is trained on N documents, each corresponding to a task-specific concept $\theta_\ell^* \overset{\text{i.i.d}}{\sim} \pi$ for each $\ell \in [N]$. Within each document, there are T examples $\{s^{t,\ell}\}_{t=1}^T$, generated independently based on the same task θ_ℓ^*. Thus, the training dataset consists of NT examples covering diverse tasks. We define $\{\mathbb{P}_\rho \mid \rho \in \mathcal{P}_{\text{LLM}}\}$ as the family of conditional distributions induced by the LLM, where $\rho \in \mathcal{P}_{\text{LLM}}$ represents the model parameters. Pretraining the autoregressive LLM involves finding the maximum likelihood estimator $\widehat{\rho}$, which is given by

$$\widehat{\rho} = \operatorname*{argmin}_{\rho \in \mathcal{P}_{\text{LLM}}} -\frac{1}{NT(H+1)} \sum_{\ell \in [N], t \in [T]} \sum_{h=0}^{H} \log \mathbb{P}_\rho\big(z_h^{t,\ell} \mid \Upsilon_{t-1,\ell}, \{z_j^{t,\ell}\}_{j=0}^{h-1}\big), \tag{3}$$

where $\Upsilon_{t,\ell} = \{s^{k,\ell}\}_{k \in [t]}$ represents the first t examples in the ℓ-th document. Once $\widehat{\rho}$ is found, we denote the distribution induced by the pretrained LLM as $\mathbb{P}_{\text{LLM}} = \mathbb{P}_{\widehat{\rho}}$.

CoT Prompting as an Estimator. After pretraining, we fix the LLM parameters to $\widehat{\rho}$ and prompt the model with a CoT prompt $\texttt{prompt}_{\text{CoT}}(n) = (\Upsilon_n, z_0^{\text{test}})$. This induces a conditional distribution $\mathbb{P}_{\text{LLM}}(\cdot \mid \texttt{prompt})$. The goal is to generate the correct final answer, denoted as y^{test} (or z_H^{test}), as defined in (2). The distribution of this final answer, $\mathbb{P}_{\text{LLM}}(y^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}(n))$, is obtained by marginalizing out the intermediate steps $(z_1^{\text{test}}, \ldots, z_{H-1}^{\text{test}})$. To evaluate the accuracy of the estimator, we compute the Kullback-Leibler (KL) divergence:

$$\texttt{err}_{\text{CoT}} = \texttt{KL}\Big(\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n), \theta^*\big), \mathbb{P}_{\text{LLM}}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\Big). \tag{4}$$

Error Sources	Description
Pretraining error	Statistical error of the pretrained LLM
Prompting error	Combination of query error and in-context error
Query error	Distributional shift between testing query z_0^{test} and n prompt examples
In-context error	Statistical error of inferring θ^* based on the n prompt examples

Table 1: Summary of the three sources of errors in CoT prompting.

where $\theta^* \in \Theta$ is the fixed but unknown latent task concept. The error $\texttt{err}_{\text{CoT}}$ is a random variable of the prompt $\texttt{prompt}_{\text{CoT}}(n)$ and the learned model parameters $\widehat{\rho}$.

Error Decomposition. The statistical error $\texttt{err}_{\text{CoT}}$ in (4) has two components: **pretraining error** and **prompting error**. The pretraining error results from limited training data and vanishes as N increases, and it is unrelated to the prompting stage. The prompting error occurs when using n examples to guide the LLM, even with perfect pretraining. More examples reduce this error as they help infer the task θ^*. Success also depends on how well the examples match the test query z_0^{test}. Lemma 4 further divides the prompting error into **query error** and **in-context error**. The query error reflects the Distributional shift between the query and examples, and the in-context error quantifies the model's difficulty inferring the task from the examples.

4.3 BMA Interpretation of CoT

In the following, we show that the CoT estimator $\mathbb{P}_{\text{LLM}}\big(\cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)$ can be understood as a Bayesian model averaging (BMA) estimator for the latent variable model in (2).

Pretrained LLM + CoT $\approx$ BMA. Suppose an LLM is pretrained using (3) with data from the model (2), it is expected to closely approximate the true distribution when N and T are large. Given a perfect pretrained model, since the examples in the CoT prompt are drawn from the same distribution as the pretraining data, we replace $\mathbb{P}_\rho$ with the true distribution and marginalize out the intermediate steps, leading to the following lemma.

Lemma 1 *Let the pretraining data be generated according to the latent variable model specified in* (2)*. Consider the population counterpart of the MLE in* (3)*, i.e., we let the number of documents N goes to infinity. Suppose that the LLMs have enough capacity, i.e., $\mathbb{P} \in \{\mathbb{P}_\rho \,|\, \rho \in \mathcal{P}_{\text{LLM}}\}$, and the CoT prompt $\texttt{prompt}_{\text{CoT}}(n)$ has nonzero density under the pretraining distribution, we have*

$$\mathbb{P}_{\text{LLM}}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) = \int_\Theta \mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta)\pi\big(\theta \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\mathrm{d}\theta.$$

A detailed proof of this lemma is deferred to Appendix D.1. This lemma shows that CoT prompting with a perfectly pretrained LLM effectively performs Bayesian Model Averaging (BMA). The CoT estimator is built in three steps: (i) the LLM first estimates the posterior distribution of the task θ; (ii) for each θ, the LLM predicts the final answer y^{test} based on the prompt; and (iii) the LLM then aggregates these predictions according to the posterior of θ. This BMA interpretation aligns with the results for vanilla ICL shown in Zhang et al.

(2023a), which we recover by setting $H = 1$. A detailed proof and extension to the more complex model are provided in Appendix D.1.

4.4 Attention Approximately Parameterizes BMA

We now show that the attention mechanism in the transformer architecture is able to encode the BMA algorithm for a special case of the latent variable model in (2).

A Simplified Model. Let f_{θ^*} in (2) be independent of θ^*, meaning inputs do not depend on θ^*. Assume F_{θ^*} encodes a linear model in the latent space: For each $h \in [H]$, let d_k and d_v be integers, and define feature mappings $k\colon \mathcal{L} \to \mathbb{R}^{d_k}$ and $v\colon \mathcal{L} \to \mathbb{R}^{d_v}$, with v invertible. Each reasoning step z_h^i follows

$$z_h^i = v^{-1}\big(\theta^* \phi(k_h^i) + \epsilon_h^i\big) = F_{\theta^*}\big(z_0^i, z_1^i, \cdots, z_{h-1}^i, \epsilon_h^i\big), \qquad \forall h \in [H], \tag{5}$$

where $k_h^i = (k(z_0^i), k(z_1^i), \cdots, k(z_{h-1}^i), 0, \cdots, 0)$ includes the features from the first $h-1$ steps and zeros padded to fit $\mathbb{R}^{H \cdot d_k}$. Here, $\phi\colon \mathbb{R}^{H \cdot d_k} \to \mathbb{R}^{d_\phi}$ maps k_h^i to some Euclidean space, and θ^* is a linear operator. The model implies k_h^i and $v(z_h^i)$ adhere to a kernelized linear model, with noise $\epsilon_h^i \sim \mathcal{N}(0, \sigma^2)$ independent of other variables. Our theoretical result relies only on the invertibility of v and the existence of feature maps v, k, and ϕ such that the model in (2) simplifies to (5). This model defines a linear dynamical system in the feature space and, due to the flexibility of the feature maps, encompasses a broad range of distributions.

The BMA Estimator. To analyze the BMA estimator under this model, we impose a Gaussian prior on θ^*, where each entry is i.i.d. $\mathcal{N}(0, \lambda)$ for some fixed $\lambda > 0$. Given the n examples in the CoT prompt $\texttt{prompt}_{\text{CoT}}(n)$, define $V^n = (v(z_h^i))_{h=1,i=1}^{H,n} \in \mathbb{R}^{d_v \times Hn}$ and $K^n = (k_h^i)_{h=1,i=1}^{H,n} \in \mathbb{R}^{Hd_k \times Hn}$. Let $\phi(K^n)$ be the $\mathbb{R}^{d_\phi \times Hn}$ feature matrix induced by K^n.

Given any $\theta \in \Theta$ as an estimate of θ^* and z_0^{test}, to predict $y^{\text{test}} = z_H^{\text{test}}$ according to the linear model in (5), it suffices to generate $\{v_h^{\text{test}} = v(z_h^{\text{test}})\}_{h \in [H]}$ autoregressively. Specifically, for any $h \geq 0$, conditioning on $\{z_0^{\text{test}}, \ldots, z_{h-1}^{\text{test}}\}$, the distribution of v_h^{test} is $\mathcal{N}(\theta\phi(k_h^{\text{test}}), \sigma^2 I)$ where we define k_h^{test} as $k_h^{\text{test}} = (k(z_0^{\text{test}}), k(z_1^{\text{test}}) \cdots, k(z_{h-1}^{\text{test}}), 0 \cdots, 0)$. Therefore, to get the BMA estimator, we aggregate the distribution of v_h^{test} according to the posterior distribution of θ, and return the mean value as the predictor, which is given by

$$\bar{v}_h^{\text{test}} = V^n \phi(K^n)^\top \big(\phi(K^n)\phi(K^n)^\top + \sigma^2/\lambda \cdot I\big)^{-1} \phi(k_h^{\text{test}}), \tag{6}$$

where I is the identity matrix of size $\mathbb{R}^{d_\phi \times d_\phi}$. The final BMA estimator is given by $\{v^{-1}(\bar{v}_h^{\text{test}})\}_{h \in [H]}$.

Estimator Produced by Transformer. We introduce a new autoregressive estimator using a transformer with softmax attention. This transformer, designed for a sequence of Hn vectors from n examples and a test instance, is a composition of a copy head, and a softmax attention layer. The copy head replicates previous reasoning steps within each example, outputting $(z_0^i, \ldots, z_h^i, 0, \ldots, 0)$ for any example i and step h, where we append $H - h + 1$ zeros so that the output vectors have the same dimension. The copy head, which can be constructed using standard transformer architectures (Feng et al., 2023), has also been observed in empirical studies (Olsson et al., 2022; Von Oswald et al., 2023). The

softmax attention layer focuses on relevant parts of the sequence and computes the attention output as follows. Given $\texttt{prompt}_{\text{CoT}}(n)$ and $h-1$ intermediate outputs $z_1^{\text{test}}, \ldots, z_{h-1}^{\text{test}}$, the output of the softmax attention as

$$\texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values}) = \sum_{i\in[n],\ell\in[H]} \frac{\exp(\langle q_h^{\text{test}}, k_\ell^i\rangle)\cdot v_\ell^i}{\sum_{i'\in[n],\ell'\in[H]} \exp(\langle q_h^{\text{test}}, k_{\ell'}^{i'}\rangle)}, \tag{7}$$

where we define $v_h^i = v(z_h^i)$ and the testing query as $q_h^{\text{test}} = \phi(k_h^{\text{test}})$. Finally, the output is passed through a transformation function v^{-1}, which yields

$$z_h^{\text{test}} = v^{-1}(\texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values})).$$

This newly generated z_h^{test} is then used to compute the query q_{h+1}^{test}, which is then used for generating z_{h+1}^{test}, and so on. See Figure 3 for an illustration of this transformer.

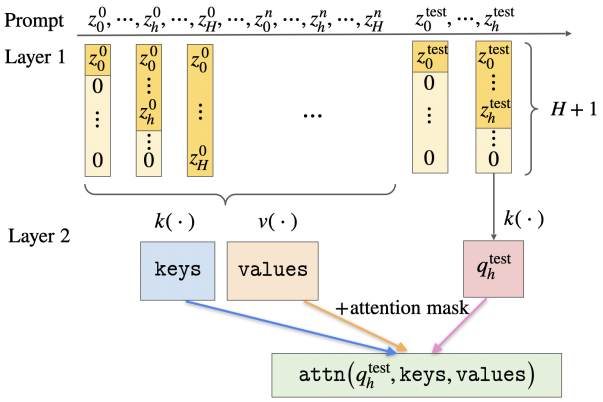


Figure 3: An illustration of how CoT is represented by a two-layer transformer. Layer 1 serves as a copy head, which copies the previous steps $\{z_j^i\}_{j=1}^{h-1}$ to the current position z_h^i. Next, the feature mappings v and k map the outputs of Layer 1 to values and keys, respectively. During the generation of z_{h+1}^{test}, the attention mechanism takes in key and value matrices $\texttt{keys}$ and $\texttt{values}$ from the demonstrations to predict the result for query q_h^{test}, where $\texttt{keys}$ and $\texttt{values}$ are formed by stacking $\{k_h^i\}_{i=1,h=1}^{n,H}$ and $\{v_h^i\}_{i=1,h=1}^{n,H}$, respectively. Note that $\texttt{keys}$ and $\texttt{values}$ do not contain keys and values computed from the generated reasoning steps $\{z_j^{\text{test}}\}_{j=1}^{h}$. We can achieve this by masking out the corresponding positions.

In the following proposition, we prove that under certain conditions, the BMA estimator coincides with the transformer output up to a scaling factor when n goes to infinity.

Proposition 2 *We assume the feature mappings k and v take bounded values and $\|v(z)\|_2 = 1$ for all input $z \in \mathcal{L}$. Besides, let ϕ in (5) be a feature map with finite dimension. Then, there exists an absolute constant C, and parameter $\lambda = n^{-2/3}$ such that for any fixed k_h^{test}, the BMA estimator in (6) and the attention output in (7) coincide as n goes to infinity up to a scaling factor C. That is, we have*

$$\lim_{n\to\infty} \max_{h\in[H]} \|\bar{v}_h^{\text{test}} - C\cdot \texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values})\|_2 = 0.$$

This proposition shows that there exists a special model satisfying (2) (the model in (5)) such that the BMA estimator of this model can be approximately implemented by a transformer. The attention output in (7) aligns with the Nadaraya–Watson Kernel regressor (Hastie et al., 2009) with an exponential kernel, while the BMA estimator in (6) is similar to ridge regression. Both estimators are consistent and converge to the same result as n increases. Detailed proof in Appendix D.2 extends results from Zhang et al. (2023a) to CoT prompting's multi-step autoregressive structure. While our feature mappings k and v are specific to reasoning steps, they can be generalized to tokens.

5. Statistical Errors of CoT Prompting

In this section, we analyze the error from the prompting stage. We first present an error decomposition and then examine vanilla CoT prompting in Section 5.1. This section addresses part of Question (b) and fully answers Question (d) from the introduction.

5.1 Statistical Errors of Vanilla CoT

We define the statistical error from CoT prompting, denoted as $\texttt{err}_{\mathrm{CoT}}$ in equation (4), which originates from both pretraining and prompting stages, as summarized in Table 1. We decompose these error sources and introduce a regularity condition for the pretrained LLM. Before proceeding, we define the partial prompt $\texttt{prompt}^h_{\mathrm{CoT}}(i)$ for any integers $i \in [0, n-1]$ and $h \in [H]$ as $\texttt{prompt}^h_{\mathrm{CoT}}(i) = \{s^j\}_{j\le i} \cup \{z_0^{i+1}, \ldots, z_{h-1}^{i+1}\}$, which includes the first i demonstration examples and the first h steps of the $(i+1)$-th example. In Section 5, we assume the prompt is drawn from the true distribution, $\texttt{prompt}_{\mathrm{CoT}}(n) \sim \mathbb{P}(\cdot \mid \theta^*)$.

Assumption 3 *We assume there exists a positive number b^* such that for any $0 \le h \le H$, $0 \le i \le n-1$, and $\texttt{prompt}^h_{\mathrm{CoT}}(i) \in \mathcal{L}^*$, we have for the data distribution $\mathbb{P}$ and the pretrained model $\mathbb{P}_{\mathrm{LLM}}$ that*

$$\sup_{z\in\mathcal{L}} \Big| \log \mathbb{P}(z_h^{i+1} = z \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(i)) - \log \mathbb{P}_{\mathrm{LLM}}(z_h^{i+1} = z \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(i)) \Big| \le b^*.$$

This assumption postulates that the true distribution $\mathbb{P}$ of the model in (2) and the distribution learned by the LLM are close. The proximity is measured in terms of the log likelihood. We will justify the existence of b^* in Section 6 under explicit assumptions on pretraining.

Lemma 4 (CoT Error Decomposition) *Under Assumption 3, the statistical error $\texttt{err}_{\mathrm{CoT}}$ in (4) can be upper bounded by the sum of a pretraining error and a prompting error, i.e.,*

$$\texttt{err}_{\mathrm{CoT}} \le \texttt{err}_{\mathrm{pre}}(\mathbb{P}, \mathbb{P}_{\widehat{\rho}}; \texttt{prompt}_{\mathrm{CoT}}(n)) + \texttt{err}_{\mathrm{prompt}}(\mathbb{P}, \theta^*, \texttt{prompt}_{\mathrm{CoT}}(n)).$$

where we define the prompting error as

$$\begin{aligned}
&\texttt{err}_{\mathrm{prompt}}(\mathbb{P}, \theta^*, \texttt{prompt}_{\mathrm{CoT}}(n)) \\
&\quad = \texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \\
&\qquad + 2\sqrt{2}Hb^* \cdot \texttt{KL}^{1/2}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big), \qquad (8)
\end{aligned}$$

and the pretraining error as

$$\begin{aligned}&\texttt{err}_{\text{pre}}(\mathbb{P}, \mathbb{P}_{\widehat{\rho}}; \texttt{prompt}_{\text{CoT}}(n)) \\ &\quad = \texttt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))\big). \end{aligned} \tag{9}$$

We provide a detailed proof of this lemma in Appendix E.1. The prompting error is defined on the distribution $\mathbb{P}$, corresponding to a perfectly pretrained LLM, making it independent of the pretraining phase. We now focus on the prompting error, assuming that z_0^{test} shares the same distribution as $\{z_0^i\}_{i\in[n]}$, implying zero query error. A distributional shift will be considered in the next section. Without a distributional shift, the test instance can be treated as the $(n+1)$-th example, since all examples are conditionally i.i.d. given θ^*. Thus, we focus on how the n prompt examples help a perfectly pretrained LLM infer θ^*, specifically addressing the in-context error.

Equivalence Class Induced by Multi-Step Reasoning. In the following, to simplify the notation, we use $X = Z_0, Z_1, \ldots, Z_H = Y$ to denote a random trajectory sampled from the model in (2). Note that the prompting error in (8) only concerns the distribution of the output Y and neglects the intermediate reasoning steps $Z_1, \ldots, Z_{H-1}$. As a result, it is possible that there exists another $\theta \in \Theta$ with the same distribution of Y. Such a relationship induces a set of equivalence classes over Θ.

Definition 5 (Equivalence Classes over Θ) *Let $\mathbb{P}(Z_0, Z_1, \cdots, Z_{H-1}, Y \,|\, \theta)$ denote the joint distribution of $Z_{0:H}$ conditioning on the latent variable $\theta^* = \theta$. We define an equivalence relation $\sim$ based on conditional density of Y given Z_0 as follows.*

$$\theta \sim \theta' \text{ if and only if } \mathbb{P}(Y = y \,|\, Z_0 = z_0, \theta) = \mathbb{P}(Y = y \,|\, Z_0 = z_0, \theta'), \quad \forall (z_0, y).$$

This relation $\sim$ induces a set of equivalence classes over Θ. In particular, for any θ, define $\Theta_{\text{eq}}(\theta) = \{\theta' \in \Theta : \mathbb{P}(y \,|\, z_0, \theta) = \mathbb{P}(y \,|\, z_0, \theta'), \forall (z_0, y)\}$ as the set of parameters equivalent to θ, i.e., the equivalence class represented by θ. Let $\widetilde{\Theta}$ denote the complete set of representatives of all disjoint *equivalent classes. Then $\Theta_{\text{eq}}(\theta) \cap \Theta_{\text{eq}}(\theta') = \emptyset$ for all $\theta, \theta' \in \widetilde{\Theta}$ and we can further write Θ as $\cup_{\theta \in \widetilde{\Theta}} \Theta_{\text{eq}}(\theta)$.*

The intuition of the equivalence relation $\sim$ is that there might be multiple reasoning paths that all lead to the correct answer. For example, Newtonian, Lagrangian, and Hamiltonian mechanics are three different approaches to classical mechanics. Their intermediate steps are different but will lead to the same answer. Based on this intuition, any parameter in $\Theta_{\text{eq}}(\theta^*)$ is equally good for predicting Y, and we only need to infer $\Theta_{\text{eq}}(\theta^*)$ from CoT prompts. We state a regularity condition for CoT prompting in terms of $\Theta_{\text{eq}}(\theta^*)$.

Assumption 6 *Given a task θ^* during CoT prompting, let $\Theta^{\complement} = \Theta \backslash \Theta_{\text{eq}}(\theta^*)$ denote the complement of the equivalence class of θ^*. We assume that there exists a strict separation between the ground truth task θ^* and any other tasks $\theta \in \Theta^{\complement}$. Specifically, there exists $\lambda > 0$ that lower bounds the Hellinger distance:*

$$\inf_{\theta \in \Theta^{\complement}} H^2\big(\mathbb{P}(Z_{0:H} = \cdot \,|\, \theta^*), \mathbb{P}(Z_{0:H} = \cdot \,|\, \theta)\big) \geq \lambda,$$

where $H^2(\cdot, \cdot)$ denotes the squared Hellinger distance. Moreover, we assume tasks in $\Theta_{\text{eq}}(\theta^*)$ are well covered by the pretraining distribution in the sense that $\pi\big(\Theta_{\text{eq}}(\theta^*)\big) > 0$.

This assumption requires the true task θ^* is λ-separated from any other θ outside of the equivalence class $\Theta_{\text{eq}}(\theta^*)$. Parameter λ serves as a margin of separation. Moreover, we assume that the prior π put considerable density on $\Theta_{\text{eq}}(\theta^*)$. This means that the task tested during prompting stage has been covered in the pretraining dataset. Based on this assumption, we establish the statistical error of the CoT estimator as follows.

Theorem 7 *Let* $\texttt{err}_{\text{pre}}$ *denote the pretraining error defined in* (9) *and assume* Θ *to be a finite and discrete set. Under Assumptions 3 and 6, with probability* $1-\delta$*, the statistical error* $\texttt{err}_{\text{CoT}}$ *defined in* (4) *satisfies*

$$\texttt{err}_{\text{CoT}} \leq \mathcal{O}\big(Hb^*\delta^{-1} \cdot \pi(\theta^*)^{-1/2} \cdot \big|\Theta^{\complement}\big| \cdot e^{-\lambda n}\big) + \texttt{err}_{\text{pre}}.$$

Here $\mathcal{O}(\cdot)$ *hides absolute constants and the probability is with respect to the randomness of the CoT prompt.*

This theorem shows that when the tasks are well separated, the prompting error converges to zero **exponentially fast** when n increases. Note that the convergence rate depends on the separation λ. A larger λ means that $\Theta_{\text{eq}}(\theta^*)$ is more distinguishable from the rest of the tasks, leading to a faster convergence rate. Besides, the error also depends on H and the size of $\Theta^{\complement}$. Intuitively, these two terms characterize how the error increases as the problem size grows. Moreover, the dependence on b^* comes from Lemma 4, which is due to replacing the pretrained LLM by the population distribution. The statistical error also depends on $\pi(\theta^*)^{-1/2}$, which means the prompting error is smaller for tasks better covered by the pretraining distribution.

Here we assume that Θ is finite. In Appendix E.2 we will further extend Theorem 7 to the more challenging case where Θ can be a continuous set and present a detailed proof.

Remark 8 (Intuition for exponential decay in n) *The exponential rate* $e^{-\lambda n}$ *follows directly from the i.i.d. product structure of the likelihood. Because the* n *demonstration trajectories* $s^1, \ldots, s^n$ *are independent given* θ^**, the joint likelihood ratio against any wrong task* $\theta \in \Theta^{\complement}$ *factorizes as* $\prod_{i=1}^{n} \mathbb{P}(s^i \mid \theta)/\mathbb{P}(s^i \mid \theta^*)$*. By the Hellinger distance identity, each factor is bounded by* $1 - \mathrm{H}^2(\mathbb{P}(\cdot \mid \theta^*), \mathbb{P}(\cdot \mid \theta)) \leq e^{-\lambda}$*, where* $\lambda > 0$ *is the separation in Assumption 6. The product of* n *such factors then decays as* $e^{-n\lambda}$*. This is the same mechanism as the Chernoff bound: exponential decay with sample size is a generic consequence of i.i.d. sampling when the per-sample distinguishability* λ *is bounded away from zero. This is empirically verified in Section 8 (Figures 10(b) and 11(b)), where the log-MSE decreases linearly in* n*, consistent with exponential decay.*

Ideally, we would like to have an upper bound that scales with $|\widetilde{\Theta}|$ because it is the actual number of all possible hypotheses when it comes to predicting Y based on X. Whereas $|\Theta^{\complement}|$ in Theorem 7 can be much larger than $|\widetilde{\Theta}|$ because it is comparable to $|\Theta|$. To have a better bound, we impose an additional assumption postulating that the distributions within each equivalence class are close.

Assumption 9 *Let* $\widetilde{\Theta}$ *be a representative set of the equivalence classes introduced in Definition 5. We assume that there exist positive numbers* α *and* α_0 *such that for all* $\theta \in \widetilde{\Theta}$ *and*

$\theta' \in \Theta_{\mathrm{eq}}(\theta)$, *we have*

$$\sup_{z_{0:H}} \left| \log \frac{\mathbb{P}(Z_{0:H} = z_{0:H} \,|\, \theta)}{\mathbb{P}(Z_{0:H} = z_{0:H} \,|\, \theta')} \right| \leq \alpha, \qquad \sup_{z_0} \left| \log \frac{\mathbb{P}(Z_0 = z_0 \,|\, \theta)}{\mathbb{P}(Z_0 = z_0 \,|\, \theta')} \right| \leq \alpha_0.$$

Moreover, we assume that $\alpha \in (0, \lambda)$, *where* λ *appears in Assumption 6.*

Assumptions 6 and 9 imply that distributions are similar within each equivalence class but disparate between equivalence classes. We establish a new upper bound as follows.

Theorem 10 *Under Assumptions 3, 6, and 9, with probability* $1 - \delta$ *over the randomness of the CoT prompt, we have*

$$\mathtt{err}_{\mathrm{CoT}} \leq \mathcal{O}\big(Hb^*\pi(\theta^*)^{-1/2}\delta^{-1}\big|\widetilde{\Theta}\big|e^{-(\lambda-\alpha)n+\alpha_0}\big) + \mathtt{err}_{\mathrm{pre}}.$$

Compared with the previous Theorem 7 that solely requires Assumption 6, we have a better dependency on the size of parameter space, from $|\Theta^{\mathfrak{C}}|$ to $|\widetilde{\Theta}|$, at a cost of a slower rate of exponential decay. The proof of this theorem is deferred to Appendix E.2, where we also include an extension to the case where Θ is continuous.

In summary, we have shown that in the prompting stage, as the number of examples grows, the statistical error of CoT prompting decays exponentially to an intrinsic error due to pretraining. In Section 7, we will extend the above results to a few variants of CoT.

5.2 Vanilla CoT versus Vanilla ICL and Truncated CoT

Recall that vanilla ICL is a special case of CoT without intermediate reasoning steps, i.e., $H = 1$. In the following, we aim to address Question (d) raised in the introduction by directly comparing vanilla ICL and CoT under the same model. We focus on the latent variable model in (2). Let θ^* denote the task during the prompting stage, and let $\mathtt{prompt}_{\mathrm{ICL}}(n) = \{z_0^i, z_H^i\}_{i=1}^n \cup \{z_0^{\mathrm{test}}\}$ and $\mathtt{prompt}_{\mathrm{CoT}}(n) = \{z_{0:H}^i\}_{i=1}^n \cup \{z_0^{\mathrm{test}}\}$ denote a ICL prompt and a CoT prompt respectively. Thus, vanilla ICL and CoT yield estimators $\mathbb{P}_{\mathrm{LLM}}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{ICL}}(n))$ and $\mathbb{P}_{\mathrm{LLM}}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))$ respectively. The following proposition shows that CoT always outperforms vanilla ICL in an average sense.

Proposition 11 (CoT Outperforms Vanilla ICL) *Let* π *denote the prior distribution over* Θ. *Consider the ideal case where the LLM is perfectly pretrained, for any number of demonstration examples* $n \geq 0$, *we have*

$$\begin{aligned}
&\mathbb{E}_{\theta^* \sim \pi} \mathbb{E}_{\mathtt{prompt}_{\mathrm{CoT}}(n) \sim \mathbb{P}(\cdot \,|\, \theta^*)} \Big[\mathtt{KL}\Big(\mathbb{P}_{\mathrm{LLM}}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \Big) \Big] \\
&\quad \leq \mathbb{E}_{\theta^* \sim \pi} \mathbb{E}_{\mathtt{prompt}_{\mathrm{ICL}}(n) \sim \mathbb{P}(\cdot \,|\, \theta^*)} \Big[\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{ICL}}(n)\big) \Big) \Big].
\end{aligned}$$

This proposition shows that, on average, **CoT performs at least as well as vanilla ICL**. Intuitively, conditioning on more information improves the posterior estimate, and since both are BMA estimators, a better posterior reduces statistical error. This property also applies to truncated CoT methods, where demonstrations omit some intermediate steps.

Vanilla ICL	Informative CoT	PI CoT-(a)	PI CoT-(b)	PI CoT-(c)	PI CoT-(d)
59.5%	81.5%	70.5%	2.5%	66%	80%

Table 2: Average accuracy of the six prompting methods on the CityEquation task. The results are based on 200 random testing instances. "PI CoT" stands for the partially informative chain of thought method.

We provide the formal details and extend Proposition 11 in Appendix G.2, showing that, on average, including more reasoning steps in the prompt is always advantageous.

However, it's important to note that the dominance of CoT over vanilla ICL **does not hold universally** for every task $\theta^* \in \Theta$. There may be specific tasks and prompt examples where CoT underperforms compared to vanilla ICL. This occurs when the intermediate reasoning steps provide little useful information. This phenomenon has been observed empirically, such as in Lanham et al. (2023) on the HellaSwag benchmark (Zellers et al., 2019). We also demonstrate this with numerical experiments on a custom toy task.

Vanilla ICL vs. CoT on the CityEquation Task. We create an arithmetic reasoning task called "CityEquation", where arithmetic operations are performed using city names. In these equations, cities are either added or subtracted, with the result computed by substituting each city name with its longitude. For example, "$\texttt{Paris} + \texttt{Beijing} = 118$" is true because the longitudes of Paris and Beijing are 2 and 116, respectively.

Data Construction. We choose 20 major cities around the world, and generate random city equations by randomly selecting two cities and an operation in $\{+, -\}$. We construct the test data set using 200 distinct equations and use another 10 different equations as the examples in the prompting stage.

Prompting Methods and Results. We test six prompting methods: vanilla ICL, informative CoT, and four partially informative CoT variants. We consider an *informative* version of CoT that includes the full reasoning path and four *partially informative* versions that either contain some irrelevant facts or omit some relevant intermediate steps. In particular, in *partially informative CoT-(b)*, we include some demographic information of the cities in the equations, which, although truthful facts, are not related to longitudes, which is the key to getting the final answer. Then the last two versions (c) and (d) additionally include some useful reasoning steps. See Appendix K for more details. When evaluating these methods, we include 10 examples in the prompt, followed by a new testing instance. The prompt is passed to GPT-4 (Achiam et al., 2023) with the temperature set to zero, and the reported answer is compared with the desired answer to evaluate the accuracy. We report the average accuracy over 200 random testing instances in Table 2.

As shown in the table, informative CoT achieves the highest accuracy. The four versions of partially informative (PI) CoT either underperform or outperform vanilla ICL, depending on the relevance of the intermediate steps. For details on failure cases and further interpretation, see Appendix K. The regression experiments in Section 8 further corroborate Proposition 11: across both the two-layer MLP and decision-tree tasks, the mean MSE of

vanilla ICL is higher than that of CoT, yet the confidence intervals overlap, consistent with average-case dominance but not instance-wise strict dominance.

6. Statistical Errors of CoT with Pretraining Errors

Recall that in Lemma 4 we decompose the CoT error $\texttt{err}_{\text{CoT}}$ into (i) a pretraining error (9) and (ii) a prompting error (8). We have analyzed the prompting error in Section 5. In this section, we establish a statistical analysis of the pretraining error and then obtain a complete characterization of the error of the estimator constructed by "pretrained LLM + CoT prompting". Thus, in this section, we answer Question (b) raised in the introduction.

We rigorously describe the pretraining process in Section 6.1. Next, in Section 6.3, we construct a class of transformer networks that directly approximates the underlying distribution $\mathbb{P}$. We characterize the pretraining error in Section 6.2 and establish the statistical errors of CoT under realistic assumptions in Section 6.4.

6.1 Setup of LLM Pretraining

Recall that we introduce the pretraining of LLM in Section 4.2. We consider the pretraining of an autoregressive LLM with data sampled from the model in (2). The LLM is a transformer that maps a sequence of reasoning steps to a probability distribution.

Transformer Architecture. We let $\mathcal{TF}(D, \eta, r, d_F, d_k, d_v)$ denote the class of transformers that maps a sequence of reasoning steps to a probability distribution over $\mathcal{L}$, where the input sequence is embedded in $\mathbb{R}^r$, followed by D sequentially stacked transformer blocks and a final softmax layer that outputs a distribution. Here the embedding contains both token and positional embedding. Moreover, each of the D transformer blocks includes a multi-head attention (MHA) layer with η parallel heads and a fully connected feedforward (FF) layer. The embedding dimensions of the queries, keys, and values in Multi-Head Attention (MHA) in (18) are d_k, d_k, and d_v, respectively. Here queries and keys share the same dimension to calculate the inner product. We assume that $d_v = r$ to guarantee that the output dimension is the same as the input dimension, which avoids defining the dimensions for modules in all the layers. The results for the general case can be easily generalized. For the Feed-Forward (FF) layer in (19), the dimensions of the hidden feature and the output are d_F and r, respectively. Both components have residual connections, followed by layer normalization. See Figure 4 for an illustration of the transformer architecture, where the transformer block consisting of a MHA and a FF layer is illustrated on the right.

As for the network parameters, for any $d \in [D]$, we let $W^d_{\text{mha}} = (W_i^{Q,d}, W_i^{K,d}, W_i^{V,d})_{i=1}^{\eta}$ denote the weight matrices of the η heads, and let $W^d_{\text{ff},1}$ and $W^d_{\text{ff},2}$ denote the weight matrices of the FF layer. The mathematical expressions of MHA and FF layers are given in (18) and (19). We adopt $\gamma_1^d, \gamma_2^d \in \mathbb{R}^{r\times r}$ to denote the parameters of the residual links in the d-th module. Moreover, for the output softmax layer, we fix the temperature as τ and let W_{softmax} denote the weight matrix. For ease of presentation, we defer the mathematical details of the transformer to Appendix I.1. We let ρ denote all the network parameters of the transformer. Furthermore, we consider a bounded transformer class with parameters

bounded in

$$\begin{aligned}\mathcal{P}_{\mathrm{LLM}} = \Big\{\rho : \|\gamma_1^d\|_\infty, \|\gamma_2^d\|_\infty \le 1, \|W_i^{Q,d}\|_F, \|W_i^{K,d}\|_F, \|W_i^{V,d}\|_F \le B_M, \\ \|W_{\mathrm{ff},1}^d\|_F, \|W_{\mathrm{ff},2}^d\|_F \le B_F, \|W_{\mathrm{softmax}}\|_{1,2} \le B_S, \forall d \in [D], i \in [\eta]\Big\} \\ \subseteq \mathcal{TF}(D, \eta, r, d_F, d_k, d_v),\end{aligned} \tag{10}$$

where B_M, B_F, B_S are upper bounds on the norms of the weight matrices. We assume these parameters are fixed and larger than one.

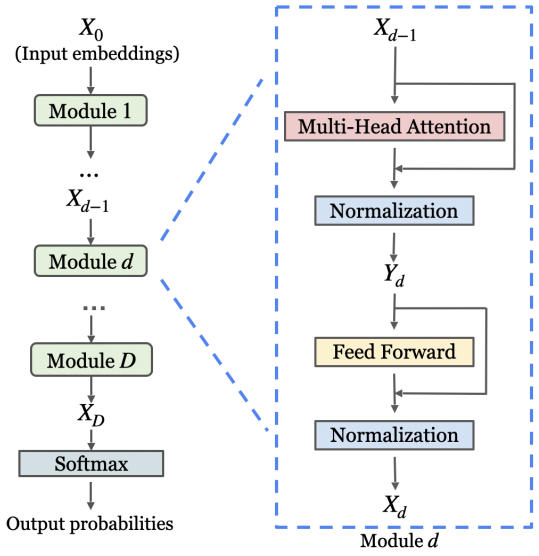


Figure 4: An illustration of a transformer with depth D. The left-hand side shows the network with D sequentially stacked transformer blocks followed by a final softmax layer. The right-hand side zooms in on a single transformer block, which comprises a MHA layer and a FF layer, connected by normalization layers.

Pretraining Data and MLE Estimation. The dataset, denoted by $\mathcal{D}_{N,T}$ contains N independent trajectories, each with T examples. For each trajectory $\ell \le N$, we first sample an i.i.d. task θ_ℓ^* from the prior π. Conditioning on the task parameter θ_ℓ^*, we generate T examples $\{s^{k,\ell}\}_{k=1}^T \sim \mathbb{P}(\cdot \,|\, \theta_\ell^*)$ according to the model in (2) and concatenate them to form a trajectory. Here $s^{k,\ell} = (z_0^{k,\ell}, \cdots, z_H^{k,\ell})$ denotes the k-th example of the task θ_ℓ^*, and we view each $z_j^{k,\ell}$ as a reasoning step. Thus a sequence contains $T(H+1)$ elements in total. For any $h \ge 0$, the reasoning steps before $z_h^{t,\ell}$ is denoted by $S_h^{t,\ell} = (\Upsilon_{t-1,\ell}, \{z_j^{t,\ell}\}_{j=0}^{h-1})$, where $\Upsilon_{t-1,\ell}$ contains the first $t-1$ examples of the ℓ-th trajectory. Then we can write the dataset $\mathcal{D}_{N,T}$ as $\{(S_h^{t,\ell}, z_h^{t,\ell})\}_{h=0,t=1,\ell=1}^{H,T,N}$. The pre-trained LLM, denoted by $\widehat{\rho}$, as defined in (3), is obtained by solving the maximum likelihood estimation (MLE) based on the dataset $\mathcal{D}_{N,T}$. We set $\mathbb{P}_{\mathrm{LLM}} = \mathbb{P}_{\widehat{\rho}}$, where $\mathbb{P}_{\widehat{\rho}}$ denotes the conditional distribution specified by the transformer with parameter $\widehat{\rho}$. We neglect the optimization issue and assume that the MLE in (3) can be

obtained. We note that when the transformer class is sufficiently expressive, we expect that $\mathbb{P}_{\mathrm{LLM}}$ learns the conditional distribution of $z_h^{t,\ell}$ given $S_h^{t,\ell}$.

We note that in the pretraining process described above, we train a transformer that takes all sequences of the reasoning steps $S \in \mathcal{L}^*$ as input and predicts the next reasoning step $z \in \mathcal{L}$. This setup can be easily generalized to the autoregressive prediction of the next token instead of the next reasoning step based on the prompt.

6.2 Pretraining Error Analysis

We will show that the pretraining error can be written as a sum of an **approximation error** and a **generalization error**. The analysis is based on the PAC-Bayes framework (McAllester, 1998; Alquier, 2021). Before presenting this result, we introduce two regularity assumptions as follows.

Assumption 12 *Note that we assume that each reasoning step in $\mathcal{L}$ is identified with a unique Euclidean vector. We assume that $\mathcal{L}$ is a bounded set. That is, there exists $R > 0$ such that $\|z\|_2 \le R$ for all $z \in \mathcal{L}$.*

This assumption ensures that the input space of the transformer network is bounded, which is commonly imposed by the literature on nonparametric statistics (Zhang et al., 2023a).

Assumption 13 *For the model in (2), we assume that for any $z \in \mathcal{L}, \theta \in \Theta$, and any sequence of reasoning steps $S \in \mathcal{L}^*$, $\mathbb{P}(z \,|\, S, \theta) > c_0$ for some constant $c_0 > 0$.*

This assumption requires the conditional probability of the next reasoning step z to be lower-bounded at any element of $\mathcal{L}$. This means that the generation of the reasoning path is stochastic. Similar assumptions have also been imposed in existing works (Xie et al., 2021; Jiang, 2023). As we will see in Appendix H.1, this assumption implies Assumption 3 with $b^* = \log(\max\{c_0^{-1}, 1 + |\mathcal{L}| \exp(B_S/\tau)\})$, where B_S appears in (10).

Besides, to simplify the notation, we let $\mathbb{E}_{S\sim\mathcal{D}}$ denote the empirical distribution with respect to the pretraining data set $\mathcal{D}_{N,T}$. Specifically, for any function $f : \mathcal{L}^* \to \mathbb{R}$ we define

$$\mathbb{E}_{S\sim\mathcal{D}}[f(S)] = N^{-1}(H+1)^{-1} \cdot T^{-1} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \mathbb{E}[f(S_h^{t,\ell})], \tag{11}$$

where the expectation is taken with respect to the joint distribution of $\mathcal{D}_{N,T}$. We establish the pretraining error in the following proposition.

Proposition 14 (Pretraining Error Bound) *Under Assumptions 12 and 13, with probability at least $1-\delta$, the pretrained LLM $\mathbb{P}_{\widehat{\rho}}$ in (3) satisfies that*

$$\begin{aligned}
&\mathbb{E}_{S\sim\mathcal{D}}\Big[\mathtt{TV}\big(\mathbb{P}(\cdot\,|\,S),\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S)\big)\Big] \\
&\quad= O\Bigg(\underbrace{\inf_{\rho^*\in\mathcal{P}_{\mathrm{LLM}}}\sqrt{\mathbb{E}_{S\sim\mathcal{D}}\mathtt{KL}\big(\mathbb{P}(\cdot|S),\mathbb{P}_{\rho^*}(\cdot|S)\big)}+\frac{\sqrt{b^*}\log(TH/\delta)}{N^{1/4}}}_{\textit{approximation error}} \\
&\qquad\qquad+\underbrace{\frac{1}{\sqrt{N}}\Big(\bar{D}\log(1+NTH\bar{B})+\log\frac{TH}{\delta}\Big)}_{\textit{generalization error}}\Bigg),
\end{aligned}$$

where $\bar{B}=\tau^{-1}R\eta B_S B_F^2 B_M^3$ and $\bar{D}=D^2r(d_F+d_k+r)+r\cdot d_y$ are parameters determined by the transformer architecture in (10). Besides, we have $b^*=\log\big(\max\{c_0^{-1},1+|\mathcal{L}|\exp(B_S/\tau)\}\big)$. We use $\Delta_{\mathrm{pre}}(N,T,\delta)$ to denote the right-hand side of this equation.

This Proposition is proved using the PAC-Bayes framework. The proof is adapted from Zhang et al. (2023a) and deferred to Appendix H.1.

Proposition 14 shows that the pretraining error can be decomposed into an approximation error and a generalization error. The approximation error is a sum of a KL divergence term, and an additional $N^{-1/4}$ term that arise from concentration. The approximation error is small if the transformer class is sufficiently expressive. Moreover, the generalization error decays to zero as N increases, and $\bar{D}$ captures the complexity of the transformer model. This error increases with the sequence length $T\cdot H$ mildly through a logarithmic factor.

6.3 Transformers as Conditional Distribution Approximators

In the following, we present the approximation result. We will construct a transformer with parameters in $\mathcal{P}_{\mathrm{LLM}}$ that captures the multi-step reasoning structure of CoT. More importantly, we will prove that the approximation error decays to zero exponentially as the network depth D increases. We first present an informal version of the theory as follows.

Proposition 15 (Approximation Error, Informal) *Let $S_h^t=(\Upsilon_{t-1},\{z_j^t\}_{j=0}^{h-1})$ be the sequence of reasoning steps that includes $t-1$ examples of reasoning paths Υ_{t-1} and the first $h-1$ steps of the t-th example $\{z_j^{t-1}\}_{j=0}^{h}$. Then if the target distribution $\mathbb{P}$ of the model in (2) has a sufficiently smooth density and the transformer model $\mathcal{P}_{\mathrm{LLM}}$ in (10) is sufficiently expressive, then there exists a transformer with at most $\mathcal{O}(D)$ number of blocks and parameter $\rho^*\in\mathcal{P}_{\mathrm{LLM}}$ such that*

$$\max_{S_h^t\in\mathcal{L}^*}\mathtt{KL}\big(\mathbb{P}(z_h^t=\cdot\,|\,S_h^t),\mathbb{P}_{\rho^*}(z_h^t=\cdot\,|\,S_h^t)\big)=O\bigg(\exp\bigg(-\frac{\big(D-C\log(2H)\big)/H\big)^{1/4}}{5B}\bigg)\bigg),$$

for any $t\in[T]$ and $0\le h\le H$ when D goes to infinity. In particular, $C>0$ is an absolute constant, and B appears in Assumption 37.

This proposition shows that the approximation error decays exponentially in D. This exponential accuracy is based on the construction of a neural network approximator in

Elbrächter et al. (2021) for smooth functions. Moreover, note that S_h^t has $(t-1)\cdot(H+1)+h$ reasoning steps in total. An appealing feature of this proposition is that the approximation error is independent of t, thanks to leveraging the permutation invariance structure of the target distribution. Specifically, when viewing $\mathbb{P}(z_h^t = \cdot \,|\, S_h^t)$ as a function of S_h^t, it is invariant to the permutation of the $t-1$ examples. Our transformer approximator directly leverages such invariance in the attention mechanism, thus obtaining an approximation error independent of t. However, $\mathbb{P}(z_h^t = \cdot \,|\, S_h^t)$ can be drastically different across $h \in [H]$. Concretely, each reasoning step represents a different procedure described by different distributions. To handle this fact, our transformer treats each step $h \in [H]$ differently and uses a separate transformer subnetwork to predict each z_h. These subnetworks, each containing multiple attention blocks, are stacked vertically. And we leverage the position embedding to let the transformer identify the step index h of S_h^t, and then pass the input to the h-th subnetwork. See Figure 5 for an illustration of the construction. The formal statement of Proposition 15 and its detailed proof are deferred to Appendix H.2.

Remark 16 (Intuition for exponential decay in D) *The exponential decay in D follows from the deep ReLU approximation theory of Elbrächter et al. (2021): a network of depth d approximates a sufficiently smooth function with error $\exp(-d^{c_1}/c_2)$ for constants $c_1, c_2 > 0$ depending on the smoothness of the target distribution, since each additional layer reduces the approximation error by a constant multiplicative factor, and the product over d layers decays exponentially in d. The argument is D/H rather than D because the network has H submodules sharing total depth D (Figure 5), so each submodule has depth $\approx D/H$, and the theorem is applied per submodule.*

6.4 Statistical Error of CoT with Out-of-Distribution Queries

In this section, we combine the analysis of pretraining error and prompting error to derive a comprehensive characterization of the statistical error of CoT. Moreover, we will tackle the **query error**, which arises due to the distributional shift of the test instance z_0^{test}.

Specifically, the query error arises if z_0^{test} is not sampled from the same task θ^* as the n prompt examples. For instance, if the examples in the prompts are about the "solving arithmetic problems", but we query a new philosophical question. Then the knowledge incorporated in the examples is not useful for answering the query and thus we expect a large error. More rigorously, let $\Upsilon_n = \{s_j\}_{j=1}^n$ denote the n CoT examples sampled the model in (2) with task θ^*. Thus, $\texttt{prompt}_{\text{CoT}}(n) = (z_0^{\text{test}}, \Upsilon_n)$. Under this model, the query has a distribution $\mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta^*)$. We let $\mu(z_0^{\text{test}} = \cdot \,|\, \Upsilon_n)$ denote the distribution of an out-of-distribution (OOD) query, whose distribution might depend on Υ_n. The difference between these two distributions reflects the query error.

Besides, we define $\mathbb{P}_{\text{CoT},\mu,\Upsilon_n}$ as the joint distribution of Υ_n and an OOD query, parameterized by the shift distribution μ and the demonstrations Υ_n:

$$\mathbb{P}_{\text{CoT},\mu,\Upsilon_n}\big(\texttt{prompt}_{\text{CoT}}(n) \,|\, \theta^*\big) = \mathbb{P}(\Upsilon_n \,|\, \theta^*) \cdot \mu(z_0^{\text{test}} \,|\, \Upsilon_n).$$

We make the following assumption about the distributional shift due to the OOD query.

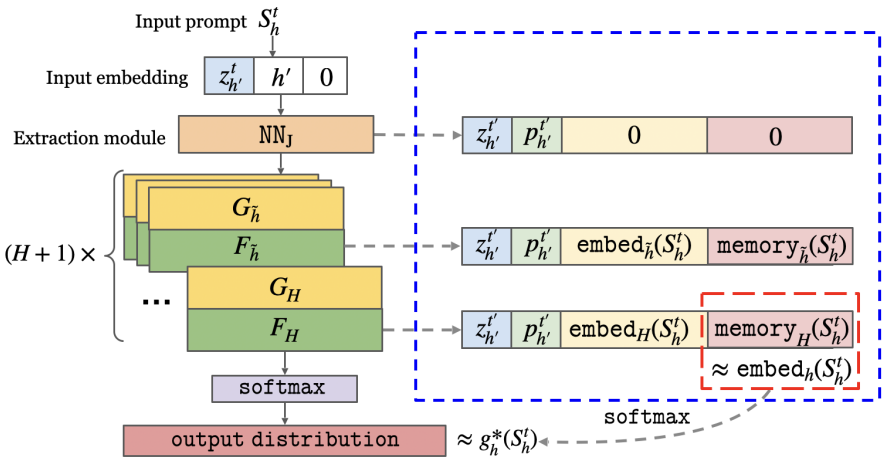


Figure 5: An illustration of the transformer constructed for proving Proposition 15. After the input embedding, the first module $\mathtt{NN}_\mathtt{J}$ extracts the step-index h from S_h^t and copies it to all previous positions. Then, the transformer goes through $H+1$ submodule pairs $\{G_{\tilde{h}}, F_{\tilde{h}}\}_{\tilde{h}=0}^{H}$, where each pair is designed to approximate the conditional distribution associated with a specific step $h \in [H]$. The output of F_H goes through a softmax output layer to get the final output distribution, which is close to the target distribution $\mathbb{P}(\cdot \,|\, S_h^t)$.

Assumption 17 *We assume the distributional shift is mild in the sense that $\mathbb{P}_{\mathrm{CoT},\mu,\Upsilon_n}$ is covered by the pretraining distribution. That is, for any fixed $\theta^* \in \Theta$, there exists a constant $\kappa > 0$ such that $\mathbb{P}_{\mathrm{CoT},\mu,\Upsilon_n}(\mathtt{prompt}_{\mathrm{CoT}}(n) \,|\, \theta^*) \le \kappa \mathbb{P}(\mathtt{prompt}_{\mathrm{CoT}}(n) \,|\, \theta^*)$ for any $\mathtt{prompt}_{\mathrm{CoT}}(n) \in \mathcal{L}^*$ with $n \le T$. For ease of notation, we abbreviate $\mathbb{P}_{\mathrm{CoT},\mu,\Upsilon_n}$ as $\mathbb{P}_{\mathrm{CoT}}$ in the remainder of the text.*

Here κ captures the magnitude of the distributional shift. This assumption requires that the test query z_0^{test} cannot be too arbitrary—its distribution should have sufficient density under the pretraining distribution. Intuitively, we cannot expect the LLMs to answer questions beyond the knowledge contained in the pretraining dataset. Note that when there is no distributional shift in the query, we have $\kappa = 1$. Then the analysis of $\mathtt{err}_{\mathrm{prompt}}$ is reduced to Theorem 7. Recall that the pretraining data distribution mixes the task distribution $\theta \sim \pi$. Under the model in (2), this assumption is satisfied if we set

$$\kappa = \sup_{z \in \mathcal{L}, S \in \mathcal{L}^*} \left| \mu(z_0^{\mathrm{test}} = z \,|\, \Upsilon_n = S) / \mathbb{P}(z_0^{\mathrm{test}} = z \,|\, \theta^*) \right|,$$

which is no more than $1/c_0$ under Assumption 13. Thus, the distributional shift is small if task θ^* is well covered in the pretraining distribution, and the distribution μ is similar to the true query distribution $\mathbb{P}(z_0^{\mathrm{test}} = \cdot \,|\, \theta^*)$.

Combining Pretraining and Prompting Errors. With this assumption, we combine Theorem 7, Proposition 15 and Proposition 14 to obtain a complete characterization of the statistical error of CoT — the statistical estimator obtained by first pretraining an LLM

using dataset $\mathcal{D}_{NT}$ and then prompting the pretrained LLM using a CoT prompt with n examples. The result is given in the following corollary.

Corollary 18 (Complete Characterization of $\texttt{err}_{\text{cot}}$) *Recall that $\texttt{err}_{\text{CoT}}$ is defined in (4). Under Assumptions 6, 12, 13, and 17, with probability at least $1-\delta$, we have*

$$\mathbb{E}_{\mathbb{P}_{\text{CoT}}}[\texttt{err}_{\text{CoT}}] \leq \mathcal{O}\bigg(Hb^* \bigg(\frac{\pi(\Theta^{\complement})}{\pi(\theta^*)} C(\theta^*)\kappa \bigg)^{1/4} \cdot \left|\Theta^{\complement}\right|^{1/2} \cdot \exp(-n\lambda/2) + \kappa T H \pi(\theta^*)^{-1} \cdot (1+b^*) \cdot \Delta_{\text{pre}}(N,T,\delta) \bigg)$$

when Θ is finite, where $C(\theta^*)$ is defined as

$$C(\theta^*) = \sup_{\theta \in \Theta^{\complement}} \sqrt{\chi^2\big(\mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta), \mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta^*)\big) + 1}.$$

In this corollary, we combine the previous result for the perfectly pretrained model in Theorem 7 with the pretraining error analysis in Proposition 14, and the query error that quantifies the distributional shift of the prompt. The proof can be found in Appendix H.4. For conciseness, we stated the result for the case where Θ is discrete and finite. This can be generalized the result to the continuous case by applying the second part of Theorem 27. This corollary shows that, when the distributional shift is mild ($\kappa = \mathcal{O}(1)$) and $|\Theta|$ is finite, to achieve any desired accuracy level $\varepsilon \in (0,1)$, it suffices to let:

- $D = H\Big(5B \cdot \log\big(\kappa T H \pi(\theta^*)^{-1} \cdot 3(1+b^*)/\epsilon\big)\Big)^4 + C\log(2H)$ for the transformer depth, where the absolute constant $C > 0$ is from Proposition 15.
- $n \geq 2/\lambda \cdot \log\left(3Hb^*\Big(\pi(\Theta^{\complement}) \cdot (\pi(\theta^*))^{-1} \cdot C(\theta^*) \cdot \kappa\Big)^{1/4} \cdot \left|\Theta^{\complement}\right|^{1/2}/\epsilon\right)$ in $\texttt{prompt}_{\text{CoT}}(n)$.
- $T \geq n+1$, and $N \geq \Big(3\sqrt{b^*} \cdot \log(TH/\delta) \cdot \kappa T H \pi(\theta^*)^{-1} \cdot (1+b^*)/\epsilon\Big)^4$ in the pretraining dataset $\mathcal{D}_{N,T}$.

Then Corollary 18 shows that $\mathbb{E}_{\mathbb{P}_{\text{CoT}}}[\texttt{err}_{\text{CoT}}] \leq \varepsilon$ with probability at least $1-\delta$.

Remark 19 (Consistency of the N and D requirements) *The sufficient conditions above provide explicit formulas for N and D in terms of T and ε, with H treated as a constant. From these formulas, $N \geq \Omega(D^4)$ follows directly: the required N scales as $(TH/\varepsilon)^4 \cdot (\log TH)^4$ — polynomial in TH/ε — while the prescribed D scales as $H \cdot (\log(TH/\varepsilon))^4$ — a polynomial in $\log(TH/\varepsilon)$. Since a polynomial in TH/ε grows far faster than any polynomial in $\log(TH/\varepsilon)$, the stated lower bound on N automatically exceeds D^4.*

7. Statistical Errors of Variants of CoT

The predictions of LLMs are inherently stochastic, which is a main source of LLM hallucination (Huang et al., 2023a; Tonmoy et al., 2024). To increase the prediction accuracy, various selection techniques such as majority vote (Wang et al., 2022) and tree search (Yao

et al., 2023) are combined with CoT. In the following, we modify Theorem 7 for a few variants of CoT, including Self-Consistency CoT (Wang et al., 2022), Tree-of-Thought (Yao et al., 2023), and Selection-Inference (Creswell et al., 2022). For simplicity, we also assume zero pretraining error and input query does not have a distributional shift, i.e., $\mathbb{P}_{\text{LLM}} = \mathbb{P}$ and $z_0^{\text{test}} \sim \mathbb{P}(\cdot \,|\, \theta^*)$.

7.1 Self-Consistency CoT (SC-COT)

Given the same prompt as in vanilla CoT, i.e., $\texttt{prompt}_{\text{CoT}}(n)$, Self-Consistency (SC)-CoT first generate K i.i.d. reasoning paths and then output the final answer by a *majority vote*. That is, we first sample K i.i.d. reasoning paths $\{z_{1:H}^{\text{test},i}\}_{i=1}^K \sim \mathbb{P}\big(\cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)$ and then report the mode of the empirical distribution of $\{y^{\text{test},i}\}_{i=1}^K$, denoted by y_K^*. The empirical distribution of $\{y^{\text{test},i}\}_{i=1}^K$ is denoted by $p_K(y) = K^{-1}\sum_{i=1}^K \mathbf{1}\{y^{\text{test},i} = y\}$, $\forall y \in \mathcal{L}$. The sample mode y_K^* is defined as $y_K^* = \text{argmax}_{y\in\mathcal{L}}\, p_K(y)$, where we pick any element if there are multiple maximizers. See Figure 6 for an illustration.

Recall that if the underlying task θ^* is already known, the answers should be generated according to $\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*)$. We assume the desired answer is the mode of this distribution and it is unique.

Assumption 20 *We define y^* as the mode of the distribution $\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*)$, i.e., $y^* = \text{argmax}_{y\in\mathcal{L}}\, \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*)$ Moreover, we define the gap between the mode and the second-largest probability mass as*

$$\epsilon = \min_{y\in\mathcal{L}, y\neq y^*} \{\mathbb{P}\big(y^{\text{test}} = y^* \,|\, z_0^{\text{test}}, \theta^*\big) - \mathbb{P}\big(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*\big)\},$$

which is assumed to be strictly positive.

This assumption ensures that the population mode y^* is uniquely defined with a margin ϵ. This condition is satisfied by reasoning problems where the answer is unique, e.g., factual commonsense and mathematical reasoning. Intuitively, when the number of examples in CoT prompt is large, $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ is close to $\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*)$, as guaranteed by Theorem 7. Then, when K is sufficiently large in SC-CoT, we expect that the sample mode y_K^* coincides with the population mode y^*. This justifies the effectiveness of SC-CoT.

Corollary 21 (Statistical Error of SC-CoT) *Consider SC-CoT prompting with K reasoning paths and n CoT examples. Under Assumptions 3, 6, and 20, when n is sufficiently large such that*

$$n = \Omega\Big(\Big(\log\big(|\Theta^{\complement}|/\pi(\theta^*)\big) + \log(1/\epsilon)\Big)\Big/\lambda\Big),$$

with probability at least $1 - e^{-\lambda n/2}$, the probability that the SC-CoT produces the wrong output decreases exponentially in K, i.e.,

$$\mathbb{P}\big(y_K^* \neq y^* \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \leq 2|\mathcal{L}| \cdot \exp\bigg(-\frac{3K\epsilon^2}{24 + 8\epsilon}\bigg).$$

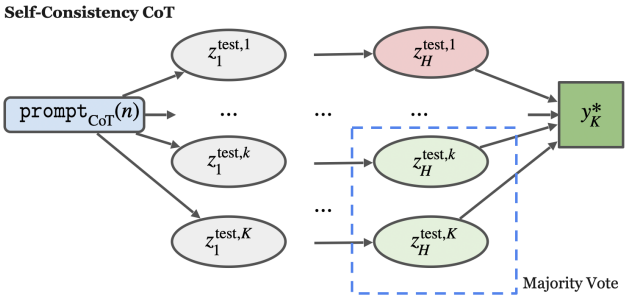


Figure 6: An illustration of the SC-CoT prompting method. This method creates the final answer y_K^* based on two steps. First, we sample K i.i.d. reasoning paths $\{z_{0:H}^{\text{test},i}\}_{i=1}^K$ given the CoT prompt, and then report y_K^* by a majority vote based on $\{y^{\text{test},i} = z_H^{\text{test},i}\}_{i=1}^K$.

This corollary shows that sampling K independent reasoning paths boosts the output accuracy. In particular, when $\epsilon \in (0,1)$, for any $\delta \in (0,1)$, as long as $K = \Omega(\log(|\mathcal{L}|/\delta)/\epsilon)$, $y_K^* = y^*$ holds with probability at least $1 - e^{-\lambda n/2} - \delta$. The proof of this corollary can be found in Appendix F.1.

7.2 Tree-of-Thought (ToT)

Recall that SC-CoT samples multiple parallel reasoning paths and performs a selection in the last step. Tree-of-Thought (Yao et al., 2023) instead proposes to include selection in each step. In this setup, the goal is to generate a reasoning path $z_{1:H}^{\text{test}}$ that maximizes a task-specific value function V_θ^*. We define this population problem as follows.

Population Problem. The goal of ToT is to select the optimal reasoning path that solves a desired task. Mathematically, for each step h, let $t_h = (z_0, \ldots, z_h)$ denote the partial history up to step h. Let V_{θ^*} be a function that maps each partial history to a value in $[0,1]$. Intuitively, V_{θ^*} can be viewed as the success probability of the partial history for solving task θ^*. Starting from z_0^{test}, the optimal reasoning path is obtained by solving

$$t_h^{\text{test},*} = (t_{h-1}^{\text{test},*}, z_h^{\text{test},*}), \qquad \text{where} \quad z_h^{\text{test},*} = \underset{z_h^{\text{test}} \in \mathcal{L}}{\operatorname{argmax}} V_{\theta^*}(t_{h-1}^{\text{test},*}, z_h^{\text{test}}), \quad t_0^{\text{test},*} = z_0^{\text{test}}. \tag{12}$$

The maximization domain for the next reasoning step, $z_h^{\texttt{test}}$, is taken over $\mathcal{L}$, a finite set representing all valid next steps. In practice, this set $\mathcal{L}$ can be defined as either the standard vocabulary of the language model or a specialized collection of domain-specific reasoning primitives. Moreover, let $\mathbb{P}(z_{0:H}^{\text{test}} = \cdot \,|\, \theta^*)$ be the task-specific distribution of the multi-step latent variable model defined in (2). At the population level, the goal is to draw samples from such a distribution, and select the optimal reasoning path according to the value function V_{θ^*}. In the following, we condition on $\texttt{prompt}_{\text{CoT}}(n)$, and thus the optimal reasoning path $z_{0:H}^{\text{test},*}$ can be regarded fixed.

Tree-of-Thought with Breadth-First-Search. As we do not have access to the distribution $\mathbb{P}(z_{0:H}^{\text{test}} = \cdot \,|\, \theta^*)$, ToT proposes to sample from the LLM and then approximately

solve (12) via selection. To simplify the notation, for each $h \in [H]$, we denote $t_{h-1}^{\text{test}} = (z_0^{\text{test}}, \ldots, z_{h-1}^{\text{test}})$, which is the partial history of the test example up to step $h-1$. In step h, instead of passing the complete prompt $\texttt{prompt}_{\text{CoT}}(n)$, we truncate each demonstration in $\texttt{prompt}_{\text{CoT}}(n)$ up to step h and denote the truncated prompt by $\texttt{prompt}_h(n) = \{z_{0:h}^i \,|\, z_{0:H}^i \in \texttt{prompt}_{\text{CoT}}(n)\}$. Then the LLM samples $z_h^{\text{test}} \sim \mathbb{P}(\cdot \mid \texttt{prompt}_h(n), t_{h-1}^{\text{test}})$ and obtain t_h^{test}, and so on.

In the sequel, we only discuss a version of ToT that maintains a candidate set of partial histories $\mathcal{T}_h$ for each step h, constructed using Breadth-First-Search (BFS). Specifically, the algorithm involves two integer parameters, K and B, which specify the number of samples drawn in each step and the size of each $\mathcal{T}_h$, respectively. Let $\mathcal{T}_0 = \{z_0^{\text{test}}\}$. Suppose $\mathcal{T}_h$ is already constructed and $|\mathcal{T}_h| = B$. Let its elements be denoted by $\{t_h^1, \ldots, t_h^B\}$. For any $b \in [B]$, the algorithm will include both t_h^b and $\texttt{prompt}_{h+1}(n)$ as the prompt sequence and do K i.i.d. one-step reasoning with the perfectly trained LLM, i.e., $\{z_{h+1}^{b,i}\}_{i=1}^K \overset{i.i.d.}{\sim} \mathbb{P}(\cdot \mid \texttt{prompt}_{h+1}(n), t_h^b)$. Thus, we obtain KB partial histories for the $(h+1)$-th step: $\{(t_h^b, z_{h+1}^{b,i})\colon i \in [K], b \in [B]\}$. Then we sort these partial histories according to the value function V_{θ^*}, and define $\mathcal{T}_{h+1}$ as the top B elements. That is,

$$\mathcal{T}_{h+1} = \{t_{h+1}^b\}_{b\in[B]} = \{\text{top } B \text{ } (t_h^b, z_{h+1}^{b,i})\text{'s in terms of } V_{\theta^*}(t_h^b, z_{h+1}^{b,i}), i \in [K], b \in [B]\}. \quad (13)$$

Finally, when $\mathcal{T}_H$ is constructed, we define $\widehat{t}_H = \text{argmax}_{t_H \in \mathcal{T}_H} V_{\theta^*}(t_H)$ and use $\widehat{t}_H$ as the final prediction. See Figure 7 for an illustration.

Compared to SC-CoT, ToT uses a more sophisticated selection method based on the value function. For this to be effective, V_{θ^*} need to align well with the task-specific distribution $\mathbb{P}(\cdot \,|\, \theta^*)$. Recall that for each h we do one-step reasoning for K times and only keep a candidate set of the histories. To ensure that the desired reasoning path is contained in the candidate set, we require that the optimal one-step reasoning $z_h^{\text{test},*}$ can be sampled out with high probability for each $h \in [H]$, which in turn requires sufficient coverage of $t_h^{\text{test},*}$ under $\mathbb{P}(\cdot \,|\, \theta^*)$ which is approximated by the LLM. For example, $t_h^{\text{test},*}$ should have sufficient probability under $\mathbb{P}(\cdot \,|\, \theta^*)$, and different tasks should have sufficient separation. We impose the following assumption for theoretical analysis.

Assumption 22 *For the given task θ^*, we assume the optimal reasoning path t_H^* is uniquely defined by (12). Moreover, we assume that the task θ^* is well covered by the pretraining distribution, i.e., $\pi(\theta^*) > 0$. Furthermore, we assume that the tasks in Θ are well separated such that the following two conditions are satisfied:*

(i) Task θ^* is uniquely identified by the optimal reasoning path, i.e., $\theta^* = \text{argmax}_{\theta' \in \Theta} \mathbb{P}(t_h^{\text{test},*} \mid \theta = \theta')$ for each $h \in [H]$;

(ii) For any $h \in [H]$, there exists $\lambda_h > 0$ such that $H^2\big(\mathbb{P}(Z_{0:h} = \cdot \,|\, \theta), \mathbb{P}(Z_{0:h} = \cdot \,|\, \theta^*)\big) \geq \lambda_h$. for all $\theta \neq \theta^*$, where $H(\cdot,\cdot)$ denotes the Hellinger distance.

We note that here we require a separation in Hellinger distance for every step h, which is slightly stronger than Assumption 6. Moreover, condition (i) above shows that the equivalence classes specified in Definition 5 are in fact singletons.

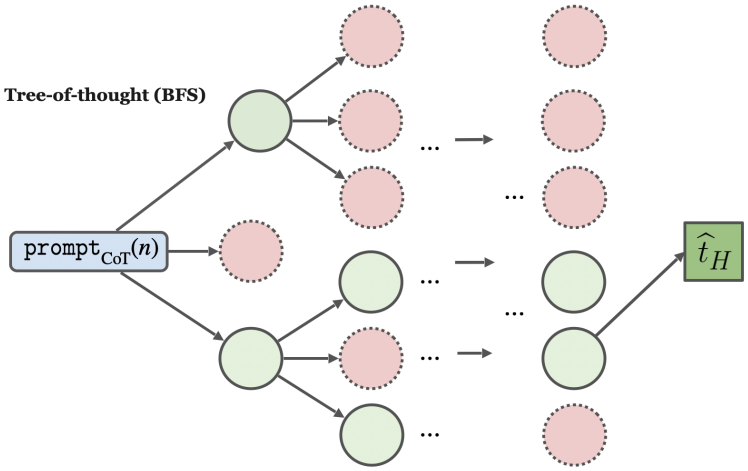


Figure 7: An illustration of ToT with BFS selection. Here BFS samples $K = 3$ i.i.d. reasoning steps based on the prompt and selects $B = 2$ partial histories. The dashed ovals represent the pruned nodes, and the others represent the active nodes. At each step $h \in [H]$, K candidates are generated per active node, and thus there are KB candidates. Then B nodes survive the selection according to the value function $V_{\theta^*}(\cdot)$. At the final step $h = H$, we select a single best candidate and output the corresponding chain as $\widehat{t}_H$.

Proposition 23 (Statistical Error of ToT) *Consider ToT prompting based on n CoT examples and BFS with $B = 1$. Let $\widehat{t}_H$ be the final output and define $\lambda^* = \min_{h\in[H]} \lambda_h$. Let $\epsilon \in (0, 1)$ be any sufficiently small number. Under Assumption 22, when n is sufficiently large such that*

$$n \geq \frac{1}{\lambda^*}\bigg(2\log\big(H|\Theta|\big) + \log\big((1-\pi(\theta^*))/\pi(\theta^*)\big) + \log(1/\epsilon)\bigg),$$

then with probability at least $1-e^{-n\lambda^*/2}$, the probability of outputting a suboptimal reasoning path $\widehat{t}_H$ decreases exponentially with K. That is, we have

$$\mathbb{P}\big(\widehat{t}_H \neq t_H^{\text{test},*} \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \leq \sum_{h=1}^{H}\Big(1 - p_h^* + \epsilon p_h^*\Big)^K,$$

where $p_h^* = \mathbb{P}(z_h^{\text{test},*} \,|\, t_{h-1}^{\text{test},*}, \theta^*)$ for each $h \in [H]$.

This proposition shows that ToT significantly reduces the probability of introducing a suboptimal optimal reasoning path, which decreases exponentially in K. Without the BFS-based selection step, even when n goes to infinity, the probability of generating $t_H^{\text{test},*}$ is only $\prod_{h\in[H]} p_h^*$. Here we only focus on the simplest case where $B = 1$, but our analysis can be generalized to $B > 1$ with some additional effort. We defer the proof to Appendix F.2.

7.3 Selection-Inference (SI)

Selection-Inference (SI) (Creswell et al., 2022) is a structured LLM reasoning method that decomposes each step of reasoning into two components—a selection module that retrieves relevant facts from the context and an inference module that predicts the next step solely based on the selected facts. To this end, Selection-Inference (SI) uses an LLM as both a selection module and an inference module through prompting. The selection module extracts information from the reasoning path and the inference module predicts the next reasoning step based on the information extracted from the selection module.

A Hierarchical Latent Variable Model. In the context of SI, we assume a special case of the model in (2) with a hierarchical structure. Specifically, we assume the latent variable θ^* has two component $\theta^* = (\theta^*_{\text{se}}, \theta^*_{\text{in}})$ and the examples of reasoning paths are i.i.d. given θ^*, which has a prior distribution π. Let $\{z_0, \ldots, z_H\}$ be a reasoning path. We let $t_h = \{z_0, \ldots, z_h\}$ be the partial history up to step h. We assume that z_{h+1} depends on t_h only through a subset of t_h, denoted by τ_{h+1}, and $\tau_{h+1} \subseteq t_h$ is selected from t_h. Specifically, the joint distribution of $\mathbb{P}(z_{0:H} \,|\, \theta^*)$ is given by

$$z_0 \sim \mathbb{P}(z_0 = \cdot \,|\, \theta^*), \qquad \tau_{h+1} \sim \mathbb{P}(\tau_{h+1} = \cdot \,|\, t_h, \theta^*_{\text{se}}), \qquad z_{h+1} \sim \mathbb{P}(z_{h+1} = \cdot \,|\, \tau_{h+1}, \theta^*_{\text{in}}), \quad (14)$$

where $t_0 = \{z_0\}$ and $t_h = t_{h-1} \cup \{z_h\}$. Intuitively, this model captures the fact that reasoning often involves summarizing existing information and making predictions. The selection module outputs a summary of the existing information that is sufficient for reasoning, and the inference module conducts reasoning based on summarized information. In the example shown in Figure 8, z_0 contains the background information and a question, τ_1 summarizes part of the information contained in z_0 and generates the first intermediate reasoning step z_1. Then τ_2 summarizes $\{z_0, z_1\}$ and z_2 is generated from τ_2, which answers the question.

SI Prompting. The SI prompting method solves a multi-step reasoning problem following the hierarchical structure specified in (14), with the unknown task θ^* inferred implicitly via in-context learning. Specifically, given a desired task θ^* and a query input z_0^{test}, we sample n i.i.d. samples from the distribution in (14), denoted by $\{z^i_{0:H}, \tau^i_{1:h},\}_{h\in[H], i\in[n]}$. We define $S_{\text{se}}(n)$ and $S_{\text{in}}(n)$ as

$$S_{\text{se}}(n) = \{t^i_{h-1}, \tau^i_h\}_{h\in[H], i\in[n]}, \qquad S_{\text{in}}(n) = \{\tau^i_h, z^i_h\}_{h\in[H], i\in[n]}, \quad (15)$$

where t^i_h is the partial history of the i-th example. That is, $S_{\text{se}}(n)$ and $S_{\text{in}}(n)$ contain the demonstration examples for selection and inference, respectively.

These demonstration examples are combined with the intermediate steps of the test example as the prompts, which are used to solve the test example. Specifically, starting from z_0^{test} and $t_0^{\text{test}} = \{z_0^{\text{test}}\}$, we generate a reasoning path via

$$\tau_h^{\text{test}} \sim \mathbb{P}_{\text{LLM}}(\cdot \,|\, S_{\text{se}}(n), t_{h-1}^{\text{test}}), \quad z_h^{\text{test}} \sim \mathbb{P}_{\text{LLM}}(\cdot \,|\, S_{\text{in}}(n), \tau_h^{\text{test}}), \quad t_h^{\text{test}} = \{t_h^{\text{test}}, z_h^{\text{test}}\}, \quad (16)$$

for all $h \in [H]$. The final output is $y^{\text{test}} = z_H^{\text{test}}$. See Figure 9 for an illustration of the prompting process. Notice that when z_0^{test}, $S_{\text{se}}(n)$ and $S_{\text{in}}(n)$ are fixed, (16) specifies a Markov chain such that the marginal distribution of y^{test} is fully determined by the LLM. We let $\mathbb{P}_{\text{SI}}(y^{\text{test}} = \cdot \,|\, S_{\text{se}}(n), S_{\text{in}}(n), z_0^{\text{test}})$ denote such a distribution, which is essentially the estimator constructed by SI prompting.

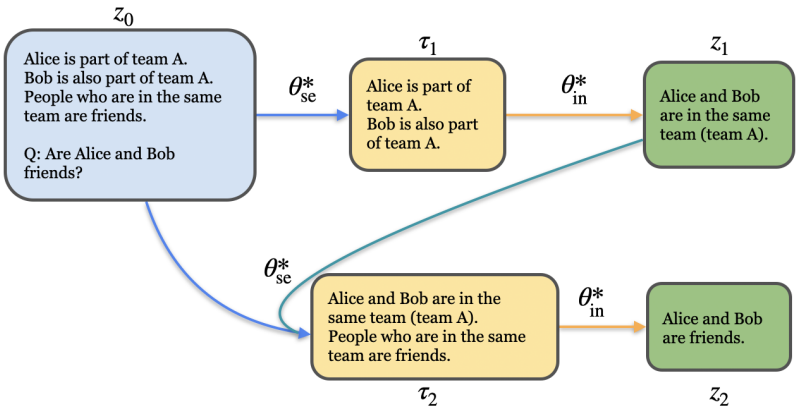


Figure 8: An example generated from the hierarchical model in (14). The query z_0 includes a question and all necessary information to answer it. The first selection step τ_1 picks two sentences from z_0 indicating which teams Alice and Bob are in. The first inference step z_1 uses the information from τ_1 to infer that Alice and Bob are on the same team. Together, τ_1 and z_1 form a single reasoning step. In the second selection step, τ_2 selects information from $t_1 = \{z_0, z_1\}$, with the first sentence of τ_2 from z_1 and the second from z_0. The second inference step z_2 answers the question using only the information provided in τ_2.

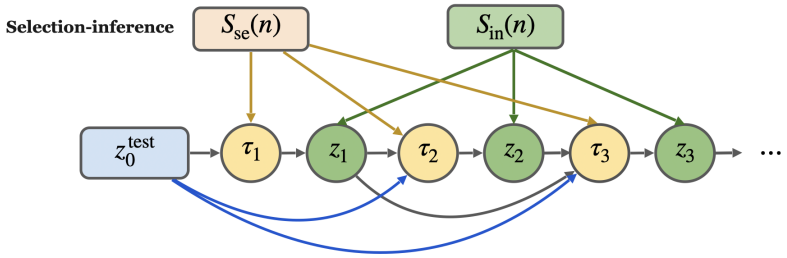


Figure 9: An illustration of the graphical model of SI prompting. In SI prompting, each step in CoT is divided into two substeps: selection and inference. The selection module retrieves relevant facts from the context t_{h-1}, and the inference module derives inference z_h based on these selected facts τ_h.

We note that SI can be viewed as a generalization of CoT in the sense that there are still H reasoning steps. However, SI has an additional hierarchical structure that selects subsets of the partial histories. We can interpret SI as a version of CoT where each step of CoT is decomposed into two substeps, aiming to conduct Bayesian inference of θ^*_{se} and θ^*_{in} separately.

In the following, we will establish the statistical error of the estimator constructed by SI prompting, under the assumption that the underlying data distribution is specified by the model in (14) and the LLM is perfectly pretrained. We introduce an assumption in the same vein as Assumption 6.

Assumption 24 *Given a target task θ^*, let $\Theta^{\complement} = \Theta \backslash \Theta_{\mathrm{eq}}(\theta^*)$ denote the complement of the equivalence class of θ^*. We assume that there exists a strict separation between the ground truth task θ^* and any other task in $\Theta^{\complement}$. Specifically, there exist positive numbers $\lambda_{\mathrm{q}}, \lambda_{\mathrm{I}}, \lambda_{\mathrm{S}} > 0$ such that*

$$\inf_{\theta \in \Theta^{\complement}} H^2\big(\mathbb{P}(z_0 = \cdot \,|\, \theta^*), \mathbb{P}(z_0 = \cdot \,|\, \theta)\big) \geq \lambda_{\mathrm{q}},$$

$$\inf_{\theta \in \Theta^{\complement}} \sum_{h=1}^{H} \mathbb{E}_{\theta^*} H^2\big(\mathbb{P}(\tau_h = \cdot \,|\, \theta^*_{\mathrm{se}}, t_{h-1}), \mathbb{P}(\tau_h = \cdot \,|\, \theta_{\mathrm{se}}, t_{h-1})\big) \geq \lambda_{\mathrm{S}},$$

$$\inf_{\theta \in \Theta^{\complement}} \sum_{h=1}^{H} \mathbb{E}_{\theta^*} H^2\big(\mathbb{P}(z_h = \cdot \,|\, \theta^*_{\mathrm{in}}, \tau_h), \mathbb{P}(z_h = \cdot \,|\, \theta_{\mathrm{in}}, \tau_h)\big) \geq \lambda_{\mathrm{I}}.$$

This assumption specifies the separation requirements for θ^*_{se} and θ^*_{in} individually. Based on this assumption, we establish the statistical error of the SI estimator as follows.

Corollary 25 (Sample Complexity of Selection-Inference) *Consider SI prompting with n examples whose distribution is given by (14) with a given task $\theta^*(\theta^*_{\mathrm{se}}, \theta^*_{\mathrm{in}})$. We assume that Θ is a finite set and the LLM is perfectly pretrained with data according to the model in (14) with a prior distribution π. Under Assumption 24, we have*

$$\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}_{\mathrm{SI}}\big(y^{\mathrm{test}} = \cdot \,|\, S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}}\big)\Big) \leq \mathcal{O}\big(\pi(\theta^*)^{-1} \delta^{-2} \big|\Theta^{\complement}\big|^2 e^{-2\lambda_{\mathrm{SI}} n}\big),$$

with probability $1 - \delta$, where $\lambda_{\mathrm{SI}} = \lambda_{\mathrm{S}} + \lambda_{\mathrm{I}} + \lambda_{\mathrm{q}}$. Here $\mathcal{O}(\cdot)$ hides absolute constants and $\mathbb{P}_{\mathrm{SI}}(y^{\mathrm{test}} \,|\, S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}})$ is the marginal distribution of y^{test} according to (16).

This corollary shows that the prompting error of the SI decays to zero exponentially fast as n goes to infinity. Moreover, here the exponential factor depends on λ_{SI}, which contains the separation of both the selection and inference parts. We defer the proof to Appendix F.3.

In summary, in this part, we extend the statistical analysis of the vanilla CoT estimator to three variants of CoT — SC-CoT, ToT, and SI. We interpret these prompting methods as statistical estimators, and establish their statistical errors under an ideal case where the pretraining error of the LLM is zero. The analysis can be easily extended to the realistic case with a nonzero pretraining error, which is separately discussed in Section 6.

8. Experiments

In this section, we provide empirical evidence for two key theoretical predictions. First, Theorem 7 establishes that the prompting error decays exponentially with the number of in-context demonstrations n; to verify this, we plot the *logarithm* of the MSE against n,

where a log-linear trend is direct evidence of exponential decay. Second, Proposition 11 shows that CoT dominates vanilla ICL on average but not necessarily instance-wise; the experiments illustrate this by comparing CoT variants with different intermediate steps and by observing that the mean MSE of vanilla ICL is larger than that of CoT, while the confidence intervals overlap. To this end, we train transformer models from scratch, where the training data is sampled from random regression tasks satisfying (2). The pretrained transformer is then tested on a new task via prompting. We present the experiment results as follows.

8.1 Experiment Settings

We consider the **in-context regression problems** (Garg et al., 2022) where the goal is to learn a class of functions $\mathcal{F} = \{f(\cdot;\theta), \theta \in \Theta)\}$ via prompting a pretrained transformer. Here, $f(\cdot;\theta)$ is a function with parameter θ. We consider two types of $\mathcal{F}$ – two-layer neural networks (NNs) and decision trees. For both cases, for any θ, we generate input-output examples of form $(x, f(x,\theta))$, together with a single intermediate step, i.e., $H = 2$. Here we assume $x \sim \mathcal{N}(0, I_{d_{\text{in}}})$ where $d_{\text{in}} = 10$ for two-layer NNs and $d_{\text{in}} = 20$ for decision trees. In other words, each CoT example is of the form $\{x, z, f(x;\theta)\}$ and each ICL example is of the form $\{x, f(x;\theta)\}$, where z is the intermediate step.

Two-Layer Neural Networks. For a two-layer NN, we write $\theta = \{W, v\}$ where $v \in \mathbb{R}^4$ and $W \in \mathbb{R}^{10\times 4}$. Under the prior distribution, W and v are independent, with $v \sim \mathcal{N}(0, I_4)$ and $W_{ij} \overset{i.i.d}{\sim} N(0, 1/2)$ for all $i \in [10]$ and $j \in [4]$. The neural network output is $f(x;\theta) = v^T \texttt{ReLU}(Wx)$, where $\texttt{ReLU}(x) = \max(0, x)$ is the ReLU activation function. We consider two kinds of CoT examples, $\text{CoT}_1 : \{x, \texttt{ReLU}(Wx), f(x;\theta)\}$ and $\text{CoT}_2 : \{x, v, f(x;\theta)\}$.

Decision Tree. We let $\theta = \{c_j, o_j\}_{j\in[15]}$ denote the parameters of a binary decision tree of depth four, where $c_j \sim \text{Unif}([20])$ and $o_j \overset{\text{i.i.d}}{\sim} \mathcal{N}(0, 1)$. The output $f(x;\theta)$ with input $x \in \mathbb{R}^{20}$ is defined as follows. Let c_1 correspond to the root node, c_2, c_3 be nodes of the second layer, $c_4, \ldots, c_7$ be nodes of the third layer, and $c_8, \ldots, c_{15}$ be nodes of the last layer. Each c_j indexes a coordinate of x and o_j is the corresponding target value. Note that $d_{\text{in}} = 20$, and thus we sample each c_j uniformly over $[20]$. To evaluate the decision tree, starting from the root node c_1, if $x[c_1] < 0$, we go to the left child c_2. Otherwise, we go to the right child c_3. Here we let $x[i]$ denote the i-th coordinate of x. Then we continue to look at the sign of $x[c_2]$ or $x[c_3]$ and go to a child node in the third layer. We continue this process until a leaf node, i.e., a node in the last layer, is reached, and we output the corresponding o_j. In other words, at each level, we look at the sign of the corresponding coordinate of the input x and move to a child. The output $f(x,\theta)$ corresponds to the number in $\{o_8, \ldots, o_{15}\}$ corresponding to the leaf node that is reached. The intermediate reasoning steps correspond to the four entries of x used to make decisions. Thus, a CoT example is of the form $\{x, x[\texttt{node}_1], \ldots, x[\texttt{node}_4], f(x,\theta)\}$, where $\texttt{node}_1, \ldots, \texttt{node}_4$ are the four nodes of the selected path, including the root and leaf nodes.

Transformer Model. We train a decoder-only transformer from the GPT-2 family (Radford et al., 2019) separately for CoT and ICL. We construct pretraining datasets following the setting in Section 6.1. That is, we sample N i.i.d. tasks $\{\theta^*_\ell\}_{\ell\in[N]}$ and T examples from each task, where $T = 101$ and $N = 2.56 \times 10^7$. Then we build a loss function similar to

that in (3), but with the negative loglikelihood replaced by the mean-squared error. The loss function is optimized using the Adam algorithm (Kingma and Ba, 2014) for 4×10^5 steps, with the batch size set to 64. We let $\texttt{TF}_{\rm CoT}$ and $\texttt{TF}_{\rm ICL}$ denote the learned transformer model.

Evaluation. To evaluate the performances, we sample a random task θ^* and n i.i.d. examples $\{x_i, f(x_i;\theta^*)\}_{i\in[n]}$, and ask the pretrained transformers to predict $f(x^{\rm test};\theta^*)$ on a new i.i.d. input $x^{\rm test}$. We include intermediate steps to these n examples to obtain CoT prompt. We evaluate the performance of CoT and ICL via the mean-squared error (MSE):

$$\texttt{MSE}_{\rm CoT} = \big[\texttt{TF}_{\rm CoT}\big(\texttt{prompt}_{\rm CoT}(n) \cup \{\widehat{z}\}\big) - f(x^{\rm test};\theta^*)\big]^2,$$
$$\texttt{MSE}_{\rm ICL} = \big[\texttt{TF}_{\rm ICL}\big(\texttt{prompt}_{\rm ICL}(n)\big) - f(x^{\rm test};\theta^*)\big]^2,$$

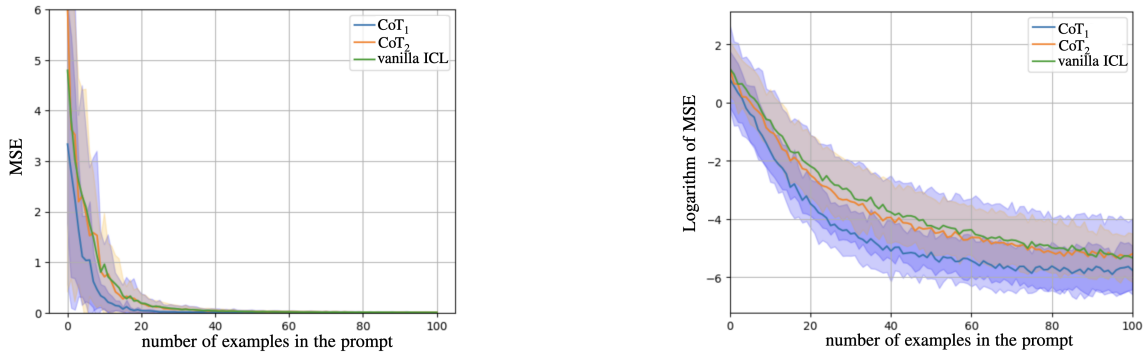
where $\widehat{z} = \texttt{TF}_{\rm CoT}\big(\texttt{prompt}_{\rm CoT}(n)\big)$ is the intermediate step predicted by the transformer on the test example. We compute the MSE by averaging over 100 independent experiments with a fixed θ^* for each n, and we test for all n in $\{1,\ldots,100\}$. We plot the final MSE by further averaging over 10 independent θ^*'s.

8.2 Experiment Results

In the following, we present the results for two-layer neural networks (NNs) and decision trees.

Two-Layer Neural Network. We plot the errors of the CoT and ICL estimators in Figure 10 against the number of demonstration examples $n \in [100]$. The curves with labels "CoT_1" and "CoT_2" are the transformer trained using the two kinds of CoT prompts, respectively, and the label " vanilla ICL" stands for the error of the vanilla ICL. In Figure 10-(a) we plot the MSE of these methods. In all these three cases, MSE decays rapidly to zero as n increases. Moreover, we observe that CoT_1 method with $z = \texttt{ReLU}(Wx)$ exhibits significant improvement over vanilla ICL. Moreover, CoT_2 only slightly improves upon vanilla ICL, which shows that the quality of intermediate steps is crucial to the success of CoT. This finding coincides with our theory and experiment in Section 5.2. A plausible explanation for the superiority of CoT_1 over CoT_2 is that the major challenge of learning a two-layer NN lies in learning the nonlinear feature $\texttt{ReLU}(Wx)$. Providing this piece of information in the prompt significantly simplifies the learning problem. Furthermore, in Figure 10-(b) we plot the logarithm of the MSE versus n. As seen in the figure, there is a linear trend for all three methods when n is smaller than 20. When n exceeds 20, MSE is very close to zero. In this case, the pretraining error is not negligible and thus the linear trend stops. Thus, Figure (b) shows that the statistical error of these methods decays exponentially in n up to a pretraining error. This observation corroborates Theorem 7.

Decision Tree. We plot the errors of CoT and vanilla ICL in Figure 11, with (a) and (b) showing the MSE and its logarithm respectively. In both cases, MSE decays rapidly as n increases and there is a strong linear trend when $n \le 65$. This aligns with our theory in Theorem 7, and the MSE after $n \ge 65$ is close to the pretraining error. Moreover, we observe that the transformer trained via the CoT method learns faster than that trained by vanilla ICL, but they achieve similar accuracy when n becomes large. When given $n = 100$ in-context demonstrations, both models give a testing loss of around 0.12.



(a) MSE (b) Logarithm of MSE

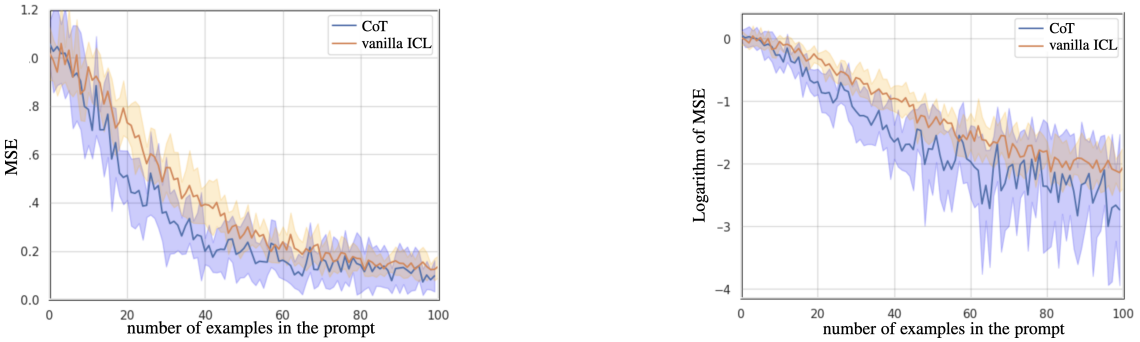
Figure 10: Statistical errors of vanilla ICL and two CoT estimators on the task of learning 2-layer MLPs. The estimators are constructed by prompting the pretrained transformer models with n demonstration examples. In Figure (a), the MSE of the three estimators decreases rapidly as n increases. In Figure (b) we plot the logarithm of MSE, which follows a strong linear trend for $n \leq 20$. Beyond $n = 20$, the MSE is dominated by the pretraining error and therefore the linear trend stops. Furthermore, the model trained with "CoT_1", using intermediate steps $z = \texttt{ReLU}(Wx)$, shows a significant improvement over vanilla ICL, while the "CoT_2" model, using $z' = v$, only shows a slight improvement. In both (a) and (b), we also plot the standard deviation computed based on 10 random experiments.

In summary, our experiments on the two-layer neural network and decision tree tasks validate the statistical error of the "pretraining LLM + CoT prompting" approach. We empirically validate the exponential error decay with respect to the number of prompt examples, and we show whether CoT significantly outperforms vanilla ICL depending on the choice of intermediate reasoning steps. In both tasks, the mean MSE of vanilla ICL is higher than that of CoT, yet the confidence intervals overlap, consistent with Proposition 11: CoT dominates vanilla ICL on average but not necessarily instance-wise. We leave further details of the experiment setup in Section 8.3.

8.3 Additional Details of Numerical Experiments

In the following, we present the details of the numerical experiments in Section 8.

Training Data and Algorithm. The transformer models are pretrained using the Adam algorithm Kingma and Ba (2014), which is a minibatch and stochastic-gradient-based algorithm. We set the batch size to 64, and in each step, we sample new training data from the model in (2). That is, we sample 64 random tasks from the prior distribution, and $T = 101$ examples from each task. In each task, we pack the T examples into a single trajectory, and build the MSE loss function by predicting the next step autoregressively. For vanilla ICL, there are $2T$ steps in total, and for CoT, there are $3T$ steps in total. We run Adam for 4×10^5 steps in total, and thus the total tasks sampled is equal to $N = 2.56 \times 10^7$.



(a) MSE (b) Logarithm of MSE

Figure 11: Statistical errors of vanilla ICL and two CoT estimators on the task of learning decision trees. The estimators are constructed by prompting the pretrained transformer models with n demonstration examples. In (a), we plot the mean-squared errors of these two estimators against the number of demonstrations. The error decreases faster with in-context examples for models trained with CoT data compared to those trained with vanilla ICL. In Figure (b) we plot the logarithm of the MSE, which reveals a linear trend in errors for both models when $n \leq 65$.

Moreover, when implementing Adam, we set the learning rate (stepsize) to be 10^{-5} and the momentum parameter to be $(0.9, 0.999)$.

GPT-2 Transformer Model. We adopt the GPT-2 transformer architecture. Here the transformer model reads in a sequence of input vectors in $\mathbb{R}^r$ and produces an output vector in the same space, where r is the embedding dimension. Additionally, to handle the vector-valued inputs with different sizes, we adopt a universal read-in function that maps the prompts into the latent embedding space of the transformer through a (learnable) linear transformation and we use separate read-out functions for the predictions of inputs $x \in \mathbb{R}^{d_{\text{in}}}$, intermediate steps $z \in \mathbb{R}^4$, and outputs $y \in \mathbb{R}$. Here $d_{\text{in}} = 10$ for two-layer NNs and $d_{\text{in}} = 20$ for decision trees. These read-in and read-out functions are all linear layers. Between these read-in and read-out functions are multiple transformer blocks stacked vertically. The details of these transformer blocks are introduced in Section 6.1 and Appendix I.1. See Figure 12 for an illustration of the transformer architecture.

For the two-layer NN task, we adopt GPT-2 transformer models with $D = 12$ layers, $\eta = 4$ heads, and an embedding dimension of $r = 128$. The embedding dimension of the queries, keys, and values is 32. For the decision tree task, we adopt GPT-2 transformer models consisting of $D = 12$ layers, $\eta = 8$ heads, and an embedding dimension of $r = 512$. The embedding dimension of the queries, keys, and values is 32.

9. Conclusion

In this paper, we explore the theoretical underpinnings of CoT prompting and its variants through a statistical lens. Under a latent variable model that depicts multi-step reasoning,

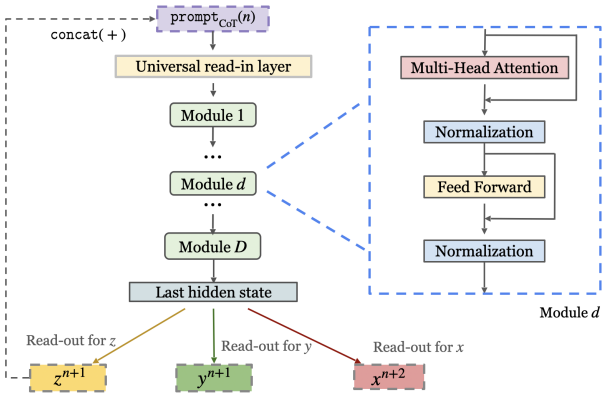


Figure 12: An illustration of the GPT-2 based transformer model. We begin by embedding the input prompt using a universal read-in function. The embedding is then processed by our backbone model, which is based on the GPT-2 architecture. We extract the last hidden state and apply separate read-out functions for inputs x, intermediate steps z, and outputs y. During testing, we concatenate the inferred z^{test} with $\texttt{prompt}_{\text{CoT}}(n)$ to infer y^{test}.

we showed the estimators induced by CoT prompting on a pretrained LLM are approximately equivalent to a Bayesian estimator. More importantly, we prove that the statistical error of CoT can be upper bounded by a sum of pretraining error and prompting error, and we explicitly analyze them separately. Moreover, we establish both theoretical and empirical comparisons between CoT and vanilla ICL, which shed new light on the role played by intermediate reasoning steps.

While our analysis primarily targets standard short CoT with fixed length H, our framework offers valuable insights into the long CoT regime. We distinguish these settings by *adaptivity* rather than mere length: short CoT uses fixed steps for straightforward tasks, while long CoT is characterized by reasoning chains that scale adaptively with intrinsic problem complexity. For example, consider the contrast between elementary arithmetic and IMO problems. We provide concrete examples in Appendix L. Our theoretical analysis provides the following two insights:

- **The Benefit of Longer Chains (Inference)**: Our framework offers a theoretical justification for the empirical success of long CoT on complex tasks. As formalized in Corollary 32, longer reasoning chains yield a richer signal per demonstration, which always reduces prediction error in expectation. From a Bayesian perspective, the CoT estimator performs Bayesian Model Averaging (BMA): additional intermediate steps act as further observations that help the model infer the latent task variable θ^*. For complex problems, such as Olympiad-level mathematics, these extended traces—including backtracks and verifications—concentrate the posterior distribution over θ^*, leading to higher accuracy.

- **The Cost of Longer Chains (Pretraining)**: However, this performance gain comes with a necessary trade-off in computational efficiency. Proposition 14 reveals that the resources required for pretraining scale with the chain length H. Specifically, to achieve a target accuracy, the transformer depth D must scale linearly with H to handle sequential dependencies, while the number of pretraining documents N scales polynomially with H. This highlights the substantial computational price of long CoT capabilities: while longer chains decrease inference error, they demand deeper networks and significantly larger pretraining datasets.

In future work, we aim to extend our theoretical framework to better understand prompting methods beyond the CoT family, analyzing both their effectiveness and potential pitfalls, and establishing more effective reasoning techniques. Designing controlled experiments that quantitatively test the dependence on the separation parameter λ, the network depth D, and the pretraining data size N (as predicted by Corollary 18) would also be a valuable empirical complement to the present theoretical analysis.

Acknowledgments

The author Zhuoran Yang gratefully acknowledges the support of NSF under the award DMS-2413243. The authors declare no competing interests.

Appendix A. More Related Works

Our work adds to the literature on theoretically understanding prompting methods. In particular, our work is closely related to CoT prompting and its variants. In addition, our work is related to the body of works that aim to understand the ability of ICL and CoT from both empirical and theoretical perspectives.

CoT Prompting and its Variants. The vanilla CoT prompting method is proposed in Wei et al. (2022) for solving multi-step reasoning problems using LLMs. Based on this work, many variants of CoT have been proposed to enhance the efficiency and reliability of LLMs in solving multi-step reasoning problems. See, e.g., Yao et al. (2023); Wang et al. (2022); Creswell et al. (2022); Zhou et al. (2022); Chen et al. (2022); Zhang et al. (2023c); Besta et al. (2024) and also see Chu et al. (2023); Zhang et al. (2023b) for recent surveys of CoT methods. In particular, our work offers a theoretical understanding for vanilla CoT and variants including SC CoT (Wang et al., 2022), SI Creswell et al. (2022), and ToT (Yao et al., 2023).

More Existing Research on Understanding ICL. Our work is closely related to the body of works that aim to understand the ability of ICL from both empirical and theoretical perspectives. From an empirical point of view, Garg et al. (2022); Min et al. (2022); Krishnamurthy et al. (2024); Zhang et al. (2022b); Dziri et al. (2024); Olsson et al. (2022) explore the understanding of the behavior and capability of ICL. In particular, Garg et al. (2022) show that transformers can learn unseen linear functions via ICL. Min et al. (2022) demonstrate that shuffled input-output pairs in few-shot ICL induce little degradation in the performance on a range of classification and multi-choice tasks. Dziri et al. (2024) study how transformer-based LLMs solve compositional tasks and their limitations in reasoning.

From a theoretical perspective, Akyürek et al. (2022); Von Oswald et al. (2023); Bai et al. (2023); Dai et al. (2023); Wang et al. (2023b) establish theoretical understandings of ICL. In addition to the Bayesian interpretation of ICL discussed in the main literature review, another line of theory suggests that LLMs perform ICL by implicitly executing iterative optimization algorithms, such as gradient descent. The works Akyürek et al. (2022); Von Oswald et al. (2023); Bai et al. (2023); Dai et al. (2023) indicate that ICL implicitly implements the gradient descent or least-square algorithms from the function approximation perspective. Hou et al. (2023) hypothesize that LLMs implicitly perform multi-step reasoning within their architecture by going through a reasoning tree. Li et al. (2023a) derive the generalization bound for ICL from the view of multi-task learning. Hahn and Goyal (2023) adopt a linguistic point of view and bounds the ICL error using description length. The works Ahn et al. (2023); Huang et al. (2023b); Fu et al. (2023); Mahankali et al. (2023); Wu et al. (2023a) consider linear attention models to study the performance of ICL, which restricts the function class that can be represented by transformers to linear functions.

Appendix B. More Background

Transformers and Attention Mechanism. The transformer model is based on the MHA mechanism (Bahdanau et al., 2014; Phuong and Hutter, 2022), together with other modules such as the tokenizer and the positional embeddings (Wang and Chen, 2020; Su et al., 2023), residual connections, feed-forward networks, and layer normalization (Ba et al., 2016). The tokenizer maps the input sequence to a sequence of vectors in Euclidean space, and the positional embeddings add the position information of tokens to these vectors.

The attention mechanism captures the relationship between different tokens, which is the backbone of transformer-based LLM (Devlin et al., 2018). The attention mechanism takes in queries, keys, and values as inputs, and outputs the response of each query as a weighted sum of values, where the weights are the similarity scores between the query and the keys. Specifically, let $K \in \mathbb{R}^{L\times d_k}$ and $V \in \mathbb{R}^{L\times d_v}$ denote the L key and value vectors, respectively. The attention output of a single query $q \in \mathbb{R}^{d_k}$ is computed as:

$$\texttt{attn}(q, K, V) = V^T \texttt{softmax}(Kq), \tag{17}$$

where $\texttt{softmax}(Kq)$ is a probability distribution over $[L]$. Here $\texttt{softmax}(Kq)$ quantifies the similarity between the query q and each row of K, which is used to aggregate the value vectors of V. The attention $\texttt{attn}(Q, K, V)$ that takes in multiple queries outputs the responses as $V^T\texttt{softmax}(KQ^T)$, where $Q \in \mathbb{R}^{d_k}$ contains L query vectors.

The predefined attention mechanism captures the relationship between the keys and queries via a single softmax module, and thus is called single-head attention. MHA refers to passing the inputs through multiple attention functions in parallel, and outputs the aggregation of these sub-modules. Taking $X \in \mathbb{R}^{L\times r}$ as the input, a MHA layer with η heads outputs

$$\texttt{mha}(X, W_{\text{mha}}) = \sum_{i=1}^{\eta} \texttt{attn}(XW_i^Q, XW_i^K, XW_i^V), \tag{18}$$

The parameter set $W_{\text{mha}} = \{W_i^Q, W_i^K, W_i^V\}_{i=1}^{\eta}$ are the weight matrices for queries, keys, and values, where $W_i^Q \in \mathbb{R}^{r\times d_k}$, $W_i^K \in \mathbb{R}^{r\times d_k}$, and $W_i^V \in \mathbb{R}^{r\times d_v}$. Intuitively, different heads can attend to different parts of the data, and thus MHA offers a more expressive model class. Compared to the MHA defined in Vaswani et al. (2017), we absorb the matrix W_i^O into W_i^V for each head (Zhang et al., 2022a).

Each MHA layer is followed by a FF layer. Given an input $X \in \mathbb{R}^{L\times r}$, a FF layer with d_F neurons maps the input X to

$$\texttt{ff}(X, W_{\text{ff}}) = \texttt{ReLU}(XW_{\text{ff},1})W_{\text{ff},2}, \qquad \text{where} \quad W_{\text{ff}} = \{W_{\text{ff},1} \in \mathbb{R}^{r\times d_F}, W_{\text{ff},2} \in \mathbb{R}^{d_F\times r}\} \tag{19}$$

are weight matrices. There are also normalization layers between the MHA and FF layers. We defer their details to Appendix I.1 for brevity.

Variants of CoT. As conditional probability models, LLMs are intrinsically *stochastic*. For problems such as solving mathematical questions, however, there is often a unique answer. To further boost the probability of finding the correct answer, variants of CoT leverage multi-step reasoning with various selection techniques to solve more complicated reasoning and decision-making problems. For instance, SC-CoT (Wang et al., 2022) uses majority

vote, ToT (Yao et al., 2023) adopts tree search methods, and SI (Creswell et al., 2022) further introduces a selection module in each reasoning step.

Notations. Let $[T]$ denote $\{1, \cdots, T\}$. We use V_i to denote the i-th coordinate of the vector V. We adopt $\{\mathbb{P}_\theta \,|\, \theta \in \Theta\}$ to denote a parametric family of distributions parameterized by $\theta \in \Theta$. The Kullback-Leibler (KL) divergence between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is denoted by

$$\texttt{KL}(\mathbb{P}, \mathbb{Q}) = \mathbb{E}_{\mathbb{P}(x)}\bigg[\log \frac{p(x)}{q(x)}\bigg],$$

where p and q are the densities of $\mathbb{P}$ and $\mathbb{Q}$ with respect to a reference distribution. Furthermore, we define the conditional KL divergence as

$$\texttt{KL}\big(\mathbb{P}(Y = \cdot \,|\, X = x), \mathbb{Q}(Y = \cdot \,|\, X = x)\big) = \mathbb{E}_{y \sim \mathbb{P}(\cdot \,|\, x)}\bigg[\log \frac{p(y \,|\, x)}{q(y \,|\, x)}\bigg],$$

which is a function of x.

Let $\mathbb{P}$ and $\mathbb{Q}$ denote two probability measures defined over a measurable space $(\Omega, \mathcal{F})$. The total variation (TV) distance between $\mathbb{P}$ and $\mathbb{Q}$ is

$$\texttt{TV}(\mathbb{P}, \mathbb{Q}) = \sup_{A \in \mathcal{F}} |\mathbb{P}(A) - \mathbb{Q}(A)|,$$

which is also half of the ℓ_1 distance between density function $p(x)$ and $q(x)$.

The Hellinger distance between two distributions $\mathbb{P}$ and $\mathbb{Q}$ is

$$\mathrm{H}(\mathbb{P}, \mathbb{Q}) = \frac{1}{\sqrt{2}}\bigg(\int \big(\sqrt{p(x)} - \sqrt{q(x)}\big)^2 \mathrm{d}x\bigg)^{1/2} = \bigg(1 - \int \sqrt{p(x)q(x)}\mathrm{d}x\bigg)^{1/2}.$$

In addition, we use $\texttt{softmax}(\cdot)$ to denote the softmax function, which maps a vector to a probability distribution. In particular, for any vector $x \in \mathbb{R}^n$ and any $i \in [n]$, the i-th entry of $\texttt{softmax}(x)$ is given by $[\texttt{softmax}(x)]_i = \exp(x_i)/\sum_{j=1}^n \exp(x_j)$. For a matrix $X \in \mathbb{R}^{d_1 \times d_2}$, we denote the i-th row and column using $X_{i:}$ and $X_{:,i}$, respectively. The $\ell_{p,q}$ norm of X is defined as $\|X\|_{p,q} = (\sum_{i=1}^{d_2} \|X_{:,i}\|_p^q)^{1/q}$. Furthermore, we use $\|X\|_{\mathrm{F}} = \|X\|_{2,2}$ to denote Frobenius norm. For a set $\mathcal{L}$, let $\mathcal{P}(\mathcal{L})$ denote the family of probability distributions over $\mathcal{L}$. We use $z_{1:h}$ to denote the vector $[z_1, \cdots, z_h]$. We adopt $\mathcal{L}^*$ to denote the set of all the sequences where each component is in the set $\mathcal{L}$.

Appendix C. A Generalized Multi-Step Latent Variable Model

In this section, we propose a **generalized multi-step latent-variable model** that removes the i.i.d requirement in the model defined in (2). This generalized model captures **(i)** the evolving relationships among examples, and **(ii)** the multi-step reasoning framework of CoT. The rationale behind **(i)** is that LLMs are pretrained on trillions of reasoning steps from documents and articles from the internet (Achiam et al., 2023). The examples in the pretraining data associated with the same task are often **not i.i.d.** For example, imagine composing a sequence of examples of the concept of "animals", we typically begin with familiar examples such as cats and dogs before progressing to less conventional ones like panthers and meerkats.

To capture both **(i)** and **(ii)**, we propose to model the joint distribution of the latent variable $\theta^* \in \Theta$, the n examples $\{s^j\}_{j=1}^n$, and the test instance $z_{0:H}^{\text{test}}$ as follows:

$$\begin{aligned} &\theta^* \sim \pi, && z_0^i = f_{\theta^*}\big(\{s^j\}_{j=1}^{i-1}, \zeta_i\big), \\ &u_h^i = g_{\theta^*}\big(\{u_k^j\}_{j=1,k=1}^{i-1,H}, u_1^i, \cdots, u_{h-1}^i, \xi_h^i\big), && z_h^i = F(z_0^i, \cdots, z_h^i, u_h^i, \epsilon_h^i), \qquad \forall h \in [H]. \end{aligned} \tag{20}$$

Here f_{θ^*} and g_{θ^*} are functions depending on θ^*, ζ_i, ξ_h^i, and ϵ_h^i are independent noise terms, and F is a function that does not depend on θ^*. The key assumption of (20) is that each reasoning step z_h^i depends on the task parameter θ^* only through a latent variable u_h^i, and these latent variables $\{u_h^i\}_{h=1,i=1}^{H,n}$ form a dynamical system in the latent space. The evolution of u_h^i depends on θ^* and all the latent variables of the previous examples $\{s^j\}_{j<i}$. Thus, each z_h^i is allowed to implicitly depend on the previous examples as well. Such a latent dynamical system captures the fact that the demonstration examples are created with a dependent structure. In comparison, in the simpler model in (2), the examples are i.i.d. given θ^*. We recover the simpler model in (2) by setting $u_h^i = \theta^*$ and $z_0^i = f_{\theta^*}\big(\zeta_i\big)$ in (20).

Intuitively, θ^* represents the latent concept specifying the task, such as "calculate twice the area code of the given country," "solve an arithmetic problem," or "write a science fiction novel." In (20), we model LLM-based reasoning as a hierarchical process: first, we generate a sequence of task-specific latent "goals" $\{u_h^i\}_{h\in[H]}$, then we translate these goals into natural language $\{z_h^i\}_{h\in[H]}$. Each goal corresponds to a specific reasoning step. The sequence of latent variables completely determines the reasoning process, and thus we assume F in (20) does not involve θ^*. See Figure 13 for an illustration of this model.

As a concrete example, consider θ^* as the task of "solving an arithmetic problem". The problem might be described in a specific context, such as using the number of apples. However, the underlying reasoning process is independent of this context and relies solely on a sequence of arithmetic operations. These operations can be viewed as the latent variables u_h^i in our model, while describing them in the context of apples can be seen as generating the natural language z_h^i from these latent variables. Additionally, the random variables ζ^i, ϵ_h^i (for $h \in [H]$) determine how the arithmetic formula is contextualized. The random variables ξ_h^i introduce stochasticity into the reasoning process. See Figure 14 for an illustration.

We remark that our model is a general formulation that recovers many existing models proposed in the existing works. Specifically, we recover the models in Jiang (2023) by setting

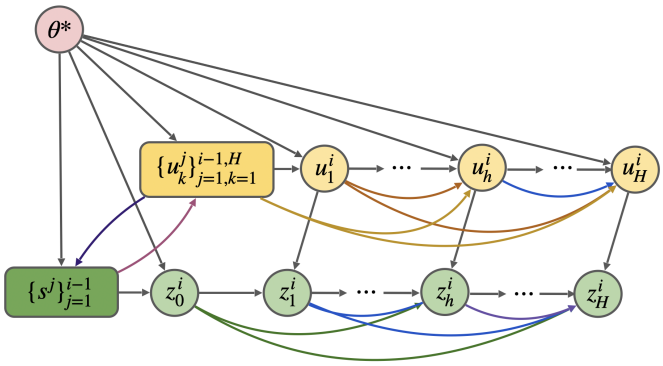


Figure 13: An illustration of the generalized multi-step latent-variable model (20) with the fixed task θ^*. Compared to the simple model (2), this model allows each step z_j^i of i-th example to depend on earlier examples via u_j^i, which depends on all latent variables previously generated. Therefore, the examples $\{s^i\}_{i=1}^n$ do not have to be i.i.d.

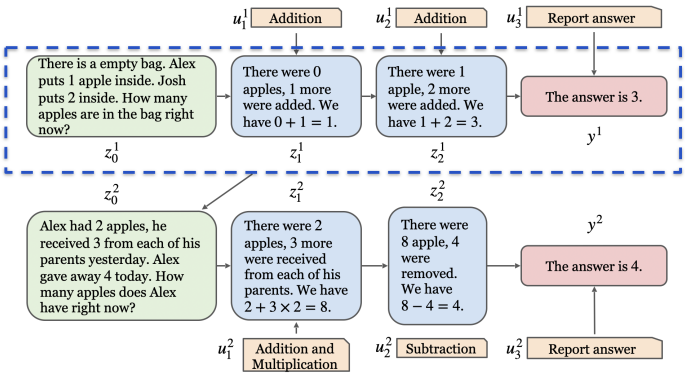


Figure 14: An example of 2 demonstrations generated from the generalized model 20, where we omit some edges in the graph to simplify the representation. In this example, we want to highlight two features of our model. The first and second rows correspond to two demonstrations. Note that the reasoning steps $\{z_h^i\}$ depend on latent concept θ^* implicitly via latent variables $\{u_h^i\}$, which give specific instructions for each step in the corresponding example. Furthermore, the latent variables $u_{1:3}^1$ are different from $u_{1:3}^2$, providing more flexibility and diversity among the demonstrations. In addition, the second query z_0^2 is not independent of the previous demonstration $z_{0:3}^1$ since it is a more complicated version of an arithmetic problem.

the latent variables $(u_1^i, \cdots, u_H^i)$ directly as the latent variable vector θ^*. Besides, we cover the models studied in Zhang et al. (2023a); Wang et al. (2023c) by setting $H = 1$.

Finally, some of our results extend to the generalized model 20. Specifically, Lemma 26 demonstrates that LLMs perform BMA during CoT under this generalized model. In Section 6, we describe the pretraining process and analyze its performance within the framework of the generalized model (20).

Appendix D. Proofs of the Results in Section 4

In this section, we prove the results in Section 4. We first prove Lemma 1 and its extension to the generalized multi-step latent variable model and then prove Proposition 2.

D.1 Proof of Lemma 1 and Its Extension

We prove Lemma 1 and introduce its extension to the generalized multi-step latent variable in (20) with a proof.

Proof [Proof of Lemma 1] When N goes to infinity, we only need to consider the population counterpart of the likelihood loss in (3), which can be written as

$$\begin{aligned}\mathcal{L}(\rho) &= -\frac{1}{T(H+1)}\sum_{t\in[T]}\sum_{h=0}^{H}\mathbb{E}\left[\log \mathbb{P}_\rho\big(z_h^t \,|\, \Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1}\big)\right] \qquad (21)\\ &= \frac{1}{T(H+1)}\sum_{t\in[T]}\sum_{h=0}^{H}\mathbb{E}\Big[\mathrm{KL}\big(\mathbb{P}\big(\cdot \,|\, \Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1}\big) \,\|\, \mathbb{P}_\rho\big(\cdot \,|\, \Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1}\big)\big)\Big] + \texttt{Const},\end{aligned}$$

where $\texttt{Const}$ denotes a constant that does not depend on ρ. Here the second equality follows from the definition of KL divergence. When the LLM class is sufficiently expressive, i.e., $\mathbb{P} \in \{\mathbb{P}_\rho \,|\, \rho \in \mathcal{P}_{\mathrm{LLM}}\}$, for any $(\Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1})$ that has nonzero density in the pretraining distribution, the minimizer of $\mathcal{L}(\rho)$ in (21) must satisfy

$$\mathbb{P}_{\mathrm{LLM}}\big(z_h^t = \cdot \,|\, \Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1}\big) = \mathbb{P}\big(z_h^t = \cdot \,|\, \Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1}\big) \qquad (22)$$

for all $h \in [H]$ and $t \in [T]$, where we let $\mathbb{P}_{\mathrm{LLM}}$ denote the distribution induced by the minimizer of $\mathcal{L}(\cdot)$. Here (22) holds for all prompts $\Upsilon_{t-1} \cup \{z_j^t\}_{j=0}^{h-1} \in \mathcal{L}^*$ with a nonzero density under $\mathbb{P}$. Thus, (22) shows that the perfectly pretrained LLM matches the pretraining distribution of predicting the *next reasoning step*.

In the following, we will prove the desired result by generalizing (22) to the next multi-step reasoning. Now, when we generate the answer y^{test} given the CoT prompt using $\mathbb{P}_{\mathrm{LLM}}$, when $\texttt{prompt}_{\mathrm{CoT}}(n)$ has a positive density under $\mathbb{P}$, by (22) we have

$$\mathbb{P}_{\mathrm{LLM}}\big(z_1^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) = \mathbb{P}\big(z_1^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big). \qquad (23)$$

Let $\widetilde{\mathcal{L}}_1 \subseteq \mathcal{L}$ denote the subset of reasoning steps such that the conditional distribution in (23) is positive when $z_1^{\mathrm{test}} \in \widetilde{\mathcal{L}}_1$, and let $\widetilde{\mathcal{L}}_1^{\mathtt{C}}$ denote its complement. Since $\mathbb{P}\big(\texttt{prompt}_{\mathrm{CoT}}(n)\big) > 0$, we have

$$\mathbb{P}\big(\texttt{prompt}_{\mathrm{CoT}}(n) \cup \{z_1^{\mathrm{test}} = z_1\}\big) > 0, \qquad \forall z_1 \in \widetilde{\mathcal{L}}_1.$$

Then using (22), since $\texttt{prompt}_{\mathrm{CoT}}(n) \cup \{z_1^{\mathrm{test}} = z_1\}$ has positive density under $\mathbb{P}$, we have

$$\mathbb{P}_{\mathrm{LLM}}\big(z_2^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1\big) = \mathbb{P}\big(z_2^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1\big), \forall z_1 \in \widetilde{\mathcal{L}}_1.$$

Moreover, we have $\mathbb{P}_{\mathrm{LLM}}(z_1^{\mathrm{test}} = z_1 \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)) = 0$ for all $z_1 \in \widetilde{\mathcal{L}}_1^{\mathfrak{C}}$. Therefore, by direct computation, we have

$$\begin{aligned}
&\mathbb{P}_{\mathrm{LLM}}\big(z_2^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \\
&\quad = \int_{\widetilde{\mathcal{L}}_1} \mathbb{P}_{\mathrm{LLM}}\big(z_2^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1\big) \cdot \mathbb{P}_{\mathrm{LLM}}\big(z_1^{\mathrm{test}} = z_1 \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \mathrm{d}z_1 \\
&\quad = \int_{\widetilde{\mathcal{L}}_1} \mathbb{P}\big(z_2^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1\big) \cdot \mathbb{P}\big(z_1^{\mathrm{test}} = z_1 \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \mathrm{d}z_1 \\
&\quad = \mathbb{P}\big(z_2^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big).
\end{aligned}$$

We can generalize this argument to $z_H^{\mathrm{test}} = y^{\mathrm{test}}$. Specifically, let $\widetilde{\mathcal{L}}^*$ denote the reasoning steps $\{z_1^{\mathrm{test}}, \ldots, z_{H-1}^{\mathrm{test}}\}$ with positive density under $\mathbb{P}(\cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))$. Let each element in $\widetilde{\mathcal{L}}^*$ be denoted by $\mathbf{z} = \{z_1, \ldots, z_{H-1}\}$. Then we have

$$\begin{aligned}
&\mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \\
&\quad = \int_{\mathbf{z} \in \widetilde{\mathcal{L}}^*} \mathbb{P}\big(y^{\mathrm{test}} = \cdot, \{z_1^{\mathrm{test}}, \ldots, z_{H-1}^{\mathrm{test}}\} = \mathbf{z} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \, \mathrm{d}\mathbf{z} \\
&\quad = \int_{\mathbf{z} \in \widetilde{\mathcal{L}}^*} \prod_{h=1}^{H} \mathbb{P}\big(z_h^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}\big) \mathrm{d}z_1 \cdots \mathrm{d}z_{H-1}.
\end{aligned} \tag{24}$$

Here the second inequality follows from the factorization of joint probability into a product of conditional probabilities. Since $\mathbb{P}(\mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}) > 0$ by the definition of $\widetilde{\mathcal{L}}^*$, by (22) we have

$$\begin{aligned}
&\mathbb{P}\big(z_h^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}\big) \\
&\qquad = \mathbb{P}_{\mathrm{LLM}}\big(z_h^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}\big).
\end{aligned} \tag{25}$$

Therefore, this equality implies that $\mathbb{P}_{\mathrm{LLM}}$ cannot assign any positive density on the complement of $\widetilde{\mathcal{L}}^*$, otherwise (25) is violated. Thus, combining (24) and (25), we conclude that

$$\begin{aligned}
&\mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \\
&\quad = \int_{\mathbf{z} \in \widetilde{\mathcal{L}}^*} \prod_{h=1}^{H} \mathbb{P}\big(z_h^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}\big) \mathrm{d}z_1 \cdots \mathrm{d}z_{H-1} \\
&\quad = \int_{\mathbf{z} \in \widetilde{\mathcal{L}}^*} \prod_{h=1}^{H} \mathbb{P}_{\mathrm{LLM}}\big(z_h^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n), z_1^{\mathrm{test}} = z_1, \ldots, z_{h-1}^{\mathrm{test}} = z_{h-1}\big) \mathrm{d}z_1 \cdots \mathrm{d}z_{H-1} \\
&\quad = \int_{\mathbf{z} \in \widetilde{\mathcal{L}}^*} \mathbb{P}_{\mathrm{LLM}}\big(y^{\mathrm{test}} = \cdot, \{z_1^{\mathrm{test}}, \ldots, z_{H-1}^{\mathrm{test}}\} = \mathbf{z} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \mathrm{d}\mathbf{z} \\
&\quad = \mathbb{P}_{\mathrm{LLM}}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big).
\end{aligned}$$

Here the second equality follows from (25) and the last equality follows from the fact that $\mathbb{P}_{\mathrm{LLM}}$ is supported on $\widetilde{\mathcal{L}}^*$. Finally, by the Bayes' rule and the fact that y^{test} is independent

of the n examples in $\texttt{prompt}_{\mathrm{CoT}}(n)$ conditioning on θ, we have

$$\begin{aligned}\mathbb{P}_{\mathrm{LLM}}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) &= \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) \\ &= \int_{\Theta} \pi\big(\theta \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)\mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta\big)\mathrm{d}\theta.\end{aligned}$$

Therefore, we conclude the proof. ■

In the following, we extend Lemma to the generalized multi-step latent variable in (20) and present its proof.

Lemma 26 *With pretraining data sampled from the generalized model in* (20)*, we consider the population counterpart of the MLE in* (3) *with N going to infinity. Suppose that the CoT prompt $\texttt{prompt}_{\mathrm{CoT}}(n)$ has nonzero density under the pretraining distribution. Then the perfectly pretrained LLMs perform BMA during CoT prompting. Namely, we have that*

$$\mathbb{P}_{\mathrm{LLM}}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) = \int_{\Theta} \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n), \theta\big)\pi\big(\theta \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)\mathrm{d}\theta.$$

Proof When the pretraining dataset is sampled according to the model in (20), by Bayes's rule, we have

$$\mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) = \int_{\Theta} \pi\big(\theta \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)\mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n), \theta\big)\mathrm{d}\theta. \tag{26}$$

Note that the n examples and testing query z_0^{test} in $\texttt{prompt}_{\mathrm{CoT}}(n)$ are not i.i.d under the generalized model.

Since the MLE loss does not concern the underlying distribution, our analysis in the proof of Lemma 1 still holds. In particular, (22) holds on all prompts with a positive density under the pretraining distribution. As a result, starting from a CoT prompt $\texttt{prompt}_{\mathrm{CoT}}(n)$ with a positive density, we can show that

$$\mathbb{P}\big(z_1^{\mathrm{test}} = \cdot, \ldots, z_H^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) = \mathbb{P}_{\mathrm{LLM}}\big(z_1^{\mathrm{test}} = \cdot, \ldots, z_H^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)$$

by writing the joint probability as a product of H conditional probabilities. Therefore, we similarly have

$$\mathbb{P}_{\mathrm{LLM}}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) = \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)$$

when N goes to infinity. We conclude the proof by combining this fact with (26). ■

D.2 Proof of Proposition 2

Proof We aim to show that the attention mechanism approximates BMA, meaning that $\bar{v}_{\mathrm{test}}$ and $\texttt{attn}(q_h^{\mathrm{test}}, \texttt{keys}, \texttt{values})$ converge to the same limit as $n \to \infty$. To this end, we

show that both quantities converge to a population-level estimator based on population matrices. The proof is divided into three steps. First, we define the population-level estimator that bridges the BMA estimator and the attention estimator. Then, in the second and the last step, we prove that these estimators converge the population-level estimator as n goes to infinity. Our proof generalizes the proof in Zhang et al. (2023a) for vanilla ICL.

Step 1: Define the population-level estimator. Recall that we focus on a simplified model defined in (5), which involves a feature mappings ϕ, k, and v. Also note that we define $k_h^i = \big(k(z_0^i), k(z_1^i), \cdots, k(z_{h-1}^i), 0, \cdots, 0\big) \in \mathbb{R}^{H \cdot d_k}$ as the features of the first $h-1$ steps of the i-th example and define $v_h^i = C_v \cdot v(z_h^i)$ for some constant $C_v > 0$. Both k_h^i and v_h^i are random variables depending on the latent variable θ^*.

We define random variables $\mathcal{K}$ and $\mathcal{V}$ as uniform mixtures of $\{k_h^i\}_{h\in[H]}$ and $\{v(z_h^i)\}_{h\in[H]}$, respectively, with $h \sim \mathrm{Unif}([H])$. That is, with probability $1/H$, $(\mathcal{K}, \mathcal{V})$ has the same distribution as $(k_h^i, v(z_h^i))$ for any $h \in [H]$. Under the model in (5), $\mathcal{V}$ and $\mathcal{K}$ are linked via a linear model, i.e., $\mathbb{E}[\mathcal{V} \,|\, \mathcal{K}] = \theta^* \phi(\mathcal{K})$. Thus, θ^* can be viewed as the parameter of a linear regression problem, with $\mathcal{K}$ being the covariate and $\mathcal{V}$ being the response. To define a population-level estimator of θ^*, we define two population matrices

$$\begin{aligned} C_{\mathcal{K}\mathcal{K}} &= 1/H \sum_{h=1}^{H} \mathbb{E}\big[\phi(k_h^i)\phi(k_h^i)^\top\big] \in \mathbb{R}^{d_\phi \times d_\phi}, \\ C_{\mathcal{V}\mathcal{K}} &= 1/H \sum_{h=1}^{H} \mathbb{E}\big[v(z_h^i)\phi(k_h^i)^\top\big] \in \mathbb{R}^{d_v \times d_\phi}. \end{aligned} \tag{27}$$

Then we have $\theta^* = C_{\mathcal{V}\mathcal{K}} C_{\mathcal{K}\mathcal{K}}^{-1}$. When using this population-level estimator of θ^* to make predictions on the test instance, for any $h \geq 0$, we use k_h^{test} as the new covariate and predict the corresponding response, which is given by $C_{\mathcal{V}\mathcal{K}} C_{\mathcal{K}\mathcal{K}}^{-1} \phi(k_h^{\text{test}})$. In the rest of the proof, we relate $\bar{v}_h^{\text{test}}$ defined in (6) and the $\texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values})$ defined in (7) to this estimator, where $q_h^{\text{test}} = \phi(k_h^{\text{test}})$.

Step 2: Relate $\bar{v}_h^{\text{test}}$ to $C_{\mathcal{V}\mathcal{K}} C_{\mathcal{K}\mathcal{K}}^{-1} \phi(k_h^{\text{test}})$. In the following, for ease of notation, we let $\|\cdot\|_{\text{op}}$ and $\|\cdot\|_2$ denote the operator norm of matrices and the L^2 norm of vectors. Note that according to the model in (5), θ^* is shared in all the H steps. Thus, we can pool all the reasoning steps together to estimate θ^*. Notice that $\bar{v}_h^{\text{test}}$ defined in (6) is exactly the ridge regression estimator for the linear model $\mathcal{V} = \theta^* \phi(\mathcal{K}) + \texttt{noise}$, where the noise term is an independent $\mathcal{N}(0, \sigma^2)$ random variable. To simplify the notation, let $L = nH$. Recall that we define $\phi(K^n)$ and V^n in Section 4.4, where $\phi(K^n) \in \mathbb{R}^{d_\phi \times L}$ and $V^n \in \mathbb{R}^{d_v \times L}$. We define the sample-based counterparts of $C_{\mathcal{K}\mathcal{K}}$ and $C_{\mathcal{V}\mathcal{K}}$ in (27) as

$$\widehat{C}_{\mathcal{K}\mathcal{K}} = L^{-1} \cdot \phi(K^n)\phi(K^n)^\top \in \mathbb{R}^{d_\phi \times d_\phi}, \qquad \widehat{C}_{\mathcal{V}\mathcal{K}} = L^{-1} \cdot (V^n)\phi(K^n)^\top \in \mathbb{R}^{d_v \times d_\phi}. \tag{28}$$

In addition, we define the empirical correlation between value vectors as

$$\widehat{C}_{\mathcal{V}\mathcal{V}} = L^{-1} \cdot (V^n)(V^n)^\top \in \mathbb{R}^{d_v \times d_v}.$$

By definition, for any $h \in [H]$, can write $\bar{v}_h^{\text{test}}$ in (6) as

$$\bar{v}_h^{\text{test}} = \widehat{C}_{\mathcal{V}\mathcal{K}} (\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1} \cdot \sigma^2/\lambda \cdot I)^{-1} \phi(k_h^{\text{test}}).$$

By triangle inequality, we have

$$\begin{aligned}
&\big\|\bar{v}_h^{\text{test}} - C_{\mathcal{V}\mathcal{K}}C_{\mathcal{K}\mathcal{K}}^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\quad\le \underbrace{\big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}}) - C_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2}_{\text{(i)}} \\
&\qquad + \underbrace{\big\|C_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}}) - C_{\mathcal{V}\mathcal{K}}C_{\mathcal{K}\mathcal{K}}^{-1}q_h^{\text{test}}\big\|_2}_{\text{(ii)}}. \qquad (29)
\end{aligned}$$

Here (i) is the variance term of ridge regression which decays with the sample size L, and (ii) is the bias term that decays with the regularization parameter. Our analysis of these terms is similar to that in Zhang et al. (2023a). Note that in contrast to Zhang et al. (2023a), $\widehat{C}_{\mathcal{K}\mathcal{K}}$ and $\widehat{C}_{\mathcal{V}\mathcal{K}}$ defined in (28) involve $\{(k_j^i, v(z_j^i))\}_{j=1,i=1}^{H,n}$ that are dependent in j. We handle such dependency by decomposing the double sum according to different step $j \in [H]$ and apply concentration to each step. We first control the norm of (i) in (29) as

$$\begin{aligned}
\text{(i)} &\le \big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}}) - \widehat{C}_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\qquad + \big\|(\widehat{C}_{\mathcal{V}\mathcal{K}} - C_{\mathcal{V}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&= \big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}(\widehat{C}_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\qquad + \big\|(\widehat{C}_{\mathcal{V}\mathcal{K}} - C_{\mathcal{V}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2, \qquad (30)
\end{aligned}$$

where the first inequality follows from the triangle inequality and the second follows from the fact that

$$\begin{aligned}
&\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1} - \widehat{C}_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1} \\
&\quad= \widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1} \\
&\qquad - \widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1} \\
&\quad= \widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}(C_{\mathcal{K}\mathcal{K}} - \widehat{C}_{\mathcal{K}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1},
\end{aligned}$$

where the equality follows from rearranging the terms. We next separately bound the two terms in (30). For the first term in the right-hand side of (30), we have that

$$\begin{aligned}
&\big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}(\widehat{C}_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\quad\le \|\widehat{C}_{\mathcal{V}\mathcal{V}}\|_{\text{op}}^{1/2}\cdot\big\|\widehat{C}_{\mathcal{K}\mathcal{K}}^{1/2}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1/2}\big\|_{\text{op}}\cdot\big\|(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1/2}\big\|_{\text{op}} \\
&\qquad\cdot\big\|(\widehat{C}_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\quad\le (L^{-1}\cdot\sigma^2/\lambda)^{-1/2}\cdot\big\|(\widehat{C}_{\mathcal{K}\mathcal{K}} - C_{\mathcal{K}\mathcal{K}})(C_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1}\phi(k_h^{\text{test}})\big\|_2, \qquad (31)
\end{aligned}$$

where the first inequality follows from the cross-covariance operator decomposition (Theorem 1 in Baker (1973)) $\widehat{C}_{\mathcal{V}\mathcal{K}} = \widehat{C}_{\mathcal{V}\mathcal{V}}^{1/2}W\widehat{C}_{\mathcal{K}\mathcal{K}}^{1/2}$ for W such that $\|W\|_{\text{op}} \le 1$, and the second inequality follows from the facts that

$$\|\widehat{C}_{\mathcal{V}\mathcal{V}}\|_{\text{op}}^2 \le L^{-1}\sum_{i=1}^{n}\sum_{h=1}^{H}\|v(z_h^i)\|_2^2 \le 1, \qquad \big\|\widehat{C}_{\mathcal{K}\mathcal{K}}^{1/2}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1/2}\big\|_{\text{op}} \le 1, \text{ and}$$
$$\big\|(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\cdot\sigma^2/\lambda\cdot I)^{-1/2}\big\|_{\text{op}} \le (L^{-1}\cdot\sigma^2/\lambda)^{-1/2}.$$

We then upper bound the second term of the right-hand side of (30) using concentration inequality. To this end, based on the random variables $\mathcal{K}$ and $\mathcal{V}$, we consider a random variable

$$\mathcal{V}\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}}) \in \mathbb{R}^{d_v}.$$

Since $\|v(z)\|_2 = 1$ for all input $z \in \mathcal{L}$, therefore $\|\mathcal{V}\|_2 \leq 1$. Since the mapping ϕ has a bounded image set, we have

$$\begin{aligned}
&\big\|\mathcal{V}\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \\
&\quad \leq \|\mathcal{V}\|_2 \cdot \|\phi(\mathcal{K})^\top\|_2 \cdot \|(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\|_{\text{op}} \cdot \|\phi(k_h^{\text{test}})\|_2 \\
&\quad \leq C \cdot H \cdot (L^{-1}\sigma^2/\lambda)^{-1}, \qquad (32)
\end{aligned}$$

where C is an absolute constant such that $\|\phi(\cdot)\|_2 \leq C \cdot H$. We further bound the expected squared norm as

$$\begin{aligned}
&\mathbb{E}\Big[\big\|\mathcal{V}\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\big\|_2^2\Big] \\
&\quad \leq \mathbb{E}\Big[\|\mathcal{V}\|_2^2 \cdot \big\|\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\big\|_2^2 \cdot \big\|\phi(k_h^{\text{test}})\big\|_2^2\Big] \\
&\quad \leq C \cdot H \cdot \mathbb{E}\Big[\big\|\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\big\|_2^2\Big] \\
&\quad \leq C \cdot H \cdot (L^{-1}\sigma^2/\lambda)^{-1} \cdot \mathbb{E}\Big[\big\langle (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(\mathcal{K}), \phi(\mathcal{K})\big\rangle\Big]. \qquad (33)
\end{aligned}$$

For the expectation in the right-hand side of (33), we further have that

$$\begin{aligned}
&\mathbb{E}\Big[\big\langle (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(\mathcal{K}), \phi(\mathcal{K})\big\rangle\Big] \\
&\quad = \mathbb{E}\Big[\text{Trace}\big((C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(\mathcal{K})\phi(\mathcal{K})^\top\big)\Big] \\
&\quad = \text{Trace}\big((C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}C_{\mathcal{K}\mathcal{K}}\big) \\
&\quad = d_\phi - L^{-1}\sigma^2/\lambda \cdot \text{Trace}\big((C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\big) \leq d_\phi, \qquad (34)
\end{aligned}$$

where we assume $\phi(\cdot)$ to be finite dimensional. The last line follows from the direct calculation and the fact that $(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}$ is positive definite. Thus, we have

$$\mathbb{E}\Big[\big\|\mathcal{V}\phi(\mathcal{K})^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\big\|_2^2\Big] \leq C \cdot H \cdot (L^{-1}\sigma^2/\lambda)^{-1} d_\phi \qquad (35)$$

for some constant $C > 0$. Recall that $\mathcal{V}$ and $\mathcal{K}$ are defined as uniform mixtures of $v(z_h^i)$ and k_h^i with $h \sim \texttt{Unif}[H]$. Therefore, (32) and (35) hold for each fixed $h \in [H]$. Specifically, note that the random variables corresponding to the same step $h \in [H]$ across different examples $i \in [n]$ are i.i.d. That is, for any fixed h, $\big\{\big(k_h^i, v(z_h^i)\big)\big\}_{i=1}^n$ is a sequence of i.i.d samples. Therefore, we can apply Lemma 49 to each $j \in [H]$. More specifically, we can apply Lemma 49 to each sample mean with fixed step index h by setting $B = 2C \cdot H \cdot (L^{-1}\sigma^2/\lambda)^{-1}$

and variance as $C \cdot H \cdot (L^{-1}\sigma^2/\lambda)^{-1} d_\phi$. Therefore, with probability at least $1-\delta$ we have

$$
\begin{aligned}
&\big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}}) - C_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\big\|_2 \qquad (36)\\
&= \bigg\| \sum_{j=1}^{H} \frac{1}{H} \bigg(\frac{1}{n}\sum_{i=1}^{n} v(z_j^i)\phi(k_j^i)^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}}) \\
&\qquad\qquad - \mathbb{E}\big[v(z_j^i)\phi(k_j^i)^\top\big](C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\bigg)\bigg\|_2 \\
&\le \sum_{j=1}^{H} \frac{1}{H} \bigg\| \bigg(\frac{1}{n}\sum_{i=1}^{n} v(z_j^i)\phi(k_j^i)^\top (C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}}) \\
&\qquad\qquad - \mathbb{E}\big[v(z_j^i)\phi(k_j^i)^\top\big](C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\phi(k_h^{\text{test}})\bigg)\bigg\|_2 \\
&\le C \cdot \bigg(H(L^{-1}\sigma^2/\lambda)^{-1} n^{-1} + \sqrt{\frac{H(L^{-1}\sigma^2/\lambda)^{-1} d_\phi}{n}} \bigg) \cdot \log\frac{2H}{\delta} \\
&= C \cdot H \cdot \bigg(H\lambda/\sigma^2 + \sqrt{\lambda \cdot d_\phi/\sigma^2} \bigg) \cdot \log\frac{2H}{\delta},
\end{aligned}
$$

where $C > 0$ is an absolute constant. Here the second line results from triangle inequality and the third line follows from Lemma 49. Similarly, we perform the same analysis for $C_{\mathcal{K}\mathcal{K}}$ and we obtain with probability at least $1-\delta$ that

$$
\begin{aligned}
&\big\|\widehat{C}_{\mathcal{K}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1} - C_{\mathcal{K}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\big\|_2 \qquad (37)\\
&\quad \le C' \cdot H \cdot \bigg(H\lambda/\sigma^2 + \sqrt{\lambda \cdot d_\phi/\sigma^2} \bigg) \cdot \log\frac{2H}{\delta}.
\end{aligned}
$$

Here $C' > 0$ is an absolute constant. Therefore, we bound the first term in the upper bound in (29) using (30), (31), (36), and (37) as

$$
\begin{aligned}
&\big\|\widehat{C}_{\mathcal{V}\mathcal{K}}(\widehat{C}_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1} - C_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1}\sigma^2/\lambda I)^{-1}\big\|_2 \qquad (38)\\
&\quad \le C'' \cdot \sqrt{\frac{L}{\sigma^2/\lambda}} \cdot H \cdot \bigg(H\lambda/\sigma^2 + \sqrt{\lambda \cdot d_\phi/\sigma^2} \bigg) \log\frac{2H}{\delta}.
\end{aligned}
$$

which holds with probability at least $1-\delta$.

To bound the second term in (29), we can apply the bound in Equation (E.13) from the proof of Zhang et al. (2023a) directly. Namely, we have that for any $q \in \mathbb{R}^{H \cdot d_k}$,

$$
\big\|C_{\mathcal{V}\mathcal{K}}(C_{\mathcal{K}\mathcal{K}} + L^{-1} \cdot \sigma^2/\lambda I)^{-1}\phi(q) - C_{\mathcal{V}\mathcal{K}} C_{\mathcal{K}\mathcal{K}}^{-1}\phi(q)\big\|_2 \le C \cdot \sigma^2/\lambda L^{-1}, \qquad (39)
$$

where $C > 0$ is an absolute constant. Combining (38) and (39), we obtain

$$
\begin{aligned}
&\big\|\bar{v}_h^{\text{test}} - \texttt{CME}(k_h^{\text{test}}, \mathbb{P}_{\mathcal{K},\mathcal{V}})\big\|_2 \\
&\quad = \mathcal{O}\bigg(\sqrt{\frac{L \cdot \lambda}{\sigma^2}} \cdot H \cdot \bigg(H\lambda/\sigma^2 + \sqrt{d_\phi \cdot \lambda/\sigma^2} \bigg) \log\frac{2H}{\delta} + \sigma^2/\lambda L^{-1} \bigg). \qquad (40)
\end{aligned}
$$

Thus the error bound in (40) goes to 0 by selecting $\sigma^2/\lambda = (nH)^{2/3}$.

Step 3: Relate $\texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values})$ to $C_{\mathcal{VK}} C_{\mathcal{KK}}^{-1} \phi(k_h^{\text{test}})$. In this step, we want to prove that $\texttt{attn}(q, \texttt{keys}, \texttt{values})$ defined in (7) converges to $C_{\mathcal{VK}} C_{\mathcal{KK}}^{-1} \phi(k_h^{\text{test}})$ as $n \to \infty$. Recall that we define $v_h^i = C_v \cdot v(z_h^i)$ as the value of softmax attention. To achieve this goal, we aim to show that $C \cdot \texttt{attn}(q, \texttt{keys}, \texttt{values}) = \int_{\mathbf{S}^{d_v - 1}} v \widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q) \mathrm{d}v$, where $\widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q)$ is an estimator of the conditional distribution of $\mathcal{V}$ given $\mathcal{K}$, and C is an absolute constant. Here we let $\mathbf{S}^d$ denote a d-dimensional unit sphere. We define the empirical kernel conditional density as follows,

$$\widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q) = \iota \frac{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big) \exp\big(\langle v_h^i, v \rangle\big)}{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big)},$$

where $\iota = 1/\int_{\mathbf{S}^{d_k - 1}} \exp\big(\langle v_h^i, v \rangle\big) \mathrm{d}v = 1/\int_{\mathbf{S}^{d_k - 1}} \exp\big(\langle v(z_h^i), v \rangle\big) \mathrm{d}v$ is a normalizing constant to make sure that $\widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}$ is a probability measure. Furthermore, note that ι does not depend on v_h^i due to the symmetry of the unit sphere.

We compute the integration over v as

$$\begin{aligned}
\int_{\mathbf{S}^{d_v - 1}} v \widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q) \mathrm{d}v &= \iota \cdot \frac{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big) \int_{\mathbf{S}^{d_v - 1}} v \exp\big(\langle v(z_h^i), v \rangle\big) \mathrm{d}v}{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big)} \\
&= \iota \cdot \frac{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big) \cdot C_1 \cdot v_h^i}{\sum_{i=1}^n \sum_{h=1}^H \exp\big(\langle k_h^i, q \rangle\big)} \\
&= \frac{C_1}{\iota} \cdot \texttt{attn}(q, \texttt{keys}, \texttt{values}).
\end{aligned}$$

Since $v, v(z_h^i) \in \mathbf{S}^{d_v - 1}$, we can apply Lemma 52 with $\gamma = 1$ to the integration $\int_{\mathbf{S}^{d_v - 1}} v \exp(\langle C_v \cdot v(z_h^i), v \rangle) \mathrm{d}v$, and obtain that the second line holds for some constant $C_1 > 0$. The last line follows directly from the definition of softmax attention 17.

Due to the condition where $\widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q) \to \mathbb{P}(v \,|\, q)$ uniformly for any $q \in \mathbf{S}^{d_q}$ as $n \to \infty$, the integral $\int_{\mathbf{S}^{d_v - 1}} v \widehat{\mathbb{P}}_{\mathcal{V} \,|\, \mathcal{K}}(v \,|\, q) \mathrm{d}v \to \mathbb{E}[\mathcal{V} \,|\, \mathcal{K} = q]$ as $n \to \infty$. Combining with the previous argument, we have shown that,

$$\frac{C_1}{\iota} \cdot \texttt{attn}(q, \texttt{keys}, \texttt{values}) \to \mathbb{E}[\mathcal{V} \,|\, \mathcal{K} = q] \qquad \text{as} \quad n \to \infty, \tag{41}$$

Combining the results from the previous two steps (40) and (41), we have shown that

$$\lim_{n \to \infty} \max_{h \in [H]} \| \bar{v}_h^{\text{test}} - C \cdot \texttt{attn}(q_h^{\text{test}}, \texttt{keys}, \texttt{values}) \|_2 = 0$$

for some absolute constant C. Therefore, we conclude the proof. Note that this proof assumes ϕ is finite-dimensional, but it also holds for infinite-dimensional ϕ. More specifically, by replacing the trace in (34) with the effective dimension of ϕ, we can still balance the rate of σ^2/λ to ensure (40) goes to zero.

■

Appendix E. Proofs and Additional Results of Section 5.1

This section provides proofs and additional results about the statistical properties of the vanilla CoT estimator. In Appendix E.1 we prove Lemma 4, which establishes the error decomposition of vanilla CoT error $\texttt{err}_{\mathrm{CoT}}$. In Appendix E.2, we extend Theorems 7 and 10 to the scenario where Θ is continuous, and provide their corresponding proofs. We conclude this section with an additional result that characterizes $\texttt{err}_{\mathrm{CoT}}$ under the general model in (20).

E.1 Proof of Lemma 4

Proof Recall that the error of the CoT estimator is defined as

$$\texttt{err}_{\mathrm{CoT}} = \texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big).$$

We decompose the KL divergence into three terms by direct computation:

$$\begin{aligned}\texttt{err}_{\mathrm{CoT}} = {} & \texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \\ & + \texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \\ & + \int \big(\mathbb{P}(y^{\mathrm{test}} = y \,|\, z_0^{\mathrm{test}}, \theta^*) - \mathbb{P}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \\ & \qquad \cdot \log \frac{\mathbb{P}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}{\mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)))} \mathrm{d}y. \end{aligned} \tag{42}$$

Note that we can upper bound the marginal log density ratio $\log(\mathbb{P}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))/\mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)))$ by aggregating the density ratio at each step. More specifically, we have

$$\begin{aligned} & \log\left(\frac{\mathbb{P}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}{\mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}\right) \\ & = \log\left(\frac{\sum_{z_{1:(H-1)} \in \mathcal{L}^*} \mathbb{P}(z^{\mathrm{test}}_{1:(H-1)} = z_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}{\sum_{z'_{1:(H-1)} \in \mathcal{L}^*} \mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_{1:(H-1)} = z'_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}\right) \\ & \le \log\left(\max_{z_{1:(H-1)} \in \mathcal{L}^*} \frac{\mathbb{P}(z^{\mathrm{test}}_{1:(H-1)} = z_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}{\mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_{1:(H-1)} = z_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}\right), \end{aligned} \tag{43}$$

where the first line follows from marginalizing over the intermediate steps $z^{\mathrm{test}}_{1:(H-1)}$, and the second line follows from generalized mediant inequality.

Next, we use the chain rule to decompose the joint distribution as

$$\begin{aligned} & \frac{\mathbb{P}(z^{\mathrm{test}}_{1:(H-1)} = z_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))}{\mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_{1:(H-1)} = z'_{1:(H-1)}, y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))} \\ & = \prod_{h=1}^{H} \frac{\mathbb{P}(z^{\mathrm{test}}_h = z_h \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n))}{\mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_h = z'_h \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n))} \le e^{H \cdot b^*}, \end{aligned} \tag{44}$$

where the first line follows directly from the chain rule of conditional probability, and the second line follows from Assumption 3. Combining (43) and (44), we have

$$\big|\log \mathbb{P}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)) - \log \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big| \le H b^*$$

for all $y \in \mathcal{L}$. Thus, we can upper bound the integral in the right-hand side of (42) by

$$2\mathtt{TV}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big) \cdot Hb^*.$$

By Pinsker's inequality, we further have that

$$\begin{aligned} &2\mathtt{TV}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big) \cdot Hb^* \\ &\quad \le 2\sqrt{2}Hb^* \cdot \Big(\mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big)\Big)^{1/2}. \end{aligned} \tag{45}$$

Combining (42) and (45), we conclude that

$$\begin{aligned} \mathtt{err}_{\mathrm{CoT}} \le\ & \underbrace{\mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big)}_{\mathtt{err}_{\mathrm{pre}}} \\ &+ \underbrace{\mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}} \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big)}_{\mathtt{err}_{\mathrm{prompt}}\text{-(i)}} \\ &+ \underbrace{2\sqrt{2}Hb^* \cdot \mathtt{KL}^{1/2}\big(\mathbb{P}(y^{\mathrm{test}} \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big)}_{\mathtt{err}_{\mathrm{prompt}}\text{-(ii)}}. \end{aligned}$$

Therefore, we conclude the proof. Here the upper bound consists of three parts. The first term characterizes the pretraining error by comparing $\mathbb{P}$ and $\mathbb{P}_{\widehat{\rho}}$. The second and third terms involve the KL divergence between the true distribution $\mathbb{P}(\cdot \,|\, \theta^*)$ and the distribution induced by the CoT prompt and $\mathbb{P}$. ■

E.2 Proofs of Theorems 7 and 10 and Extension

In the sequel, we generalize the results in Theorems 7 and 10 to the scenario where Θ is continuous and provide the corresponding proofs. The proofs of the supporting lemmas used in this subsection are deferred to Appendix I.2.

We begin this section by specifying the distance measurement on the general Θ, which can be continuous. For any two hidden concept $\theta_1, \theta_2 \in \Theta$, we define a *loglikelihood metric* between them as

$$\|\theta_0 - \theta_1\|_\Theta = \sup_{(Z_0, Z_1, \cdots, Z_{H-1}, Y)} \left| \log \frac{\mathbb{P}(Z_0, Z_1, \cdots, Z_{H-1}, Y \,|\, \theta_0)}{\mathbb{P}(Z_0, Z_1, \cdots, Z_{H-1}, Y \,|\, \theta_1)} \right|.$$

We note that the loglikelihood metric is indeed a semi-metric. For any $\theta \in \Theta$, any other concepts in the neighborhood of it share a similar conditional distribution on one example. Based on this semi-metric, we then define an α-cover of $\mathcal{S}$ and the corresponding $\mathcal{N}(\alpha, \mathcal{S})$ for any number $\alpha > 0$ and any set $\mathcal{S} \subseteq \Theta$.

That is, $\mathcal{C}(\alpha, \mathcal{S}) = \{\theta_i\}_{i=1}^N \subseteq \Theta$ is an α-cover of $\mathcal{S}$ of size N if $\mathcal{S} \subseteq \bigcup_{i=1}^N B(\theta_i, \alpha)$, where $B(\theta_i, \alpha)$ is a neighborhood of θ_i with radius α, i.e.,

$$B(\theta, \alpha) = \big\{\widetilde{\theta} \in \Theta^{\complement} : \|\widetilde{\theta} - \theta\|_\Theta \le \alpha\big\}. \tag{46}$$

The minimal N such that there exists a α-cover of $\mathcal{S}$ of size N is called the covering number of $\mathcal{S}$, which is denoted as $\mathcal{N}(\alpha, \mathcal{S})$. In the following, we consider the complement of the equivalence class of the target concept $\Theta^{\complement} = \Theta \backslash \Theta_{\mathrm{eq}}(\theta^*)$. Without misunderstanding, we adopt $\mathcal{N}(\alpha)$ to denote $\mathcal{N}(\alpha, \Theta^{\complement})$ in the following. Then we restate Theorem 7 with Θ allowed to be a continuous set.

Theorem 27 *Let* $\mathtt{err}_{\mathrm{pre}}$ *denote the pretraining error defined in* (9) *and denote* $\Theta^{\complement} = \Theta \backslash \Theta_{\mathrm{eq}}(\theta^*)$. *Under Assumptions 3 and 6, the statistical error* $\mathtt{err}_{\mathrm{CoT}}$ *defined in* (4) *is bounded under the following two cases:*

- *When* Θ *is a discrete and finite set, with probability* $1-\delta$, *we have*

$$\mathtt{err}_{\mathrm{CoT}} \leq \mathcal{O}\big(Hb^* \cdot \pi(\theta^*)^{-1/2} \cdot \delta^{-1} \cdot \big|\Theta^{\complement}\big| \cdot e^{-\lambda n}\big) + \mathtt{err}_{\mathrm{pre}}.$$

- *When* Θ *is continuous, let* $\mathcal{N}(\alpha)$ *denote the covering number of* $\Theta^{\complement}$ *with precision* α *with respect to the log-likelihood metric* $\|\cdot\|_{\Theta}$. *Under Assumption 13, with probability* $1-\delta$, *we have*

$$\mathtt{err}_{\mathrm{CoT}} \leq \mathcal{O}\Big(Hb^* \cdot \pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)^{-1} \cdot \sqrt{c_0^{-nH} \cdot \delta^{-2} \cdot \mathcal{N}(\alpha)^2 \cdot e^{-2n\lambda} + n\alpha}\Big) + \mathtt{err}_{\mathrm{pre}}.$$

Here the probability is with respect to the randomness of the CoT prompt.

Moreover, we remark that when the LLM is perfectly pretrained, with probability $1-\delta$ *with respect to the randomness of CoT prompt, we have*

$$\mathtt{err}_{\mathrm{CoT}} \leq \begin{cases} \mathcal{O}\big(\pi(\theta^*)^{-1}\delta^{-2}|\Theta^{\complement}|^2 e^{-2\lambda n}\big) & \text{when } \Theta \text{ is finite,} \\ \mathcal{O}\big(\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)^{-1} c_0^{-nH}\delta^{-2}\mathcal{N}(\alpha)^2 e^{-2n\lambda} + n\alpha\big) & \text{when } \Theta \text{ is continuous.} \end{cases}$$

When the set Θ is continuous, the statistical rate becomes slower due to the complicated structure of the task parameter space Θ. In the theorem statement, we do not specify the value of the covering number α. In fact, α should be chosen depending on n such that $\alpha \cdot n$ converges to 0 as n increases. One can easily obtain a more concrete statistical rate by specifying a concrete α depending on specific assumptions of the covering number. For instance, suppose $\mathcal{N}(\alpha) = \mathcal{O}(\alpha^{-V})$ for some constant V, by selecting

$$\alpha = \mathcal{O}\big(n^{-1/(2V+1)} \exp\{-n(2\lambda + H \ln c_0)/(2V+1)\}\big),$$

prompting error $\mathtt{err}_{\mathrm{prompt}}$ is $\mathcal{O}\big(n^{V/(2V+1)} \exp\{-n(\lambda + H \ln c_0/2)/(2V+1)\}\big)$ with probability at least $1-\delta$.

Proof [Proof of Theorem 7]

We divide the proof into three parts. In Step 1, we derive an upper bound of KL divergence to prepare for later analysis. In Step 2, we analyze the case where $\Theta^{\complement}$ is discrete and finite. In Step 3, we extend to continuous $\Theta^{\complement}$.

Step 1: Derive an upper bound of KL divergence. We first invoke the following proposition to establish an upper bound of $\mathtt{KL}(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)))$.

Proposition 28 *For some fixed task $\theta^* \in \Theta$, we provide upper bounds for the KL-divergence of the ground truth distribution $\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*)$ from the conditional pretrained distributions $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ as follows*

$$\begin{aligned}&\texttt{KL}\Big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\Big)\\ &\quad\le \log\left(1 + \frac{\int_{\Theta^{\complement}} \mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)d\theta}{\int_{\Theta_{\text{eq}}(\theta^*)} \mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)d\theta}\right).\end{aligned}$$

Proof See Appendix I.2 for a detailed proof. ■

This proposition applies evidence lower bound (Kingma and Welling, 2013) to upper bound the KL divergence using likelihood ratios. The proof involves using a variational distribution that is only supported on Θ_{eq} and proportional to the posterior distribution. This proposition reduces the problem of bounding the KL divergence to comparing the likelihood functions on Θ_{eq} and $\Theta^{\complement}$. We consider the cases where Θ is finite and continuous separately.

Step 2: Statistical rate for the case with a discrete and finite Θ. In this step, we assume the parameter space Θ is discrete and finite. Then by Proposition 28, we have

$$\begin{aligned}&\texttt{KL}\left(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\right)\\ &\quad\le \log\left(1 + \frac{\sum_{\theta\in\Theta^{\complement}} \mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)}{\sum_{\theta\in\Theta_{\text{eq}}(\theta^*)} \mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)}\right)\\ &\quad\le \log\left(1 + \frac{\sum_{\theta\in\Theta^{\complement}} \mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)}{\mathbb{P}\big(\texttt{prompt}_{\text{CoT}}(n) \,\big|\, \theta^*\big)\pi(\theta^*)}\right).\end{aligned} \tag{47}$$

Here the first inequality follows from Proposition 28 by changing the integration signs into summations. In the second inequality, we drop some terms in the denominator to get an upper bound.

Note that we have converted the upper bound into the logarithm of a weighted sum of likelihood ratios, with the weights being the prior distribution. We invoke the following lemma to establish an upper bound for each likelihood ratio.

Lemma 29 *Let $S_n = \{S^i\}_{i\in[n]}$, where $S^i = z^i_{0:H} \overset{i.i.d.}{\sim} \mathbb{P}(\cdot \,|\, \theta^*)$ denotes a set of n reasoning paths of length H sampled independently from the model in (2) with task θ^*. Let $J_i \subseteq [H-1]$ for each $i \in [n]$, and we use $S^i_{J_i}$ to denote a truncated version of the i-th trajectory S^i corresponding to the indices specified by J_i. Namely, $S^i_{J_i} = \{z^i_0\} \cup \{z^i_j : j \in J_i\} \cup \{z^i_H\}$. Then for any $n \ge 1$, $\theta \in \Theta$, and $\delta > 0$, we have*

$$\frac{\mathbb{P}(\{S_{J_i}\}_{i=1}^n \,|\, \theta)}{\mathbb{P}(\{S_{J_i}\}_{i=1}^n \,|\, \theta^*)} \le \exp\left(-2\sum_{i=1}^n \mathrm{H}^2\big(\mathbb{P}(S_{J_i} \,|\, \theta^*), \mathbb{P}(S_{J_i} \,|\, \theta)\big) + 2\log(\delta^{-1})\right),$$

with probability at least $1-\delta$. Here the probability is with respect to the randomness of S_n, and $\mathrm{H}^2(\cdot,\cdot)$ denotes the squared Hellinger distance.

Proof See Appendix I.2 for a detailed proof. ■

This lemma provides an upper bound of the likelihood ratio of generating trajectories $\{S_{J_i}\}_{i=1}^n$ from two distributions $\mathbb{P}(\cdot\,|\,\theta)$ and $\mathbb{P}(\cdot\,|\,\theta^*)$, where $\mathbb{P}(\cdot\,|\,\theta^*)$ is the ground truth distribution. The upper bound is related to the Hellinger distance between them.

Recall that $\texttt{prompt}_{\mathrm{CoT}}(n)$ contains n complete trajectories and a testing query z_0^{test}. Applying Lemma 29 to any $\theta \in \Theta^{\complement}$ and θ^* and taking a union bound, we conclude that,

$$\begin{aligned}
&\sup_{\theta\in\Theta^{\complement}} \left\{\frac{\mathbb{P}\big(\texttt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}{\mathbb{P}\big(\texttt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)}\right\} \\
&\quad\le \sup_{\theta\in\Theta^{\complement}} \exp\Big(-2n\mathrm{H}^2\big(\mathbb{P}(Z_{0:H}\,|\,\theta^*),\mathbb{P}(Z_{0:H}\,|\,\theta)\big) - 2\mathrm{H}^2\big(\mathbb{P}(z_0\,|\,\theta^*),\mathbb{P}(z_0\,|\,\theta)\big) \\
&\qquad\quad + 2\log\big(\delta^{-1}|\Theta^{\complement}|\big)\Big) \\
&\quad\le \exp\big(-2n\lambda + 2\log\big(\delta^{-1}|\Theta^{\complement}|\big)\big)
\end{aligned} \tag{48}$$

holds with probability at least $1-\delta$. The first inequality follows from Lemma 29 and the second inequality follows from Assumption 6 and the fact that $\mathrm{H}^2\big(\mathbb{P}(z_0\,|\,\theta^*),\mathbb{P}(z_0\,|\,\theta)\big) \ge 0$. Thus, plugging this inequality into the upper bound in (47), with probability at least $1-\delta$, we have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\texttt{prompt}_{\mathrm{CoT}}(n)\big)\Big) \\
&\quad\le \log\left[1+\frac{1}{\pi(\theta^*)}\cdot\exp\big(-2n\lambda+2\log\big(\delta^{-1}|\Theta^{\complement}|\big)\big)\cdot\sum_{\theta\in\Theta^{\complement}}\pi(\theta)\right] \\
&\quad= \log\left[1+\frac{1-\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}{\pi(\theta^*)}\cdot\exp\Big(-2n\lambda+2\log\big(\delta^{-1}|\Theta^{\complement}|\big)\Big)\right],
\end{aligned}$$

Therefore, when Θ is discrete and finite, with probability at least $1-\delta$, we have

$$\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\texttt{prompt}_{\mathrm{CoT}}(n)\big)\Big) = \mathcal{O}\big(\pi(\theta^*)^{-1}\delta^{-2}|\Theta^{\complement}|^2 e^{-2\lambda n}\big), \tag{49}$$

where $\mathcal{O}(\cdot)$ only omits an absolute constant. Here we use the fact that $1-\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)<1$. Recall that the prompting error is defined as

$$\begin{aligned}
\texttt{err}_{\mathrm{prompt}} = {}& \mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,\texttt{prompt}_{\mathrm{CoT}}(n))\big) \\
&+ 2\sqrt{2}Hb^*\sqrt{\mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,\texttt{prompt}_{\mathrm{CoT}}(n))\big)}.
\end{aligned}$$

Therefore, we conclude that for any task θ^* with separation from $\Theta^{\complement}$, the prompting error goes to zero exponentially fast at a rate of order $\mathcal{O}\big(Hb^*\pi(\theta^*)^{-1/2}\delta^{-1}|\Theta^{\complement}|e^{-\lambda n}\big)$. This proves Theorem 7.

Step 3: Statistical rate for the case with a continuous Θ. Our analysis in **Step 2** requires Θ to be discrete and finite. To handle the continuous case, we use the cover of $\Theta^{\complement}$ to discretize it, at the cost of introducing an additional error involving the covering

granularity. Specifically, let $\mathcal{C}(\alpha)$ be an α-cover of Θ according to the semi-metric $\|\cdot\|_\Theta$. For any θ, let the neighborhood $B(\theta,\alpha)$ be defined in (46). Let $\mathcal{N}(\alpha)$ denote the covering number $\mathcal{N}(\alpha,\Theta^{\mathtt{C}})$. Using the minimal α-cover of $\Theta^{\mathtt{C}}$, we can construct a partition of $\Theta^{\mathtt{C}}$ into at most $\mathcal{N}(\alpha)$ disjoint sets, with each set contained in a neighborhood of radius α. To see this, let $\mathcal{C}(\alpha)=\{\theta_i\}_{i=1}^{\mathcal{N}(\alpha)}$ be the α-cover of $\Theta^{\mathtt{C}}$. We can construct set $C(\theta_i)\subset B(\theta_i,\alpha)$ such that $\{C(\theta_i)\}_{i=1}^{\mathcal{N}(\alpha)}$ form a partition of $\Theta^{\mathtt{C}}$ by shrinking each $B(\theta_i,\alpha)$ to remove overlapping parts. Then, we have $C(\theta_i)\cap C(\theta_j)=\emptyset$ for all $i\neq j$, and $\bigcup_{i=1}^{\mathcal{N}(\alpha)} C(\theta_i)=\Theta^{\mathtt{C}}$. We characterize the discretization error due to the α-cover via

$$
\begin{aligned}
&\log\left(\frac{\int_{\theta\in\Theta^{\mathtt{C}}}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi(\theta)\mathrm{d}\theta}{\sum_{\theta\in\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C(\theta)\big)}\right)\\
&\quad=\log\left(\frac{\sum_{\theta\in\mathcal{C}(\alpha)}\pi\big(C(\theta)\big)\int_{\theta'\in C(\theta)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)\cdot\pi(\theta')/\pi\big(C(\theta)\big)\mathrm{d}\theta'}{\sum_{\theta\in\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C(\theta)\big)}\right)\\
&\quad\le\log\left(\max_{\theta\in\mathcal{C}(\alpha)}\frac{\int_{\theta'\in C(\theta)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)\pi(\theta')/\pi\big(C(\theta)\big)\mathrm{d}\theta'}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}\right)\\
&\quad\le\log\left(\sup_{\theta\in\mathcal{C}(\alpha),\theta'\in C(\theta)}\frac{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}\right)\le n\alpha.
\end{aligned}
\tag{50}
$$

The first equality follows from decomposing the integral taken over $\Theta^{\mathtt{C}}$ into a double integral taken over the covering $\mathcal{C}(\alpha)$ and then within the induced partition $C(\theta)$. The second and third inequality follows from generalized mediant inequality. The last inequality follows from the definition of $B(\theta,\alpha)$ in (46) and the fact that $C(\theta)\subseteq B(\theta,\alpha)$ for all $\theta\in\mathcal{C}(\alpha)$.

Now we have controlled the error introduced by approximating $\Theta^{\mathtt{C}}$ using $\mathcal{C}(\alpha)$. Next, we apply Assumption 13 to lower bound the likelihood integrated over $\Theta_{\mathrm{eq}}(\theta^*)$:

$$
\int_{\Theta_{\mathrm{eq}}(\theta^*)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)\pi(\theta')\mathrm{d}\theta'\ge c_0^{nH}\cdot\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\cdot\pi(\Theta_{\mathrm{eq}}(\theta^*)).
\tag{51}
$$

The inequality follows from the fact that $\mathtt{prompt}_{\mathrm{CoT}}(n)$ is of length nH and Assumption 13, which provides a lower bound for the conditional probability of the next reasoning step.

Combining (51) and (50) and using the same technique as in **Step 2**, we obtain

$$
\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big)\\
&\quad\le\log\left(1+\frac{\int_{\theta\in\Theta^{\mathtt{C}}}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi(\theta)\big)\mathrm{d}\theta}{c_0^{nH}\cdot\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\cdot\pi(\Theta_{\mathrm{eq}}(\theta^*))}\right)\\
&\quad\le\log\left(1+\frac{\sum_{\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C_n(\theta)\big)}{c_0^{nH}\cdot\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\cdot\pi(\Theta_{\mathrm{eq}}(\theta^*))}\right)+n\alpha\\
&\quad\le\log\left[1+\frac{1-\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}{c_0^{nH}\cdot\pi(\Theta_{\mathrm{eq}}(\theta^*))}\exp\Big(-2n\lambda+2\log\big(\delta^{-1}\mathcal{N}(\alpha)\big)\Big)\right]+n\alpha,
\end{aligned}
$$

with probability at least $1-\delta$. Here the first inequality is due to Proposition 28 and (51). The second inequality follows from (50), accounting for the discretization error induced by

the α-cover. The last inequality follows from the same strategy as in **Step 2**, where we apply Lemma 29. Finally, we conclude that , with probability at least $1-\delta$, we have

$$\begin{aligned}&\mathtt{KL}\Big(\mathbb{P}(y^{\text{test}}=\cdot\,|\,z_0^{\text{test}},\theta^*),\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\mathtt{prompt}_{\text{CoT}}(n)\big)\Big)\\&\quad=\mathcal{O}\Big(\pi\big(\Theta_{\text{eq}}(\theta^*)\big)^{-1}\cdot c_0^{-nH}\cdot\delta^{-2}\cdot\mathcal{N}(\alpha)^2\cdot e^{-2n\lambda}+n\alpha\Big),\end{aligned}$$

where $\mathcal{O}(\cdot)$ only hides absolute constants and we use the fact that the numerator $1-\pi\big(\Theta_{\text{eq}}(\theta^*)\big)<1$. Therefore, we conclude the proof. ■

The rest of this section generalizes Theorem 10 to the case where Θ is continuous and provides the proof.

Theorem 30 *Let* $\mathtt{err}_{\text{pre}}$ *denote the pretraining error defined in* (9)*, and let* $\Theta^{\mathfrak{C}}=\Theta\backslash\Theta_{\text{eq}}(\theta^*)$ *with* $\widetilde{\Theta^{\mathfrak{C}}}$ *as its representative set. Under Assumptions 3, 6, and 9, the statistical error* $\mathtt{err}_{\text{CoT}}$ *defined in* (4) *is bounded under the following two cases:*

- *When* $\widetilde{\Theta^{\mathfrak{C}}}$ *is a discrete and finite set, with probability* $1-\delta$*, we have*

$$\mathtt{err}_{\text{CoT}}\le\mathcal{O}\big(Hb^*\cdot\pi(\theta^*)^{-1/2}\cdot\delta^{-1}\cdot\big|\widetilde{\Theta^{\mathfrak{C}}}\big|\cdot e^{-(\lambda-\alpha)n+\alpha_0}\big)+\mathtt{err}_{\text{pre}}.$$

- *When* Θ *is continuous, let* $\mathcal{N}(\alpha)$ *denote the covering number of* $\widetilde{\Theta^{\mathfrak{C}}}$ *with precision* α*. With probability* $1-\delta$*, we have:*

$$\mathtt{err}_{\text{CoT}}\le\mathcal{O}\Big(Hb^*\cdot\pi\big(\Theta_{\text{eq}}(\theta^*)\big)^{-1}\cdot\sqrt{\delta^{-2}\cdot\mathcal{N}(\alpha)^2\cdot e^{-2n(\lambda-\alpha)+2\alpha_0}+n\alpha}\Big)+\mathtt{err}_{\text{pre}}.$$

Here the probability is with respect to the randomness of CoT prompts $\mathtt{prompt}_{\text{CoT}}(n)$*. Parameters* α *and* α_0 *are introduced in Assumption 9.*

Moreover, we remark that when the LLM is perfectly pretrained, with probability $1-\delta$ *with respect to the randomness of CoT prompt, we have*

$$\mathtt{err}_{\text{CoT}}\le\begin{cases}\mathcal{O}\big(\pi\big(\Theta_{\text{eq}}(\theta^*)\big)^{-1}\cdot\delta^{-2}\cdot|\widetilde{\Theta^{\mathfrak{C}}}|^2\cdot e^{-2(\lambda-\alpha)n+2\alpha_0}\big) & \text{when } \Theta \text{ is finite,}\\ \mathcal{O}\big(\delta^{-2}\cdot\pi\big(\Theta_{\text{eq}}(\theta^*)\big)^{-1}\cdot\mathcal{N}(\alpha)^2\cdot e^{-2n(\lambda-\alpha)+2\alpha_0}+n\alpha\big) & \text{when } \Theta \text{ is continuous.}\end{cases}$$

This error bound follow from the fact that, when the LLM is perfectly pretrained, $\mathrm{err}_{\text{CoT}}=\mathtt{KL}\big(\mathbb{P}(y^{\text{test}}=\cdot\,|\,z_0^{\text{test}},\theta^*),\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\mathtt{prompt}_{\text{CoT}}(n)\big)\big)$.

Theorem 30 builds on Theorem 27 by incorporating Assumption 9, which postulates that distributions within each equivalence class are close. This leads to improved dependency on the parameter space size, shifting from $\Theta^{\mathfrak{C}}$ to $\widetilde{\Theta^{\mathfrak{C}}}$, despite a slower rate of exponential decay. Note that the statistical rate of $\mathrm{err}_{\text{CoT}}$ with a continuous $\widetilde{\Theta^{\mathfrak{C}}}$ does not require Assumption 13, which is required by Theorem 27. The reason is technical: Assumption 9 plays a similar role as Assumption 13 and is sufficient to establish the result.

Proof [Proof of Theorem 10] We split the proof into three parts. First, we apply Proposition 28 and Assumption 9 to derive an upper bound of KL divergence involving $\widetilde{\Theta^{\mathtt{C}}}$. Then in the second and last part, we consider discrete and continuous cases separately.

Step 1: Derive upper bound for KL divergence at an equivalence class level. We fix some concept θ^* as the true latent task parameter. Let $\Theta^{\mathtt{C}} = \Theta \backslash \Theta_{\mathrm{eq}}(\theta^*)$ and $\widetilde{\Theta^{\mathtt{C}}}$ denote a representative set of $\Theta^{\mathtt{C}}$. In light of Proposition 28, we first write

$$\begin{aligned}
&\int_{\Theta^{\mathtt{C}}} \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)\mathrm{d}\theta \\
&\quad = \int_{\theta\in\widetilde{\Theta^{\mathtt{C}}}} \pi\big(\Theta_{\mathrm{eq}}(\theta)\big) \int_{\theta'\in\Theta_{\mathrm{eq}}(\theta)} \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta'\big)\pi(\theta')/\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathrm{d}\theta'\mathrm{d}\theta. \qquad (52)
\end{aligned}$$

That is, we decompose the integral over $\Theta^{\mathtt{C}}$ into a double integral: the inner integral averages the likelihood $\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \mid \theta'\big)$ within each equivalence class, and the outer integral averages across all equivalence classes. By Assumption 9, there exists a representative set $\widetilde{\Theta}$ such that θ^* is in $\widetilde{\Theta}$, and distributions within the same equivalence class are close to each other. Thus, we can derive both upper and lower bounds for the averaged likelihood within each equivalence class. For any $\theta \in \widetilde{\Theta}$, we have

$$\exp(-n\alpha - \alpha_0) \le \int_{\theta'\in\Theta_{\mathrm{eq}}(\theta)} \frac{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta'\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta\big)} \cdot \frac{\pi(\theta')}{\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)}\mathrm{d}\theta' \le \exp(n\alpha + \alpha_0). \qquad (53)$$

Combining (53) and (52) with Proposition 28, we have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,\big|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big) \\
&\quad \le \log\left(1 + \frac{\int_{\theta\in\widetilde{\Theta^{\mathtt{C}}}} \pi\big(\Theta_{\mathrm{eq}}(\theta)\big) \int_{\theta'\in\Theta_{\mathrm{eq}}(\theta)} \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta'\big)\pi(\theta')/\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathrm{d}\theta'\mathrm{d}\theta}{\int_{\theta''\in\Theta_{\mathrm{eq}}(\theta^*)} \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta''\big)\pi(\theta'')\mathrm{d}\theta''}\right) \\
&\quad \le \log\left(1 + \frac{e^{n\alpha+\alpha_0} \cdot \int_{\theta\in\widetilde{\Theta^{\mathtt{C}}}} \pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta\big)\mathrm{d}\theta}{e^{-n\alpha-\alpha_0} \cdot \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta^*\big)\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\right).
\end{aligned}$$

The first inequality follows from Proposition 28 and (52) and the second inequality is due to (53). Note that the representative set $\widetilde{\Theta^{\mathtt{C}}}$ may not be unique, but this does not affect our analysis because we use $\theta \in \widetilde{\Theta^{\mathtt{C}}}$ only as a reference and the value of $\mathbb{P}(\mathtt{prompt}_{\mathrm{CoT}}(n) \,|\, \theta)$ stays the same within each equivalence class. The definition is consistent under any selection of the representative set. Therefore, we rewrite the upper bound for the KL divergence at the equivalence class level:

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,\big|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big) \\
&\quad \le \log\left(1 + e^{2n\alpha+2\alpha_0} \frac{\int_{\theta\in\widetilde{\Theta^{\mathtt{C}}}} \pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta\big)\mathrm{d}\theta}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,\big|\, \theta^*\big)\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\right). \qquad (54)
\end{aligned}$$

Here, Assumption 9 reduces the integration region from $\Theta^{\mathtt{C}}$ to $\widetilde{\Theta^{\mathtt{C}}}$, at the cost of introducing the terms $e^{2\alpha n+2\alpha_0}$.

Step 2: Statistical rate for the discrete case. In this step, we assume the parameter space $\widetilde{\Theta^{\mathtt{C}}}$ is discrete and finite. For any $\theta \in \widetilde{\Theta^{\mathtt{C}}}$, with probability at least $1-\delta$, we obtain an upper bound for the averaged likelihood ratio as follows:

$$\begin{aligned}
&\frac{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)} \\
&\quad\le \exp\bigg(-2n\mathrm{H}^2\big(\mathbb{P}(Z_0,Z_1,\cdots,Z_{H-1},Y\,|\,\theta^*),\mathbb{P}(Z_0,Z_1,\cdots,Z_{H-1},Y\,|\,\theta)\big) \\
&\qquad\quad -2\mathrm{H}^2\big(\mathbb{P}(Z_0\,|\,\theta^*),\mathbb{P}(Z_0\,|\,\theta)\big)+2\log\delta^{-1}\bigg) \\
&\quad\le \exp\big(-2n\lambda+2\log\delta^{-1}\big). \qquad (55)
\end{aligned}$$

Here the first inequality follows from the Lemma 29. The second inequality is due to the Assumption 6, which specifies that θ and θ^* are strictly separated with a margin λ, and the non-negativity of the Hellinger distance. Combining (55) with (54), with probability at least $1-\delta$, we have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big) \\
&\quad\le \log\bigg(1+e^{2n\alpha+2\alpha_0}\cdot\frac{\sum_{\theta\in\widetilde{\Theta^{\mathtt{C}}}}\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\bigg) \\
&\quad\le \log\bigg[1+\frac{e^{2n\alpha+2\alpha_0}}{\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\cdot\sum_{\theta\in\widetilde{\Theta^{\mathtt{C}}}}\exp\big(-2n\lambda+2\log\big(\delta^{-1}|\widetilde{\Theta^{\mathtt{C}}}|\big)\big)\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\bigg] \\
&\quad= \log\bigg[1+\frac{1-\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}{\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\cdot\exp\bigg(-2n(\lambda-\alpha)+2\alpha_0+2\log\big(\delta^{-1}|\widetilde{\Theta^{\mathtt{C}}}|\big)\bigg)\bigg].
\end{aligned}$$

Here the first inequality follows from (54) and the fact that $\widetilde{\Theta^{\mathtt{C}}}$ is discrete and finite. The second inequality is due to (55), and the final line results from rearranging terms.

In sum, we have that when $\widetilde{\Theta^{\mathtt{C}}}$ is discrete and finite, with probability at least $1-\delta$ we have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big) \\
&\quad= \mathcal{O}\big(\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)^{-1}\cdot\delta^{-2}\cdot|\widetilde{\Theta^{\mathtt{C}}}|^2\cdot e^{-2(\lambda-\alpha)n+2\alpha_0}\big), \qquad (56)
\end{aligned}$$

where $\mathcal{O}(\cdot)$ only hides absolute constants. We conclude that when Θ is discrete and finite, the CoT prompting error decays exponentially to zero and it depends on $\Theta^{\mathtt{C}}$ only through $|\widetilde{\Theta^{\mathtt{C}}}|$. The upper bound in (56) establishes Theorem 10.

Step 3: Convergence rate for the continuous case. It remains to consider the case where Θ is continuous. Similar to the proof of Theorem 27, we use the α-cover with respect to the likelihood metric $\|\cdot\|_\Theta$ to discretize $\widetilde{\Theta^{\mathtt{C}}}$ at the cost of introducing an additional error.

For any $\alpha>0$, let $\mathcal{C}(\alpha)$ denote an α-covering of $\widetilde{\Theta^{\mathtt{C}}}$ with covering number $\mathcal{N}(\alpha)$ and let $\{C(\theta_i)\}_{\theta\in\mathcal{N}(\alpha)}$ denote the partition of $\widetilde{\Theta^{\mathtt{C}}}$ induced by $\mathcal{C}(\alpha)$. We bound the discretization

error due to the α-cover as follows:

$$\begin{aligned}
&\log\left(\frac{\int_{\theta\in\widetilde{\Theta^{\mathfrak{C}}}}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathrm{d}\theta}{\sum_{\theta\in\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C_n(\theta)\big)}\right)\\
&\quad=\log\left(\frac{\sum_{\theta\in\mathcal{C}(\alpha)}\pi\big(C(\theta)\big)\int_{\theta'\in C(\theta)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)\pi\big(\Theta_{\mathrm{eq}}(\theta')\big)/\pi\big(C(\theta)\big)\mathrm{d}\theta'}{\sum_{\theta\in\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C(\theta)\big)}\right)\\
&\quad\le\log\left(\max_{\theta\in\mathcal{C}(\alpha)}\frac{\int_{\theta'\in C(\theta)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)\pi\big(\Theta_{\mathrm{eq}}(\theta')\big)/\pi\big(C(\theta)\big)\mathrm{d}\theta'}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}\right)\\
&\quad\le\log\left(\sup_{\theta\in\mathcal{C}(\alpha),\theta'\in C(\theta)}\frac{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta'\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)}\right)\quad\le n\alpha. \qquad (57)
\end{aligned}$$

Here the first equality follows from decomposing the integral over $\widetilde{\Theta^{\mathfrak{C}}}$ into a double integral using the partition structure. The second and third inequalities follow from the generalized mediant inequality. The last inequality is derived from the definition of $\mathcal{C}(\alpha)$.

Now we have controlled the discretization error; combining this with the analysis from **Step 2**, we obtain that the following inequality holds with probability at least $1-\delta$:

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big)\\
&\quad\le\log\left(1+e^{2n\alpha+2\alpha_0}\cdot\frac{\int_{\theta\in\widetilde{\Theta^{\mathfrak{C}}}}\pi\big(\Theta_{\mathrm{eq}}(\theta)\big)\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\mathrm{d}\theta}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\right)\\
&\quad\le\log\left(1+e^{2n\alpha+2\alpha_0}\cdot\frac{\sum_{\theta\in\mathcal{C}(\alpha)}\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta\big)\pi\big(C_n(\theta)\big)}{\mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\,|\,\theta^*\big)\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\right)+n\alpha\\
&\quad\le\log\left[1+\frac{1-\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}{\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)}\cdot\exp\Big(-2n(\lambda-\alpha)+2\alpha_0+2\log\big(\delta^{-1}\mathcal{N}(\alpha)\big)\Big)\right]+n\alpha,
\end{aligned}$$

where $\mathcal{O}(\cdot)$ only hides absolute constants, and the randomness comes from the stochasticity of CoT prompts $\mathtt{prompt}_{\mathrm{CoT}}(n)$. The first inequality follows from (54), and the second inequality follows from (57). The final inequality is due to Lemma 29. Therefore, we conclude that with probability at least $1-\delta$ we have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}}=\cdot\,|\,z_0^{\mathrm{test}},\theta^*),\mathbb{P}\big(y^{\mathrm{test}}=\cdot\,|\,\mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big)\\
&\qquad=\mathcal{O}\big(\delta^{-2}\cdot\pi\big(\Theta_{\mathrm{eq}}(\theta^*)\big)^{-1}\cdot\mathcal{N}(\alpha)^2\cdot e^{-2n(\lambda-\alpha)+2\alpha_0}+n\alpha\big),
\end{aligned}$$

where $\mathcal{O}(\cdot)$ omits absolute constants. Thus, we conclude the proof. ■

Appendix F. Proofs of the Results in Section 7

In this appendix, we prove the statistical error bounds for the three CoT variants discussed in Section 7: self-consistency CoT, tree-of-thought, and selection-inference.

F.1 Proof of Corollary 21

Proof

There are two notions of sample size in self-consistency CoT: the number of examples in CoT prompt n, and the number of reasoning paths K. These two notions have different roles. A large n ensures that the distribution of the perfectly pretrained LLM, $\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)$, approximates the desired distribution $\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*\big)$. Whereas a large K ensures that the sample mode y_k^* approximates the population mode.

In the following, we prove the corollary in two steps. We first show that the population mode of $\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)$ coincides with y^* when n is sufficiently large. Then we prove that y_K^* finds the population mode of $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ when K is sufficiently large. The final statistical error can be obtained by combining these two steps.

Step 1: Mode of $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ converges to y^*. In the first step, we show that there exists n^* such that, as long as $n \geq n^*$, the mode of $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ coincides with y^*, the mode of $\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*\big)$. This step leverages Theorem 7.

Suppose there exists some $\xi > 0$ such that

$$\texttt{KL}\Big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\Big) \leq \xi,$$

Then by Pinsker's inequality, we obtain a bound on the TV distance,

$$\texttt{TV}\Big((\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\Big) \leq \sqrt{2\xi}.$$

Recall that we assume that y^{test} belongs to a finite set $\mathcal{Y}$. Then we have

$$\begin{aligned}
\sqrt{2\xi} &\geq \texttt{TV}\Big((\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\Big) \\
&= \frac{1}{2} \sum_{y \in \mathcal{Y}} \Big| \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) - \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \Big| \\
&\geq \frac{1}{2} \max_{y \in \mathcal{Y}} \Big| \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) - \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \Big|.
\end{aligned}$$

Thus we can sandwich $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$ for any y by

$$\mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) - 2\sqrt{2\xi} \leq \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \leq \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) + 2\sqrt{2\xi}.$$

We plug in different values of y in the above inequality and obtain

$$\max_{y \neq y^*} \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \leq \max_{y \neq y^*} \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) + 2\sqrt{2\xi}, \tag{58}$$

$$\mathbb{P}\big(y^{\text{test}} = y^* \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \geq \mathbb{P}(y^{\text{test}} = y^* \,|\, z_0^{\text{test}}, \theta^*) - 2\sqrt{2\xi}. \tag{59}$$

Recall the definition of ϵ in Assumption 20. Combining (58) and (59) we have

$$\begin{aligned}\mathbb{P}\big(y^{\text{test}} = y^* \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) &\geq \mathbb{P}(y^{\text{test}} = y^* \,|\, z_0^{\text{test}}, \theta^*) - 2\sqrt{2\xi} \\ &\geq \max_{y \neq y^*} \mathbb{P}(y^{\text{test}} = y \,|\, z_0^{\text{test}}, \theta^*) + \epsilon - 2\sqrt{2\xi} \\ &\geq \max_{y \neq y^*} \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) + \epsilon - 4\sqrt{2\xi}, \end{aligned} \tag{60}$$

where the first inequality follows from (58), the second follows from the definition of ϵ, and the last one follows from (59). Hence, as long as $\epsilon - 4\sqrt{2\xi} > 0$, we ensure that y^* is also the unique mode of $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$. Now if we set the prompt size to be

$$n^* = C \cdot \Big(\log\big(|\Theta^{\mathfrak{L}}|/\pi(\theta^*)\big) + \log(1/\epsilon) \Big) \Big/ \lambda, \tag{61}$$

where C is a sufficiently large absolute constant. We now leverage Theorem 7 with a perfectly pretrained LLM. Specifically, setting $\delta = e^{-\lambda n/2}$ in (49), we conclude that, with probability at least $1 - e^{-\lambda n/2}$, when $n \geq n^*$, we have

$$\texttt{KL}\Big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \Big) \leq \epsilon^2/128. \tag{62}$$

Combining (60) and (62), we conclude that, when n is sufficiently large such that $n \geq n^*$, with probability $1 - e^{-\lambda n/2}$,

$$\mathbb{P}\big(y^{\text{test}} = y^* \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) \geq \max_{y \neq y^*} \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big) + \epsilon/2, \tag{63}$$

where n^* is defined in (61). Thus, y^* is also the mode of $\mathbb{P}(y^{\text{test}} = y^* \,|\, \texttt{prompt}_{\text{CoT}}(n))$ with high probability. Now we conclude **Step 1**.

Step 2: Sample mode y_K^* converges to population mode y^*. In this step, we use concentration to show that y_K^* converges to y^* when K is sufficiently large. Recall that we assume y^{test} takes values in a finite set $\mathcal{Y}$ and also recall that we define an empirical distribution $p_K(y) = K^{-1} \sum_{i=1}^K \mathbf{1}\{y^{\text{test},i} = y\}$ for all $y \in \mathcal{Y}$. Thus, for any $y \in \mathcal{Y}$, $Kp_K(y)$ is a binomial variable with distribution $\texttt{Bin}\big(K, p\big)$, where $p = \mathbb{P}(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n))$. Thus by Bernstein's inequality for binomial distribution, for any $t > 0$ we have

$$\begin{aligned}&\mathbb{P}\big(\big|p_K(y) - \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\big| \geq t\big) \\ &\quad \leq 2\exp\bigg(- \frac{3Kt^2}{6 \cdot \mathbb{P}\big(y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\big(1 - \mathbb{P}(y \,|\, \texttt{prompt}_{\text{CoT}}(n))\big) + 2t} \bigg) \leq 2\exp\bigg(- \frac{6Kt^2}{3 + 4t} \bigg)\end{aligned}$$

where the second inequality follows from the fact that $p(1-p) \leq 1/4$ for any $p \in \mathbb{R}$. Now we set $t = \epsilon/4$ and take a union bound over $y \in \mathcal{Y}$ to obtain that

$$\mathbb{P}\Big(\max_{y \in \mathcal{Y}} \big|p_K(y) - \mathbb{P}\big(y^{\text{test}} = y \,|\, \texttt{prompt}_{\text{CoT}}(n)\big)\big| \geq \epsilon/4 \Big) \leq 2|\mathcal{Y}| \cdot \exp\bigg(- \frac{3K\epsilon^2}{24 + 8\epsilon} \bigg). \tag{64}$$

Recall that y_K^* and y^* are the modes of p_K and $\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))$, respectively. Also note that (63) implies that there is a gap of $\epsilon/2$ between the mode of $\mathbb{P}(y^{\text{test}} =$

$\cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))$ and its second largest probability mass. If $y_K^* \neq y^*$, there must exist some $y \in \mathcal{Y}$ such that

$$|p_K(y) - \mathbb{P}\big(y^{\mathrm{test}} = y \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)| \geq \epsilon/4.$$

Therefore, we can upper bound $\mathbb{P}\big(y_K^* \neq y^* \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big)$ using (64).

Therefore, we conclude that with a perfectly pretrained LLM and a discrete and finite Θ, when is sufficiently large such that $n \geq n^*$, we have

$$\mathbb{P}\big(y_K^* \neq y^* \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) \leq 2|\mathcal{Y}| \cdot \exp\bigg(-\frac{3K\epsilon^2}{24+8\epsilon}\bigg)$$

with probability at least $1-e^{-\lambda n/2}$. Here n^* is define in (61). Thus we conclude the proof. ■

F.2 Proof of Proposition 23

Proof

In ToT prompting, there are two notions of sample size: the number of examples in the CoT prompt n and the number of candidates generated at each step K. Additionally, there is a breadth limit parameter B, which controls the number of candidates that continue to the next step. For the ease of notation, we write $z_h^{\mathrm{test},*}$ as z_h^* and $t_h^{\mathrm{test},*}$ as t_h^*.

Intuitively, a large n ensures that the distribution induced by a perfectly pretrained LLM $\mathbb{P}(z_h^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_h(n), t_{h-1}^*)$ approximates the true distribution $\mathbb{P}(z_h^{\mathrm{test}} = \cdot \,|\, t_{h-1}^*, \theta^*)$ for each $h \in [H]$. A large K ensures that the optimal z_h^* appears in the samples for each $h \in [H]$, which is then selected by BFS.

The proof involves two steps. First, we show that for any $h \in [H]$, $p_h = \mathbb{P}(z_h^{\mathrm{test}} = z_h^* \mid \texttt{prompt}_h(n), t_{h-1}^*)$ is close to $p_h^* = \mathbb{P}(z_h^{\mathrm{test}} = z_h^* \mid t_{h-1}^*, \theta^*)$ when n is large enough, where z_h^* is the optimal next step $z_h^* = \mathrm{argmax}_{z_h} V_{\theta^*}(t_{h-1}^*, z_h)$. Second, we demonstrate that a large K helps find the optimal trajectory t_H^* by iteratively sampling from the LLM-induced distribution $\mathbb{P}(z_h^{\mathrm{test}} = \cdot \mid \texttt{prompt}_h(n), t_{h-1}^*)$. Finally, we combine both arguments to present the statistical error for ToT prompting.

Step 1: LLM-induced probability p_h approximates p_h^* for large n. In this step, we show p_h approaches p_h^* as n increases, which is similar to Step 1 of the proof of Theorem 7.

For each $h \in [H]$, $\texttt{prompt}_h(n)$ contains the CoT examples in terms of the prediction in the h-th step. We want to lower bound the probability of outputting z_h^* by prompting a perfectly pretrained LLM. For simplicity, we assume Θ is finite and discrete. For each $h \in [H]$, using Bayes rule, we write the posterior as

$$\pi(\theta^* \,|\, \texttt{prompt}_h(n), t_{h-1}^*) = \bigg(1 + \sum_{\theta \neq \theta^*} \frac{\mathbb{P}(\texttt{prompt}_h(n) \,|\, \theta)\mathbb{P}(t_{h-1}^* \,|\, \theta)\pi(\theta)}{\mathbb{P}(\texttt{prompt}_h(n) \,|\, \theta^*)\mathbb{P}(t_{h-1}^* \,|\, \theta^*)\pi(\theta^*)}\bigg)^{-1}.$$

Under Assumption 22, θ^* maximizes $\mathbb{P}(t_h^* \,|\, \theta)$ for each $h \in [H]$. Thus, we have

$$\begin{aligned}\pi(\theta^* \,|\, \texttt{prompt}_h(n), t_{h-1}^*) &\geq \bigg(1 + \sum_{\theta \neq \theta^*} \frac{\mathbb{P}(\texttt{prompt}_h(n) \,|\, \theta)\pi(\theta)}{\mathbb{P}(\texttt{prompt}_h(n) \,|\, \theta^*)\pi(\theta^*)}\bigg)^{-1} \\ &\geq \bigg(1 + \sum_{\theta \neq \theta^*} \frac{\pi(\theta)}{\pi(\theta^*)} \cdot \exp\big[-2n\mathrm{H}^2\big(\mathbb{P}(t_h \,|\, \theta), \mathbb{P}(t_h \,|\, \theta^*)\big) + 2\log(\delta^{-1}|\Theta|)\big]\bigg)^{-1} \\ &\geq \bigg(1 + \frac{1-\pi(\theta^*)}{\pi(\theta^*)} \cdot \exp\big(-2n\lambda_h + 2\log(\delta^{-1}|\Theta|)\big)\bigg)^{-1} \qquad (65)\end{aligned}$$

with probability at least $1-\delta$. Here the second inequality follows from Lemma 29 in the proof of Theorem 7 and the third inequality follows from Assumption 22. Thus we derive a lower bound for p_h as

$$\begin{aligned}p_h &= \sum_{\theta \in \Theta} \mathbb{P}(z_h^* \,|\, t_{h-1}^*, \theta)\pi(\theta \,|\, t_{h-1}^*, \texttt{prompt}_h(n)) \geq \mathbb{P}(z_h^* \,|\, t_{h-1}^*, \theta^*)\pi(\theta^* \,|\, t_{h-1}^*, \texttt{prompt}_h(n)) \\ &\geq p_h^* \cdot \bigg(1 + \frac{1-\pi(\theta^*)}{\pi(\theta^*)} \cdot \exp\big(-2n\lambda_h + 2\log(\delta^{-1}|\Theta|)\big)\bigg)^{-1}.\end{aligned}$$

with probability at least $1-\delta$ with respect to the randomness of the CoT prompt. The first inequality follows from omitting terms corresponding to $\theta \neq \theta^*$. The second inequality is due to (65). Now we use the fact that $(1+x)^{-1} \geq 1-x$ for $x \in [0,1]$ to obtain that

$$p_h^* - p_h \leq p_h^* \cdot \frac{1-\pi(\theta^*)}{\pi(\theta^*)} \cdot \exp\big(-2n\lambda_h + 2\log(\delta^{-1}|\Theta|)\big). \qquad (66)$$

Therefore, we conclude that $p_h^* - p_h$, the difference in the probabilities evaluated at z_h^* for LLM-induced distribution and the true distribution, decreases exponentially with the number of examples n. Recall that we define $\lambda^* = \min_{h \in [H]} \lambda_h$. For any $\epsilon \in (0,1)$, we define

$$n^* = \frac{1}{\lambda^*}\bigg(2\log\big(H|\Theta|\big) + \log\big((1-\pi(\theta^*))/\pi(\theta^*)\big) + \log(1/\epsilon)\bigg). \qquad (67)$$

Then by taking $\delta = e^{-n\lambda^*/2}$ and combining (66), we conclude that with probability at least $1 - e^{-n\lambda^*/2}$, when $n \geq n^*$, $p_h \geq p_h^* \cdot (1-\epsilon)$ holds for every $h \in [H]$. Thus we conclude **Step 1**.

Step 2: Large K improves the selection of z_h^*. For any $h \in [H]$, let

$$C_h = \sum_{b=1}^{B} \sum_{i=1}^{K} \mathbf{1}\{(t_{h-1}^b, z_h^{b,i}) = t_h^*\}$$

denote the number of candidates in $\mathcal{T}_h$ that match the optimal partial history t_h^* at step $h \in [H]$, where we define $\mathcal{T}_h$ in (13). Recall that we set $B = 1$. Therefore, each C_h, $h \in [H]$ is a binomial random variable, where

$$C_1 \sim \texttt{Bin}(K, p_1), \qquad C_h \sim \texttt{Bin}(\min\{B=1, C_{h-1}\}K, p_h) \text{ for } 2 \leq h \leq H.$$

The algorithm outputs t_H^* if and only if $C_h \geq 1$ for all $h \in [H]$. The following gives the probability of outputting the optimal trajectory from the search tree:

$$\begin{aligned}\mathbb{P}\big(\widehat{t}_H = t_H^* \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) &= \mathbb{P}\big(C_1, \cdots, C_H \geq 1 \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) \\ &\geq \prod_{h=1}^{H} \mathbb{P}\big(C_h \geq 1 \,|\, \texttt{prompt}_h(n), t_{h-1}^*\big) \\ &= \prod_{h=1}^{H} \big(1 - (1-p_h)^K\big) \\ &\geq 1 - \sum_{h=1}^{H} \big(1-p_h\big)^K. \end{aligned} \tag{68}$$

The last inequality is because $\prod_{i=1}^m (1-x_i) \geq 1 - \sum_{i=1}^m x_i$ for $x_i \in [0,1]$.

Combining (68) and the conclusion of **Step 1**, we conclude that with a perfectly pre-trained LLM and a discrete and finite Θ, ToT prompting using BFS with $B = 1$ incurs the following statistical error with probability at least $1 - e^{-n\lambda^*/2}$:

$$\mathbb{P}\big(\widehat{t}_H \neq t_H^* \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)\big) \leq \sum_{h=1}^{H} \Big(1 - p_h^* + p_h^* \epsilon\Big)^K.$$

Here n is sufficiently large such that $n \geq n^*$ where n^* is defined in (67) and $\epsilon \in (0,1)$ is an arbitrary number. Therefore, we conclude the proof. ■

F.3 Proof of Corollary 25

Proof The main idea of selection-inference prompting is to break down each step in vanilla CoT into two separate stages: selection and inference. We decompose each task θ into two components: θ_{se} and θ_{in}, with underlying distributions $\mathbb{P}(\tau_h = \cdot \,|\, t_{h-1}, \theta_{\mathrm{se}})$ and $\mathbb{P}(z_h = \cdot \,|\, \tau_h, \theta_{\mathrm{in}})$, respectively. We follow and modify the proof for vanilla CoT in Appendix E.2 to derive the statistical rate of SI prompting.

The proof consists of two steps. We first derive an upper bound for the KL divergence using a ratio of two integrals. Then we bound such a ratio using the separation among the probability distributions with different parameters in Θ.

Step 1: Deriving an upper bound for the KL divergence. We invoke Proposition 28 to derive an upper bound for the KL divergence as follows:

$$\begin{aligned}&\mathtt{KL}\bigg(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}_{\mathrm{SI}}\big(y^{\mathrm{test}} = \cdot \,|\, S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}}\big)\bigg) \\ &\leq \log\left(1 + \frac{\sum_{\Theta^{\complement}} \mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta\big)\pi(\theta)\mathrm{d}\theta}{\sum_{\Theta_{\mathrm{eq}}(\theta^*)} \mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta'\big)\pi(\theta')\mathrm{d}\theta'}\right) \\ &\leq \log\left(1 + \frac{\sum_{\Theta^{\complement}} \mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta\big)\pi(\theta)}{\mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta^*\big)\pi(\theta^*)}\right). \end{aligned} \tag{69}$$

The first inequality is obtained by applying Proposition 28 with substituting $\texttt{prompt}_{\mathrm{CoT}}(n)$ by $(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}})$, and in the second inequality we exclude all terms corresponding to $\theta \neq \theta^*$ in the denominator. We can directly apply Proposition 28 since this proposition only requires the prompt to be generated from $\mathbb{P}(\cdot \,|\, \theta^*)$, but does not assume a specific statistical dependency relationship in the prompt. Here $\mathbb{P}_{\mathrm{SI}}(y^{\mathrm{test}} \,|\, S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}})$ is the marginal distribution of y^{test} according to (16).

Step 2: Statistical rate for discrete and finite Θ. The key to analyzing (69) is to derive an upper bound of the likelihood ratio

$$\mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta\big) / \mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}} \,\big|\, \theta^*\big).$$

Recall that we define $S_{\mathrm{se}}(n)$ and $S_{\mathrm{in}}(n)$ in (15). By construction, the likelihood of a single piece of trajectory can be decomposed as

$$\begin{aligned}
&\mathbb{P}\big(S_{\mathrm{se}}(1), S_{\mathrm{in}}(1) \,\big|\, \theta\big) \\
&\quad = \mathbb{P}(z_0 \,|\, \theta) \prod_{h=1}^{H} \mathbb{P}\big(\tau_h \,\big|\, \theta_{\mathrm{se}}, t_{h-1}\big) \cdot \prod_{j=1}^{H} \mathbb{P}\big(z_j \,\big|\, \theta_{\mathrm{in}}, \tau_j\big).
\end{aligned} \tag{70}$$

Note that the n reasoning paths are independent conditioning on θ. We decompose the likelihood ratios into a sum of independent terms according to (70) and then apply Lemma 50. With probability at least $1-\delta$, we have

$$\begin{aligned}
&\frac{1}{2} \log \frac{\mathbb{P}(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n) \,|\, \theta)}{\mathbb{P}(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n) \,|\, \theta^*)} \\
&\quad = \frac{1}{2} \sum_{i=1}^{n} \left[\log \frac{\mathbb{P}(z_0^i \,|\, \theta)}{\mathbb{P}(z_0^i \,|\, \theta^*)} + \sum_{h=1}^{H} \log \frac{\mathbb{P}\big(\tau_h^i \,\big|\, \theta_{\mathrm{se}}, t_{h-1}^i\big)}{\mathbb{P}\big(\tau_h^i \,\big|\, \theta_{\mathrm{se}}^*, t_{h-1}^i\big)} + \sum_{j=1}^{H} \log \frac{\mathbb{P}\big(z_j^i \,\big|\, \theta_{\mathrm{in}}, \tau_j^i\big)}{\mathbb{P}\big(z_j^i \,\big|\, \theta_{\mathrm{in}}^*, \tau_j^i\big)} \right] \\
&\quad \leq \sum_{i=1}^{n} \log \left[\mathbb{E}_{\theta^*} \left(\frac{\mathbb{P}(z_0^i \,|\, \theta)}{\mathbb{P}(z_0^i \,|\, \theta^*)} \right)^{1/2} \right] + \sum_{h=1}^{H} \sum_{k=1}^{n} \log \left[\mathbb{E}_{\theta^*} \left(\frac{\mathbb{P}(\tau_h^k \,|\, \theta_{\mathrm{se}}, t_{h-1}^k)}{\mathbb{P}(\tau_h^k \,|\, \theta_{\mathrm{se}}^*, t_{h-1}^k)} \right)^{1/2} \right] \\
&\qquad + \sum_{j=1}^{H} \sum_{m=1}^{n} \log \left[\mathbb{E}_{\theta^*} \left(\frac{\mathbb{P}\big(z_j^m \,\big|\, \theta_{\mathrm{in}}, \tau_j^m\big)}{\mathbb{P}\big(z_j^m \,\big|\, \theta_{\mathrm{in}}^*, \tau_j^m\big)} \right)^{1/2} \right] + \log((2H+1)\delta^{-1}).
\end{aligned} \tag{71}$$

Here the first equality follows from summing over the decomposition of likelihood ratios as shown in (70), and inequality follows is obtained by applying Lemma 50 to each sum. Using

(71) and the fact that $x-1 \geq \log(x)$, we further obtain that

$$\begin{aligned}
&\frac{1}{2}\log\frac{\mathbb{P}(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n)\,|\,\theta)}{\mathbb{P}(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n)\,|\,\theta^*)} \\
&\quad\leq \sum_{i=1}^{n}\left[\mathbb{E}_{\theta^*}\left(\frac{\mathbb{P}(z_0^i\,|\,\theta)}{\mathbb{P}(z_0^i\,|\,\theta^*)}\right)^{1/2} - 1 + \sum_{h=1}^{H}\left(\mathbb{E}_{\theta^*}\frac{\mathbb{P}(\tau_h^i\,|\,\theta_{\mathrm{se}}, t_{h-1}^i)}{\mathbb{P}(\tau_h^i\,|\,\theta_{\mathrm{se}}^*, t_{h-1}^i)}\right)^{1/2} - 1\right) \\
&\qquad + \sum_{j=1}^{H}\left(\mathbb{E}_{\theta^*}\left(\frac{\mathbb{P}\big(z_j^i\,\big|\,\theta_{\mathrm{in}}, \tau_j^i\big)}{\mathbb{P}\big(z_j^i\,\big|\,\theta_{\mathrm{in}}^*, \tau_j^i\big)}\right)^{1/2} - 1\right)\Bigg] + \log\big((2H+1)\delta^{-1}\big) \\
&\quad= -n\sum_{h=1}^{H}\Big[\mathbb{E}_{\theta^*}\mathrm{H}^2\big(\mathbb{P}(\tau_h\,|\,\theta^*, t_{h-1}), \mathbb{P}(\tau_h\,|\,\theta, t_{h-1})\big) \\
&\qquad + \mathbb{E}_{\theta^*}\mathrm{H}^2\big(\mathbb{P}(z_h\,|\,\theta^*, \tau_h), \mathbb{P}(z_h\,|\,\theta, \tau_h)\big)\Big] - n\mathrm{H}^2\big(\mathbb{P}(z_0\,|\,\theta^*), \mathbb{P}(z_0\,|\,\theta)\big) \\
&\qquad + \log\big((2H+1)\delta^{-1}\big). \qquad (72)
\end{aligned}$$

Here the final line follows from the definition of Hellinger distance.

Now we apply Assumption 24 to derive an upper bound of the likelihood ratio using the constant $\lambda_{\mathrm{SI}} = \lambda_{\mathrm{S}} + \lambda_{\mathrm{I}} + \lambda_{\mathrm{q}}$, replacing the constant λ from Theorem 7. Combining (72) and Assumption 24, we have that when Θ is discrete and finite,

$$\begin{aligned}
&\frac{\mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}}\,|\,\theta\big)}{\mathbb{P}\big(S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}}\,|\,\theta^*\big)} \\
&\quad\leq \exp\big(-2n\lambda_{\mathrm{SI}} + 2\log\big((2H+1)\delta^{-1}\big)\big) \text{ with probability at least } 1-\delta.
\end{aligned}$$

Therefore, we have that when $\Theta^{\complement}$ is discrete and finite, then with probability at least $1-\delta$,

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot\,|\,z_0^{\mathrm{test}}, \theta^*), \mathbb{P}_{\mathrm{SI}}\big(y^{\mathrm{test}} = \cdot\,|\,S_{\mathrm{se}}(n), S_{\mathrm{in}}(n), z_0^{\mathrm{test}}\big)\Big) \\
&\quad= \mathcal{O}\big(\pi(\theta^*)^{-1}\cdot\delta^{-2}\cdot|\Theta^{\complement}|^2\cdot e^{-2\lambda_{\mathrm{SI}}n}\big),
\end{aligned}$$

where $\mathcal{O}(\cdot)$ omits only absolute constants. Therefore, we conclude the proof. ■

Appendix G. Proofs and Auxiliary Results of Section 5.2

In this appendix, we collect the proofs and auxiliary results supporting the comparison between vanilla CoT and vanilla ICL presented in Section 5.2.

G.1 Proof of a Generalized Version of Proposition 11

In the following, we generalize Proposition 11 to handle comparisons of different truncated CoT methods, which covers vanilla ICL as a special case. For simplicity, we assume zero pretraining error and input query does not have a distributional shift, i.e., $\mathbb{P}_{\mathrm{LLM}} = \mathbb{P}$ and $z_0^{\mathrm{test}} \sim \mathbb{P}(\cdot \,|\, \theta^*)$. Before presenting the result, we state the following regularity assumption.

Assumption 31 *For a fixed CoT prompt* $\texttt{prompt}_{\mathrm{CoT}}(n)$*, we define truncated CoT prompts with fixed intermediate step indices as* $\texttt{prompt}_{\mathcal{J}}(n) = \{z_0^i, y^i\}_{i=1}^n \cup \{z_j^i\}_{j\in\mathcal{J}} \cup \{z_0^{\mathrm{test}}\}$ *with fixed index set* $\mathcal{J} \subseteq [H-1]$*, where* $\texttt{prompt}_{\mathrm{CoT}}(n) = \{z_0^i, \cdots, z_H^i\}_{i=1}^n \cup \{z_0^{\mathrm{test}}\}$*. We assume that* $\mathbb{P}\big(y^{\mathrm{test}} \,|\, \texttt{prompt}_{\mathcal{J}}(n)\big)$ *is a mixture of* $\mathbb{P}\big(y^{\mathrm{test}} \,|\, \theta, z_0^{\mathrm{test}}\big)$ *aggregated with respect to the posterior of* θ *based on truncated CoT demonstrations. Specifically, the density* $\mathbb{P}\big(y^{\mathrm{test}} \,|\, \theta, z_0^{\mathrm{test}}\big)$ *is obtained by marginalizing the omitted intermediate steps* $z_{i\notin\mathcal{J}}$ *from the joint distribution induced by the CoT model* (2)*.*

The truncated CoT method recovers CoT by setting $\mathcal{J} = [H-1]$ and vanilla ICL by setting $\mathcal{J} = \emptyset$. This assumption ensures that the estimators induced by different truncated CoT methods are comparable. This can be achieved by training LLMs using a truncated dataset and then prompted via the truncated CoT. Specifically, each $\mathcal{J}$, we define $\mathcal{D}_{\mathcal{J}}$ as a truncated version of the CoT data $\mathcal{D}$ obtained by omitting intermediate steps with indices $j \notin \mathcal{J}$. Then we can pretrain an LLM using $\mathcal{D}_{\mathcal{J}}$ using the same MLE loss as in (3), and then prompt the learned model using $\texttt{prompt}_{\mathcal{J}}(n)$. Assumption 31 requires LLMs to process different truncated CoT prompts by making posterior inferences based on their respective pretraining data. This is the case when the LLM is pretrained using $\mathcal{D}_{\mathcal{J}}$, following a similar argument as in Section 4.3. Based on this assumption, we establish a hierarchy of CoT methods in terms of statistical error.

Corollary 32 (Comparison of CoT Methods) *Let* π *represent the task distribution over space* Θ*. Let* $\texttt{prompt}_{\mathrm{CoT}}(n)$ *be a CoT prompt. Given* $\mathcal{J} \subseteq \mathcal{J}' \subseteq [H-1]$*, define* $\texttt{prompt}_{\mathcal{J}}(n) = \{z_0^i, y^i\}_{i=1}^n \cup \{z_j^i\}_{j\in\mathcal{J}} \cup \{z_0^{\mathrm{test}}\}$ *and* $\texttt{prompt}_{\mathcal{J}'}(n) = \{z_0^i, y^i\}_{i=1}^n \cup \{z_j^i\}_{j\in\mathcal{J}'} \cup \{z_0^{\mathrm{test}}\}$*, where* $\texttt{prompt}_{\mathrm{CoT}}(n) = \{z_0^i, \cdots, z_H^i\}_{i=1}^n \cup \{z_0^{\mathrm{test}}\}$*. Under the Assumption 31, for any number of examples* $n \geq 0$*, we have*

$$\begin{aligned}
&\mathbb{E}_{\theta^*\sim\pi}\mathbb{E}_{\texttt{prompt}_{\mathcal{J}'}(n)\sim\mathbb{P}(\cdot\,|\,\theta^*)}\Big[\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathcal{J}'}(n)\big)\Big)\Big] \\
&\quad \leq \mathbb{E}_{\theta^*\sim\pi}\mathbb{E}_{\texttt{prompt}_{\mathcal{J}}(n)\sim\mathbb{P}(\cdot\,|\,\theta^*)}\Big[\mathtt{KL}\Big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathcal{J}}(n)\big)\Big)\Big].
\end{aligned}$$

We notice that such an inequality only holds in an average case by taking an expectation with respect to $\theta^* \sim \pi$*.*

G.2 Proof of Corollary 32

Proof

To simplify the notation, let $\mathbb{E}_{\theta^*}$ denote $\mathbb{E}_{\texttt{prompt}_{\mathrm{CoT}}(n)\sim\mathbb{P}(\cdot\,|\,\theta^*)}$. First, we compute the difference of KL divergences of two index sets $\mathcal{J}$ and $\mathcal{J}'$ on a fixed task θ^*:

$$\begin{aligned}
&\mathbb{E}_{\theta^*}\left[\texttt{KL}\big(\mathbb{P}(y^{\text{test}}=\cdot\,|\,z_0^{\text{test}},\theta^*),\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)\big)\right]\\
&\qquad-\mathbb{E}_{\theta^*}\left[\texttt{KL}\big(\mathbb{P}(y^{\text{test}}=\cdot\,|\,z_0^{\text{test}},\theta^*),\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)\big)\right]\\
&=\mathbb{E}_{\theta^*}\mathbb{E}_{y^{\text{test}}\sim\mathbb{P}(\cdot\,|\,z_0^{\text{test}},\theta^*)}\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]. \qquad (73)
\end{aligned}$$

Next, we take expectation of (73) with respect to $\theta^*\sim\pi$ to obtain

$$\begin{aligned}
&\mathbb{E}_{\pi}\mathbb{E}_{\theta^*}\mathbb{E}_{y^{\text{test}}\sim\mathbb{P}(\cdot\,|\,z_0^{\text{test}},\theta^*)}\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]\\
&=\int_{\mathcal{L}\times\mathcal{L}^*}\mathbb{P}\big(\texttt{prompt}_{\mathcal{J}'}(n)\big)\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]\\
&\quad\cdot\left(\int_{\Theta}\pi\big(\theta^*\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)\mathbb{P}(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n),\theta^*)\mathrm{d}\theta^*\right)\mathrm{d}y^{\text{test}}\mathrm{d}\texttt{prompt}_{\mathcal{J}'}(n). \qquad (74)
\end{aligned}$$

Applying the Bayes' rule, we have

$$\begin{aligned}
&\int_{\mathcal{L}}\left(\int_{\Theta}\pi\big(\theta^*\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)\cdot\mathbb{P}(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n),\theta^*)\cdot\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]\mathrm{d}\theta^*\right)\mathrm{d}y^{\text{test}}\\
&=\mathbb{E}_{y^{\text{test}}\sim\mathbb{P}(\cdot\,|\,\texttt{prompt}_{\mathcal{J}'}(n))}\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]. \qquad (75)
\end{aligned}$$

Thus, by (75) and interchanging the order of integration in (74), we have

$$\begin{aligned}
&\mathbb{E}_{\pi}\mathbb{E}_{\theta^*}\mathbb{E}_{y^{\text{test}}\sim\mathbb{P}(\cdot\,|\,z_0^{\text{test}},\theta^*)}\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]\\
&=\mathbb{E}_{\texttt{prompt}_{\mathcal{J}'}(n)\sim\mathbb{P}}\mathbb{E}_{y^{\text{test}}\sim\mathbb{P}(\cdot\,|\,\texttt{prompt}_{\mathcal{J}'}(n))}\left[\log\frac{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big)}{\mathbb{P}\big(y^{\text{test}}\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)}\right]\\
&=\mathbb{E}_{\mathbb{P}}\left[\texttt{KL}\big(\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\texttt{prompt}_{\mathcal{J}'}(n)\big),\mathbb{P}\big(y^{\text{test}}=\cdot\,|\,\texttt{prompt}_{\mathcal{J}}(n)\big)\right]\geq 0. \qquad (76)
\end{aligned}$$

Therefore, the expectation of the difference between the two KL divergences in (74) can be expressed as (76), another KL divergence and thus is nonnegative for any number of samples n. Therefore, we conclude that on average, truncated CoT methods with steps $\mathcal{J}'\subseteq[H-1]$ is no worse than truncated CoT methods with fewer intermediate steps $\mathcal{J}\subset\mathcal{J}'$. Since vanilla ICL corresponds to the special case where $\mathcal{J}=\emptyset$, we conclude that on average, CoT is no worse than vanilla ICL. This completes the proof. ■

Appendix H. Proof and Auxiliary Results of Section 6

In this section, we prove the results in Section 6. In particular, in Section H.1 we prove Proposition 14, in Section H.2 we introduce the details of the approximation error analysis for pretraining, and in Section H.4 we prove Corollary 18.

H.1 Proof of Proposition 14

In the following, we prove Proposition 14 under the generalized multi-step latent variable model introduced in Section C, which contains the model in (2) as a special case.

Proof In this proof, we adopt the PAC-Bayes framework (McAllester, 1998; Alquier, 2021) to decompose the error and control each component. This proof consists of two steps. We first decompose the pretraining error into three parts using the PAC-Bayes framework, and then control each term to conclude the proof. Our proof is adapted from Zhang et al. (2023a), which analyzes the generalization error for pretraining an LLM based on ICL data, i.e., $H = 1$. We explain the structure of the proof in detail and highlight the similarities and differences from Zhang et al. (2023a).

Step 1: Error decomposition using the PAC-Bayes framework. We first decompose the pretraining error $\mathbb{E}_{\mathcal{D}}\mathtt{TV}\big(\mathbb{P}(\cdot\,|\,S), \mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S)\big)^2$ to prepare for further analysis. Recall that the pretraining dataset with N trajectories with T examples is $\mathcal{D}_{N,T} = \{(S_h^{t,\ell}, z_h^{t,\ell})\}_{h=0,t=1,\ell=1}^{H,T,N}$, where $z_h^{t,\ell} \sim \mathbb{P}(\cdot\,|\,S_h^{t,\ell})$, and $S_{h+1}^{t,\ell} = S_h^{t,\ell} \cup \{z_h^{t,\ell}\}$. Under the construction of $\mathcal{D}_{N,T}$ under the general model in (20), the training data admits a sequential structure. For any $h \in \{0, \ldots, H\}$, $t \in [T]$ and $\ell \in [N]$, we let $\mathcal{F}_h^{t,\ell}$ denote a σ-algebra defined as

$$\mathcal{F}_h^{t,\ell} = \sigma\text{-}\mathtt{algebra}\Big(\big\{z_{h'}^{t',\ell'} : \ell < \ell, \text{or } \ell' = \ell, t' < t, \text{or } \ell' = \ell, t' = t, h' < h\big\}\Big),$$

which is the σ-algebra generated by all the random variables in $\mathcal{D}_{N,T}$ appearing before $z_h^{t,\ell}$. Moreover, we define a sequence of ghost samples as $\widetilde{\mathcal{D}}_{N,T} = \{(\widetilde{S}_h^{t,\ell}, \widetilde{z}_h^{t,\ell})\}_{h=0,t=1,\ell=1}^{H,T,N}$, where $\widetilde{z}_h^{t,\ell} \sim \mathbb{P}(\cdot\,|\,\widetilde{S}_h^{t,\ell})$, and $\widetilde{S}_h^{t,\ell} = S_h^{t,\ell}$. Here $\widetilde{z}_h^{t,\ell}$ is independent of $z_h^{t,\ell}$ and all random variables in $\mathcal{D}_{N,T}$ generated later than $S_h^{t,\ell}$. In the following, we use this ghost sample and Donsker-Varadhan representation to decompose the error.

Donsker-Varadhan representation (MacKay, 2003) states that, for any distribution $P, Q \in \mathcal{P}_{\mathrm{LLM}}$ and for any function $g : \mathcal{P}_{\mathrm{LLM}} \to \mathbb{R}$ such that $\mathbb{E}_{\rho\sim Q}[\exp\big(g(\rho)\big)] < \infty$, we have

$$\mathbb{E}_{\rho\sim P}[g(\rho)] \le \mathtt{KL}(P, Q) + \log \mathbb{E}_{\rho\sim Q}[\exp\big(g(\rho)\big)]. \tag{77}$$

To proceed, we choose $Q \in \mathcal{P}_{\mathrm{LLM}}$ independent of both the dataset $\mathcal{D}_{N,T}$ and ghost dataset $\widetilde{\mathcal{D}}_{N,T}$ and $P \in \mathcal{P}_{\mathrm{LLM}}$ to be potentially dependent on the dataset $\mathcal{D}_{N,T}$ but independent of the newly sampled $\{\widetilde{z}_h^{t,\ell}\}$ in the ghost dataset. To simplify the notation, we omit the subscripts and write the datasets as $\mathcal{D}$ and $\widetilde{\mathcal{D}}$ respectively. In the sequel, we use $\mathbb{E}_{\mathcal{D}}$ or $\mathbb{E}_{\widetilde{\mathcal{D}}}$ to denote the expectation with respect to the joint distribution of $\mathcal{D}$ and $\widetilde{\mathcal{D}}$, respectively. We set the function g as $g(\rho) = L(\rho, \mathcal{D}) - \log \mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho, \widetilde{\mathcal{D}})) \,|\, \mathcal{D}]$, where

$$L(\rho, \widetilde{\mathcal{D}}) = -\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H} \log \frac{\mathbb{P}(\widetilde{z}_h^{t,\ell} \,|\, S_h^{t,\ell})}{\mathbb{P}_\rho(\widetilde{z}_h^{t,\ell} \,|\, S_h^{t,\ell})}. \tag{78}$$

Moreover, for any $\ell \in [N]$, $t \in [T]$, and $h \in \{0, \dots, H\}$, we let $L^{\ell,t,h}(\rho, \widetilde{D})$ denote the partial sum of $L(\rho, \widetilde{D})$ with the last term being $-1/4 \cdot (\log \mathbb{P}(\widetilde{z}_h^{t,\ell} \mid S_h^{t,\ell}) - \log \mathbb{P}_\rho(\widetilde{z}_h^{t,\ell} \mid S_h^{t,\ell}))$. We note that $L(\rho, \widetilde{D})$ itself is a random variable where the randomness stems from both ρ and $\widetilde{D}$.

Exponentiating both sides of (77) and taking expectations with respect to $\mathcal{D}$ on both sides, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\Big[\exp\Big(\mathbb{E}_{\rho\sim P}\big[L(\rho,\mathcal{D}) - \log \mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))]\big] - \mathtt{KL}(P,Q)\Big)\Big] \\
&\quad \le \mathbb{E}_{\mathcal{D}}\mathbb{E}_{\rho\sim Q}[\exp\big(L(\rho,\mathcal{D}) - \log \mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}]\big)] \\
&\quad = \mathbb{E}_{\rho\sim Q}\mathbb{E}_{\mathcal{D}}[\exp\big(L(\rho,\mathcal{D}) - \log \mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))]\,|\,\mathcal{D}\big)] = \mathbb{E}_{\rho\sim Q}\mathbb{E}_{\mathcal{D}}\left[\frac{\exp(L(\rho,\mathcal{D}))}{\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}]}\right],
\end{aligned}$$

where in the first equality we exchange the order of expectations due to the independence between the dataset $\mathcal{D}$ and the prior Q. By the construction of $L(\rho, \widetilde{\mathcal{D}})$ in (78), we have

$$\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}] = \prod_{\ell=1}^{N}\prod_{t=1}^{T}\prod_{h=0}^{H}\mathbb{E}_{\widetilde{z}_h^{t,\ell}}\Big[\exp\Big(-1/4\cdot\big(\log\mathbb{P}(\widetilde{z}_h^{t,\ell}\mid S_h^{t,\ell}) - \log\mathbb{P}_\rho(\widetilde{z}_h^{t,\ell}\mid S_h^{t,\ell})\big)\Big)\Big].$$

Notice that this is a random variable that is measurable under $\mathcal{F}_{N,T,H}$. Moreover, conditioning on $\mathcal{F}_{N,T,H}$, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\big[\exp(L(\rho,\mathcal{D}))\,\big|\,\mathcal{F}_{N,T,H}\big] \\
&\quad = \exp\big(L^{N,T,H-1}(\rho,\mathcal{D})\big)\cdot\mathbb{E}_{z_H^{T,N}}\Big[\exp\Big(-1/4\cdot\big(\log\mathbb{P}(z_H^{T,N}\mid S_H^{T,N}) - \log\mathbb{P}_\rho(z_H^{T,N}\mid S_H^{T,N})\big)\Big)\Big].
\end{aligned}$$

Since $\widetilde{z}_H^{T,N}$ and $z_H^{T,N}$ have the same conditional distribution, we obtain that

$$\mathbb{E}_{\mathcal{D}}\left[\frac{\exp(L(\rho,\mathcal{D}))}{\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}]}\,\middle|\,\mathcal{F}_{N,T,H}\right] = \frac{\exp\big(L^{N,T,H-1}(\rho,\mathcal{D})\big)}{\mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp(L^{N,T,H-1}(\rho,\widetilde{\mathcal{D}})\,|\,\mathcal{D}]}.$$

Then, using the tower property, we similarly have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}}\left[\mathbb{E}_{\mathcal{D}}\left[\frac{\exp(L(\rho,\mathcal{D}))}{\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}]}\,\middle|\,\mathcal{F}_{N,T,H}\right]\,\middle|\,\mathcal{F}_{N,T,H-1}\right] \\
&\quad = \mathbb{E}_{\mathcal{D}}\left[\frac{\exp\big(L^{N,T,H-1}(\rho,\mathcal{D})\big)}{\mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp(L^{N,T,H-1}(\rho,\widetilde{\mathcal{D}})\,|\,\mathcal{D}]}\,\middle|\,\mathcal{F}_{N,T,H-1}\right] = \frac{\exp\big(L^{N,T,H-2}(\rho,\mathcal{D})\big)}{\mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp(L^{N,T,H-2}(\rho,\widetilde{\mathcal{D}})\,|\,\mathcal{D}]}. \quad (79)
\end{aligned}$$

Recursively apply conditional expectations to (79) with respect to the filtration $\{\mathcal{F}_{\ell,t,h}\}$, we obtain that

$$\mathbb{E}_{\mathcal{D}}\Big[\exp(L(\rho,\mathcal{D}))/\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))\,|\,\mathcal{D}]\Big] = 1.$$

Therefore, we have

$$\mathbb{E}_{\mathcal{D}}\Big[\exp\Big(\mathbb{E}_{\rho\sim P}\big[L(\rho,\mathcal{D}) - \log\mathbb{E}_{\widetilde{\mathcal{D}}}[\exp(L(\rho,\widetilde{\mathcal{D}}))]\big] - \mathtt{KL}(P,Q)\Big)\Big] \le 1.$$

Applying the Chernoff bound to it, we obtain a high probability bound as follows. With probability at least $1-\delta$, we have

$$-\mathbb{E}_{\rho\sim P}\Big[\log \mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp\big(L(\rho,\widetilde{\mathcal{D}})\big)\big]\Big] \le -\mathbb{E}_{\rho\sim P}\big[L(\rho,\mathcal{D})\big] + \mathtt{KL}(P,Q) + \log\frac{1}{\delta}. \tag{80}$$

Now we separately bound the left-hand side and the right-hand side of (80). Similar to the derivation in Zhang et al. (2023a), for the left-hand side of (80), using the definition of $L(\rho,\widetilde{\mathcal{D}})$ and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\log \mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp\big(L(\rho,\widetilde{\mathcal{D}})\big)\,|\,\mathcal{D}\big] \qquad (81)\\
&= \log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_\rho(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\\
&= \log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\Big(\log\frac{\mathbb{P}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}+\log\frac{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_\rho(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\Big)\bigg)\,\bigg|\,\mathcal{D}\bigg]\\
&\le \frac{1}{2}\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\\
&\qquad + \frac{1}{2}\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_\rho(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,|\,\mathcal{D}\bigg],
\end{aligned}$$

where the last inequality follows from Cauchy-Schwarz inequality. To see this, note that for two random variables X and Y, Cauchy-Schwarz inequality implies that

$$\mathbb{E}[\exp((X+Y)/2)] \le \sqrt{\mathbb{E}[\exp(X)]\cdot\mathbb{E}[\exp(Y)]}.$$

According to the definition of Hellinger distance, we have $1-H^2(Q_1,Q_2)=\int_x\sqrt{q_1(x)q_2(x)}\mathrm{d}x = \mathbb{E}_{x\sim Q_2}\sqrt{q_1(x)/q_2(x)}$. Therefore, we can rewrite the first term on the right-hand side of (81) using Hellinger distance as follows,

$$\begin{aligned}
&\frac{1}{2}\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\\
&\quad= \frac{1}{2}\log \mathbb{E}_{\widetilde{\mathcal{D}}}\left[\prod_{\ell=1}^{N}\prod_{t=1}^{T}\prod_{h=0}^{H}\sqrt{\frac{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}}\,\middle|\,\mathcal{D}\right],\\
&\quad= \frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\Big(1-\mathrm{H}^2(\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell}),\mathbb{P}(\cdot\,|\,S_h^{t,\ell}))\Big). \qquad (82)
\end{aligned}$$

Due to the fact that $\log(1-x)\le -x$ for $x\in[0,1)$, we further upper bound (82) as follows,

$$\begin{aligned}
\text{RHS of (82)} &\le -\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathrm{H}^2(\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell}),\mathbb{P}(\cdot\,|\,S_h^{t,\ell}))\\
&\le -\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathrm{TV}^2\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell}),\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell})\big), \qquad (83)
\end{aligned}$$

where the second line follows from the fact that $2\mathrm{H}^2(Q_1, Q_2) \geq \mathtt{TV}^2(Q_1, Q_2)$. Applying (83) to the left-hand side of (80), we thus have

$$-\mathbb{E}_{\rho\sim P}\Big[\log \mathbb{E}_{\widetilde{\mathcal{D}}}\big[\exp\big(L(\rho,\widetilde{\mathcal{D}})\big)\,|\,\mathcal{D}\big]\Big] \geq \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathrm{TV}^2\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell})\big) \tag{84}$$
$$- \frac{1}{2}\mathbb{E}_{\rho\sim P}\bigg[\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}_{\widehat{\rho}}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(\widetilde{z}_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\bigg].$$

Next, we upper bound the right-hand side of (80). For any $\rho' \in \mathcal{P}_{\mathrm{LLM}}$, we have

$$\begin{aligned}
&-\mathbb{E}_{\rho\sim P}\big[L(\rho,\mathcal{D})\big] + \mathtt{KL}(P,Q) + \log(\frac{1}{\delta}) \qquad (85)\\
&= \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell}\,|\,S_h^{t,\ell})} + \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{\rho\sim P}\bigg[\log\frac{\mathbb{P}_{\rho'}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg] + \mathtt{KL}(P,Q) + \log(\frac{1}{\delta})\\
&\leq \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell}\,|\,S_h^{t,\ell})} + \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{\rho\sim P}\bigg[\log\frac{\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg] + \mathtt{KL}(P,Q) + \log(\frac{1}{\delta}),
\end{aligned}$$

where the last inequality holds by noting that $\widehat{\rho}$ maximizes the likelihood function.

We next choose ρ' as the projection of $\mathbb{P}$ (in terms of the KL divergence) onto the space of all parameterized learnable models $\{\mathbb{P}_\rho \,|\, \rho \in \mathcal{P}_{\mathrm{LLM}}\}$, i.e.,

$$\rho' = \operatorname*{argmin}_{\rho^*\in\mathcal{P}_{\mathrm{LLM}}} \mathbb{E}_{S\sim\mathcal{D}}\mathtt{KL}\big(\mathbb{P}(\cdot|S)\|\mathbb{P}_{\rho^*}(\cdot|S)\big).$$

Combining inequalities (84) and (85), we thus upper bound the desired pretraining error as a sum of a few terms as follows

$$\begin{aligned}
&\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathrm{TV}^2\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell})\big)\\
&\quad\leq \underbrace{\frac{1}{2}\mathbb{E}_{\rho\sim P}\bigg[\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\bigg]}_{\text{(I.i)}} \qquad (86)\\
&\qquad+ \underbrace{\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{\rho\sim P}\bigg[\log\frac{\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg]}_{\text{(I.ii)}}\\
&\qquad+ \underbrace{\frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}}_{\text{(II)}} + \underbrace{\mathtt{KL}(P,Q)}_{\text{(III)}} + \log\frac{1}{\delta}.
\end{aligned}$$

Here the first two errors (I.i) and (I.ii) represent the fluctuation error due to the randomness of $\rho \sim P$, (II) is the approximation error that characterizes the discrepancy between the

true distribution $\mathbb{P}$ and its best approximator $\mathbb{P}_{\rho'}$, and (III) is the KL divergence between P and Q. Note that the left-hand side of (86) can be written as

$$NT \cdot (H+1)/4 \cdot \mathbb{E}_{S\sim\mathcal{D}}\big[\mathtt{TV}^2\big(\mathbb{P}(\cdot\,|\,S), \mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S)\big)\big].$$

With the error decomposition in (86), we conclude **Step 1**. In the following, we will specify distributions P and Q.

Step 2: Control each term in the decomposition of pretraining error. In this step, we control each term in the error decomposition (86).

Our first step is to control the fluctuation errors (I.i) and (I.ii), which describe the log density ratio between $\mathbb{P}_{\widehat{\rho}}$ and $\mathbb{P}_{\rho}$. The errors (I.i) and (I.ii) are small when ρ is close to $\widehat{\rho}$. Therefore we control these two terms by setting the support of P to be a neighborhood around $\widehat{\rho}$. Specifically, for each weight matrix and residual link specified by

$$\widehat{\rho} = \Big(W_{\mathtt{softmax}}, \{W^d_{\mathrm{ff},1}, W^d_{\mathrm{ff},2}, \{W_i^{Q,d}, W_i^{K,d}, W_i^{V,d}\}_{i=1}^{\eta}, \gamma_1^d, \gamma_2^d\}_{d=1}^{D}\Big),$$

we construct a ball with a radius shrinking at rate $1/(NT(H+1))$. More specifically, we define

$$P = \mathcal{B}_S \cdot \prod_{d=1}^{D} \mathcal{B}_M(d) \cdot \mathcal{B}_F(d) \cdot \mathcal{B}_R(d), \tag{87}$$

where we define the balls around each weight matrix in each layer d as

$$\begin{aligned}
\mathcal{B}_S &= \mathtt{Unif}\big(B(W_{\mathrm{softmax}}, r_S, \|\cdot\|_{1,2})\big),\\
\mathcal{B}_R(d) &= \mathtt{Unif}\big(B(\gamma_1^d, r^{(d)}_{\gamma,1}, |\cdot|)\big)\cdot \mathtt{Unif}\big(B(\gamma_2^d, r^{(d)}_{\gamma,2}, |\cdot|)\big),\\
\mathcal{B}_F(d) &= \mathtt{Unif}\big(B(W^d_{\mathrm{ff},1}, r^{(d)}_{F,1}, \|\cdot\|_{\mathrm{F}})\big)\cdot \mathtt{Unif}\big(B(W^d_{\mathrm{ff},2}, r^{(d)}_{F,2}, \|\cdot\|_{\mathrm{F}})\big),\\
\mathcal{B}_M(d) &= \prod_{i=1}^{\eta} \mathtt{Unif}\big(B(W_i^{Q,d}, r_V^{(d)}, \|\cdot\|_{\mathrm{F}})\big)\cdot \mathtt{Unif}\big(B(W_i^{Q,d}, r_Q^{(d)}, \|\cdot\|_{\mathrm{F}})\big)\cdot \mathtt{Unif}\big(B(W_i^{Q,d}, r_K^{(d)}, \|\cdot\|_{\mathrm{F}})\big).
\end{aligned}$$

The ball around center x with radius r is defined as $B(x, r, \|\cdot\|) = \{y\,|\,\|x-y\| \le r\}$. And $\mathtt{Unif}(\mathcal{S})$ denotes uniform distribution over the set $\mathcal{S}$. Finally, we specify the radius as follows,

$$\begin{aligned}
r_K^{(d)} = r_Q^{(d)} &= R^{-1}\eta^{-1}(1+B_F^2)^{-1}B_M^{-2}\alpha_d^{-1}/(NT(H+1)),\\
r_{F,1}^{(d)} = r_{F,2}^{(d)} &= R^{-1}B_F^{-1}\alpha_d^{-1}/(NT(H+1)),\\
r_V^{(d)} &= R^{-1}\eta^{-1}(1+B_F^2)^{-1}\alpha_d^{-1}/(NT(H+1)),\\
r_{\gamma,1}^{(d)} &= R^{-1}(1+B_F^2)^{-1}\alpha_d^{-1}/(NT(H+1)),\\
r_{\gamma,2}^{(d)} &= R^{-1}\alpha_d^{-1}/(NT(H+1)),\\
r_S &= \tau B_s^{-1}/(NT(H+1)),\\
\text{where } \alpha_d &= \frac{2}{\tau}B_S(1+B_F^2)\big(1+\eta B_M(1+4B_M^2)\big)^{D-d}.
\end{aligned}$$

Under this assignment of P, we can control (I.i) and (I.ii) by invoking the following lemma from Zhang et al. (2023a).

Lemma 33 *We set the distribution P to be (87), which is a uniform distribution over a neighborhood around $\widehat{\rho}$ with radius proportional to $1/NT(H+1)$. Under Assumptions 12 and 13, we have*

$$\frac{1}{2}\mathbb{E}_{\rho\sim P}\bigg[\log \mathbb{E}_{\widetilde{\mathcal{D}}}\bigg[\exp\bigg(-\frac{1}{2}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\log\frac{\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg)\,\bigg|\,\mathcal{D}\bigg]\bigg] + \frac{1}{4}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{\rho\sim P}\bigg[\log\frac{\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}\bigg] = \mathcal{O}(1).$$

Proof See Appendix F.2 by Zhang et al. (2023a) for a detailed proof. ■

This lemma quantifies how $\mathbb{P}_\rho$ changes when ρ is getting closer to $\widehat{\rho}$. In the following, we briefly outline the proof and refer readers to the original work by Zhang et al. (2023a) for more details. The proof consists of two steps. The first step is to control the TV distance between $\mathbb{P}_\rho$ and $\mathbb{P}_{\widehat{\rho}}$ using the differences between the layer parameters specified in ρ and $\widehat{\rho}$. The second step sets $\rho\sim P$, where the distribution P in (87) is supported on a neighborhood around $\widehat{\rho}$. Then for any $\rho\in\texttt{supp}(P)$, we can control the log density ratio between $\mathbb{P}_{\widehat{\rho}}$ and $\mathbb{P}_\rho$ using the radius defined in (87) as follows:

$$\log\big(\mathbb{P}_{\widehat{\rho}}(z_h^{t,\ell}\mid S_h^{t,\ell})/\mathbb{P}_{\rho}(z_h^{t,\ell}\mid S_h^{t,\ell})\big) = \mathcal{O}\big(1/(NT(H+1))\big) \tag{88}$$

for any $(z_h^{t,\ell}, S_h^{t,\ell})$. Therefore, we conclude that the fluctuation error (I) has a rate of $\mathcal{O}(1)$.

Next, we control error (II) using the following lemma.

Lemma 34 *Under Assumptions 12 and 13, with probability at least $1-\delta$, we have*

$$\frac{1}{NT(H+1)}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\bigg(\log\frac{\mathbb{P}(z_h^{t,\ell}\,|\,S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell}\,|\,S_h^{t,\ell})} - \mathbb{E}_{S_h^{t,\ell}}\texttt{KL}\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell})\,|\,\mathbb{P}_{\rho'}(\cdot\,|\,S_h^{t,\ell})\big)\bigg) \le b^*\sqrt{\frac{1}{2N}\log\frac{T(H+1)}{\delta}}. \tag{89}$$

Proof See Appendix I.3 for details. ■

This lemma controls error (II). A key part of the proof is to derive the log-density bound

$$\big|\log\mathbb{P}(z\,|\,S) - \log\mathbb{P}_{\widehat{\rho}}(z\,|\,S)\big| \le b^* = \log\max\{c_0^{-1}, 1+|\mathcal{L}|\exp(B_S/\tau)\}, \tag{90}$$

which provides the explicit form of b^* mentioned earlier in Assumption 3. The proof involves applying Hoeffding's inequality, along with the log-density bound in (90), to the left-hand side of (89).

Furthermore, we control (III), the KL divergence between P and Q, using the following lemma obtained from Zhang et al. (2023a). To make sure that $\texttt{supp}(P)\subseteq\texttt{supp}(Q)$, we set Q to be uniformly distributed over $\mathcal{P}_{\text{LLM}}$. More specifically, we have

$$Q = \mathcal{B}_S'\cdot\prod_{d=1}^{D}\mathcal{B}_M'(d)\cdot\mathcal{B}_F'(d)\cdot\mathcal{B}_R'(d), \tag{91}$$

where we define the balls around each weight matrix in each layer d as

$$\mathcal{B}'_F(d) = \prod_{i=1}^{2} \texttt{Unif}\big(B(0, B_F, \|\cdot\|_{\mathrm{F}})\big), \quad \mathcal{B}'_M(d) = \prod_{i=1}^{3\eta} \texttt{Unif}\big(0, B_M, \|\cdot\|_{\mathrm{F}})\big),$$
$$\mathcal{B}'_S = \texttt{Unif}\big(B(0, B_S, \|\cdot\|_{1,2})\big), \qquad \mathcal{B}'_R(d) = \prod_{i=1}^{2} \texttt{Unif}\big(B(1/2, 1/2, |\cdot|)\big).$$

Lemma 35 *Let $\bar{D} = D^2 \cdot r \cdot (d_F + d_k + r) + r \cdot |\mathcal{L}|$ and $\bar{B} = \tau^{-1} R h B_S B_F^2 B_M^3$. Let distributions P and Q be defined as in* (87) *and* (91)*, respectively. We have that under Assumptions 12 and 13,*

$$\texttt{KL}(P, Q) = \mathcal{O}\big(\bar{D} \log(1 + NTH\bar{B})\big).$$

Proof See Equation (F.9) in Appendix F.2 of Zhang et al. (2023a) for a detailed proof. ■

This Lemma is proved by directly computing the KL divergence between two uniform distributions P and Q, where $\texttt{supp}(P) \subseteq \texttt{supp}(Q)$. The calculation can be found in Appendix F.2 of Zhang et al. (2023a).

Applying Lemmas 33, 34, and 35 to the three errors in (86), we have that with probability at least $1 - \delta$,

$$\begin{aligned}
&\frac{1}{NT(H+1)} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \texttt{TV}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot \mid S_h^{t,\ell})\big) \qquad (92)\\
&\le \left(\frac{1}{NT(H+1)} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \texttt{TV}^2\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot \mid S_h^{t,\ell})\big)\right)^{1/2}\\
&\le \mathcal{O}\Bigg(\underbrace{\frac{\sqrt{b^*}}{N^{1/4}} \log\frac{TH}{\delta} + \inf_{\rho' \in \mathcal{P}_{\mathrm{LLM}}} \sqrt{\frac{1}{NTH} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \mathbb{E}_{S_h^{t,\ell}} \texttt{KL}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}) \,|\, \mathbb{P}_{\rho'}(\cdot \mid S_h^{t,\ell})\big)}}_{\text{(II)}}\\
&\qquad + \underbrace{\sqrt{1/(NTH)}}_{\text{(I)}} + \underbrace{\sqrt{\frac{\bar{D}}{NTH}} \log(1 + NTH\bar{B})}_{\text{(III)}} + \sqrt{1/(NTH)} \log(1/\delta)\Bigg),\\
&\le \mathcal{O}\Bigg(\frac{\sqrt{b^*}}{N^{1/4}} \log\frac{TH}{\delta} + \inf_{\rho' \in \mathcal{P}_{\mathrm{LLM}}} \sqrt{\frac{1}{NTH} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \mathbb{E}_{S_h^{t,\ell}} \texttt{KL}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}) \,|\, \mathbb{P}_{\rho'}(\cdot \mid S_h^{t,\ell})\big)}\\
&\qquad + \sqrt{\frac{\bar{D}}{NTH}} \log(1 + NTH\bar{B})\Bigg),
\end{aligned}$$

where the first line follows from Cauchy-Schwarz inequality, the second line follows from upper bounding the three errors in (86) using Lemmas 33, 34, and 35. The last line drops two terms that are dominated by the rest.

In the final step, we will change the left-hand side of (92) to its expectation. We control the difference between (92) and its expectation using the following lemma.

Lemma 36 *Under Assumptions 12 and 13, with probability at least $1-\delta$, we have*

$$\frac{1}{NT(H+1)}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\left(\mathbb{E}_{S_h^{t,\ell}}\left[\mathtt{TV}\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell}),\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell})\big)\right]-\mathtt{TV}\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell}),\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S_h^{t,\ell})\big)\right)$$
$$=\mathcal{O}\bigg(\frac{1}{\sqrt{N}}\Big(\bar{D}\log(1+NTH\bar{B})+\log\frac{TH}{\delta}\Big)\bigg).$$

Proof See Appendix I.3 for details. ■

This lemma follows from establishing uniform convergence between the TV distances and their expectations that hold for any distribution P. Adding Lemma 36 to (92), we obtain the rate for pretraining error:

$$\begin{aligned}
&\mathbb{E}_{S\sim\mathcal{D}}\big[\mathtt{TV}\big(\mathbb{P}(\cdot\,|\,S),\mathbb{P}_{\widehat{\rho}}(\cdot\,|\,S)\big)\big]\\
&\quad=\mathcal{O}\bigg(\frac{\sqrt{b^*}}{N^{1/4}}\log\frac{TH}{\delta}+\inf_{\rho'\in\mathcal{P}_{\mathrm{LLM}}}\sqrt{\frac{1}{NTH}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{S_h^{t,\ell}}\mathtt{KL}\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell})\,|\,\mathbb{P}_{\rho'}(\cdot\,|\,S_h^{t,\ell})\big)}\\
&\qquad+\frac{1}{\sqrt{N}}\Big(\bar{D}\log(1+NTH\bar{B})+\log\frac{TH}{\delta}\Big)+\sqrt{\frac{\bar{D}}{NTH}}\log(1+NTH\bar{B})\bigg)\\
&\quad=\mathcal{O}\bigg(\frac{\sqrt{b^*}}{N^{1/4}}\log\frac{TH}{\delta}+\inf_{\rho'\in\mathcal{P}_{\mathrm{LLM}}}\sqrt{\frac{1}{NTH}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\mathbb{E}_{S_h^{t,\ell}}\mathtt{KL}\big(\mathbb{P}(\cdot\,|\,S_h^{t,\ell})\,|\,\mathbb{P}_{\rho'}(\cdot\,|\,S_h^{t,\ell})\big)}\\
&\qquad+\frac{1}{\sqrt{N}}\Big(\bar{D}\log(1+NTH\bar{B})\Big),
\end{aligned}$$

where the final line follows from dropping the last term in the second line, which is dominated by the rest. Therefore, we conclude the proof.

■

H.2 Formal Statement of Proposition 15

In this section, we formally state Proposition 15 and provide its proof. For simplicity, we derive the approximation error bound for reasoning steps of dimension one, i.e., we regard $\mathcal{L}$ as a subset of $\mathbb{R}$. Our method can be readily generalized to higher-dimensional cases (Elbrächter et al., 2021). In this proof, we construct networks with specific parameters such that the KL divergence between the target distribution $\mathbb{P}$ and its best transformer neural network approximation $\mathbb{P}_{\widehat{\rho}}$ decays exponentially as the network depth increases.

We let T be the maximal number of examples included in the prompt. Thus, $L = T(H+1)$ is the largest number of reasoning steps included in the prompt. We let $S_h^t = \{\Upsilon_{t-1}, z_{0:(h-1)}^t\}$ denote the collection of $t-1$ examples of reasoning paths and a partial trajectory of length h of the t-th example. Here $h \in \{0,\ldots,H\}$. Note that the desired transformer neural network takes each S_h^t as the input and outputs an element in the probability distribution over $\mathcal{L}$ as the conditional distribution of z_h^t. That is, the transformer takes a sequence of reasoning steps as the input and outputs a probability distribution.

Since each position of the input is indexed by (t, h), to simplify the notation, we use $L'(t,h) = (t-1)(H+1)+h$ to denote the length of S_h^t. For any $t, t' \in [T]$ and $h, h' \in \{0, \ldots, H\}$, we write

$$(t', h') < (t, h) \qquad \text{if and only if} \qquad L'(t', h') = (t'-1)(H+1)+h' < L'(t, h). \tag{93}$$

That is, $(t', h') < (t, h)$ if and only if $z_{h'}^{t'}$ appears earlier than z_h^t.

In the sequel, we fix $t \in [T]$ and $0 \le h \le H$ and focus on the problem of approximating the conditional distribution of z_h^t. We abbreviate $L'(t,h)$ as L' when the meaning is clear from the context. The target distribution is denoted by a function $g_h^* : \mathcal{L}^{L'} \to \mathbb{R}^{|\mathcal{L}|}$, i.e., $g_h^*(S_h^t) = \mathbb{P}(z_h^t = \cdot \,|\, S_h^t)$. That is, for any $z \in \mathcal{L}$, the z-th entry of $g_h^*(S_h^t)$ is equal to $\mathbb{P}(z_h^t = z \,|\, S_h^t)$. Functions $\{g_h^*\}_{h=0}^H$ are the target functions and we want to **construct a single transformer that approximates all of them**.

Function class containing $\{g_h^*\}_{h=0}^H$. Under the general model introduced in Appendix C, by the Bayes' rule, function $g_h^*(\cdot)$ is invariant to permutations of the $L(t,h)$ reasoning steps in S_h^t. Theorem 2 in Zaheer et al. (2017) proves that any permutation invariant function of a function admits a factorization structure. In particular, there exists $w_h^* : \mathbb{R} \to \mathbb{R}^{|\mathcal{L}|}$ and $\psi_h^* : \mathcal{L} \to \mathbb{R}$ for all $h \in \{0, \ldots, H\}$ and $t \in [T]$ such that

$$g_h^*(S_h^t) = w_h^*\bigg(\frac{1}{L'}\bigg(\sum_{i=1}^{t-1}\sum_{j=0}^{H}\psi_h^*(z_j^i) + \sum_{j'=0}^{h-1}\psi_h^*(z_{j'}^t)\bigg)\bigg). \tag{94}$$

In particular, if $h = 0$, the second summation $\sum_{j'=0}^{h-1}\psi_h^*(z_{j'}^t)$ is set to zero. Let $w_{h,i}^*$ denote the i-th component of w_h^* for all $i \in [|\mathcal{L}|]$.

In the following, we let $\mathcal{S}^\infty([-B,B],\mathbb{R})$ denote the set of real-valued smooth functions on $[-B,B]$ equipped with the ℓ_∞-norm $\|f\|_\infty = \sup_{x\in[-B,B]}|f(x)|$. We define $\mathcal{S}_B$ as the set of smooth functions with bounded derivatives:

$$\mathcal{S}_B = \Big\{ f \in \mathcal{S}^\infty([-B,B],\mathbb{R}) \,|\, \big\|f^{(n)}\big\|_\infty \le C_{\mathcal{S}} \cdot n! \text{ for all } n \in \mathbb{N}_+, \text{ and } \|f\|_\infty \le C_A \Big\},$$

where $f^{(n)}$ is the n-th order derivative of f, $C_{\mathcal{S}}$ is a constant, and $\mathbb{N}_+$ is the set of positive integers. Here $\mathcal{S}_B$ contains functions whose high-order derivatives grow moderately fast in magnitude. We impose some regularity assumptions on functions $\{g_h^*\}_{h=0}^H$ as follows.

Assumption 37 *We assume that there exists $B, C_A > 0$ such that for any $h \in \{0, \ldots, H\}$, we have $\psi_h^*, \tau \log w_{h,i}^* \in \mathcal{S}_B$ for $i \in [|\mathcal{L}|]$, where τ is the temperature of the LLMs and $w_{h,i}^*$ is the i-th entry of w_h^* in (94). Moreover, without loss of generality, we assume $C_{\mathcal{S}} = 1$.*

This assumption states that the target functions $\{g_h^*\}_{h=0}^H$ are sufficiently smooth in the sense that all functions appearing in the factorization in (94) are smooth. We establish the approximation error in the following proposition.

Proposition 38 (Formal Statement of Proposition 15) *Let $S_h^t = (\Upsilon_{t-1}, \{z_j^t\}_{j=0}^{h-1})$ be the sequence of reasoning steps that includes $t-1$ examples of reasoning paths Υ_{t-1} and the first $h-1$ steps of the t-th example $\{z_j^t\}_{j=0}^{h-1}$. Let D denote a sufficiently large integer,*

consider the parameter class $\mathcal{P}_{\mathrm{LLM}}$ *in* (10) *with* $d_F \geq 18|\mathcal{L}| + 4$, $B_S \geq (C_A + 1) \cdot \sqrt{|\mathcal{L}|}$, $B_M \geq \max\{8\log(8TH), \sqrt{B^2 + (H+1)^2 + 1}\}$, *and*

$$B_F \geq C_F \cdot \sqrt{B^2 + H^2 + C_A^2 \cdot |\mathcal{L}|} \cdot 16^{D'} \cdot |\mathcal{L}|^{3/2},$$

where $D' = (D - C_p \log(3H))/(H+1)$, $C_F, C_p > 0$ *are absolute constants, and* C_A *is from Assumption 37. Under Assumptions 12, 13 and 37, there exists a transformer with at most* $\mathcal{O}(D)$ *transformer blocks and parameter* $\rho^* \in \mathcal{P}_{\mathrm{LLM}}$ *satisfying*

$$\max_{S_h^t \in \mathcal{L}^*} \mathtt{KL}\big(\mathbb{P}(z_h^t = \cdot \mid S_h^t), \mathbb{P}_{\rho^*}(z_h^t = \cdot \mid S_h^t)\big) = \mathcal{O}\bigg(\exp\bigg(-\frac{\big(D - C\log(2H))/H\big)^{1/4}}{5B}\bigg)\bigg),$$

for all $t \in [T]$ *and* $h \in \{0, \ldots, H\}$, *where* $C > 0$ *is an absolute constant. The integer* H *is the length of a reasoning trajectory,* B *is the parameter from Assumption 37, and* $|\mathcal{L}|$ *is the alphabet size of the output distribution. We note that* $\mathcal{O}(\cdot)$ *is with respect to the asymptotic regime where* D *goes to infinity.*

This proposition shows that the approximation error decays exponentially to zero as D increases. The proof is based on an explicit construction of a transformer neural network that estimates $\{g_h^*\}_{h=0}^H$ altogether. The transformer architecture follows the one described in Appendix I.1. In particular, the transformer has $H+1$ submodules that approximate each g_h^* separately. Besides, we assume $C_{\mathcal{S}} = 1$ in Assumption 37 only to simplify the presentation. Our approximation result can be modified for a general $C_{\mathcal{S}}$ by changing the constants in the upper bound correspondingly.

The proof of this proposition is technical and lengthy. We present a detailed proof in Appendix H.3 and give an overview as follows.

Overview of the Proof of Proposition 38. Note that the transformer takes S_h^t as the input and outputs a probability distribution over $\mathcal{L}$, where $h \in \{0, \ldots, H\}$ and $t \in [T]$. We fix some arbitrary (t, h) and consider the problem of predicting z_h^t.

As introduced in Appendix I.1, in the transformer architecture, the input sequence is first embedded in a Euclidean space and then passed through a series of transformer blocks. Then the output goes through a softmax output layer to generate a probability distribution. Intuitively, when predicting z_h^t using S_h^t, we need to first extract the step-index h and then apply an approximation of g_h^*. To achieve this goal, our transformer includes an extraction module $\mathtt{NN}_{\mathtt{J}}$ followed by $H+1$ approximation and selection modules $\{G_{h'}, F_{h'}\}_{h'=0}^H$. Here $\mathtt{NN}_{\mathtt{J}}$ adds the desired step-index, i.e., h, to all the $L'(t,h)$ locations. The approximation module $G_{h'}$ approximates the target distribution $g_{h'}^*$ for all $h' \in \{0, \ldots, H\}$. Selection modules $\{F_{h'}\}_{h'=0}^H$ are used to select the particular approximation module G_h with step-index h. The output of the final selection module F_H is then passed to a softmax layer, which produces the output distribution. We list the components of the transformer architecture as follows. Also see Figure 5 for an illustration.

- **Input embedding**: Given a prompt $S_h^t \in \mathcal{L}^*$, we construct an *input embedding to* prepare for further processing of the input, which is defined as $X_{\mathrm{NN}}^{(0)}$ in (100). Here $X_{\mathrm{NN}}^{(0)}$ is a sequence of vectors of length $L'(t,h)$, where each vector has length 3, including the value of the reasoning step and its step-index.

- **Extraction module** $\texttt{NN}_{\texttt{J}}$: The extraction module $\texttt{NN}_{\texttt{J}}$ extracts the step-index h from the last reasoning step z_{h-1}^t and copies it to all previous reasoning steps $\{z_{h'}^{t'}\}_{(t',h')<(t,h-1)}$. Specifically, this module takes input embedding $X_{\mathrm{NN}}^{(0)} \in \mathbb{R}^{L'\times 3}$ in (100) and outputs a vector sequence of length $L'(t,h)$, where each vector is in $\mathbb{R}^{2+2|\mathcal{L}|}$. In each vector indexed by (t',h'), the first entry is the value of the reasoning step $z_{h'}^{t'}$, and the second entry is approximately equal to h, the step-index of the desired output z_h^t. The remaining $2|\mathcal{L}|$ entries of each vector are all set to zero. The output of $\texttt{NN}_{\texttt{J}}$ is then fed into a sequence of $H+1$ approximation and selection modules.

- **Approximation module** $G_{h'}$: For any $h' \in \{0,\dots,H\}$, $G_{h'}$ computes an embedding of S_h^t, denoted by $\texttt{embed}_{h'}(S_h^t)$, which is a vector-valued function in $|\mathcal{L}|$. In particular, $\texttt{embed}_{h'}(S_h^t)$ is used to approximate the target distributions $g_{h'}^*(S_h^t)$ after a softmax transformation. Each $G_{h'}$ is a mapping that maps a sequence of $L' = L'(t,h)$ vectors in $\mathbb{R}^{2+2|\mathcal{L}|}$ to a vector sequence of the same shape, i.e., a function between $\mathbb{R}^{L'\times(2+2|\mathcal{L}|)}$ to itself. Here $G_{h'}$ only changes the columns with indices in $\{3,\dots,2+2|\mathcal{L}|\}$ and sets them to $\texttt{embed}_{h'}(S_h^t) \in \mathbb{R}^{L'\times|\mathcal{L}|}$, where $\texttt{embed}_{h'}$ is a matrix-valued mapping that maps $S_h^t \in \mathbb{R}^{L'\times 1}$ to a matrix in $\mathbb{R}^{L'\times|\mathcal{L}|}$.

- **Selection module** $F_{h'}$: For any $h' \in \{0,\dots,H\}$, $F_{h'}$ checks if its index h' matches the extracted index h. When viewing each $F_{h'}$ as a matrix-valued mapping from $\mathbb{R}^{L'\times(2+2|\mathcal{L}|)}$ to itself, it only changes the last $|\mathcal{L}|$ columns of the matrix and uses them as a "memory". In particular, it approximately adds $\texttt{embed}_{h'}(S_h^t)\cdot\mathbb{1}\{h'=h\}$ to the memory and passes it to the subsequent modules. As a result, after the last selection module, F_H, the last $|\mathcal{L}|$ columns of the output matrix are given by $\sum_{h'=0}^{H}\texttt{embed}_{h'}(S_h^t)\cdot\mathbb{1}\{h'=h\} \approx \texttt{embed}_h(S_h^t)$. Thus, by combining the approximation and selection modules, we eventually obtain $\texttt{embed}_h(S_h^t)$ approximately.

- **Output softmax layer**: Finally, we pass the output of F_H, $\texttt{embed}_h(S_h^t)$, to a $\texttt{softmax}$ function to produce the output distribution $\widehat{g}_h^*(S_h^t)$, which is closed to the desired output $g_h^*(S_h^t)$.

H.3 Proof of Proposition 38

Before the formal proof, we would like to highlight that our construction in the proof is based on a slightly generalized version of the transformer structure in Section 6.1. We note that this slight generalization can be easily taken into account in the generalization error in Proposition 14. Here, we first define a single **transformer block** as follows, which takes $X \in \mathbb{R}^{L\times r}$ as input and output $Y \in \mathbb{R}^{L\times r}$.

$$\begin{aligned} Z &= \texttt{NL}\big(\texttt{mha}(X, W_{\mathrm{mha}}) + X\gamma_1\big), \\ Y &= \texttt{NL}\big(\texttt{ff}(Z, W_{\mathrm{ff}}, b_{\mathrm{ff}}) + Z\gamma_2\big), \end{aligned} \tag{95}$$

where γ_1 and γ_2 are diagonal matrices, and the fully connected feed-forward (FF) network $\texttt{ff}$ is defined as

$$\texttt{ff}(X_{\mathrm{in}}, W_{\mathrm{ff}}, b_{\mathrm{ff}}) = \texttt{ReLU}(X_{\mathrm{in}}W_{\mathrm{ff},1} + \mathbf{1}^\top b_{\mathrm{ff},1})W_{\mathrm{ff},2} + \mathbf{1}^\top b_{\mathrm{ff},2}. \tag{96}$$

Compared to the FF layer in Section 6.1, this FF layer has two additional bias terms $b_{\mathrm{ff},1} \in \mathbb{R}^{d_F}$ and $b_{\mathrm{ff},1} \in \mathbb{R}^r$. Here, $\mathtt{NL}$ is the row-wise ℓ_2-normalization layer, which is defined in (140) in Appendix I.1. This function projects each row of the input matrix into the unit ℓ_2-ball. Moreover, the multi-head attention (MHA) layer $\mathtt{mha}$ is defined in (18). In particular, a transformer block can be viewed as a four-layer neural network, where both $\mathtt{ff}$ and $\mathtt{mha}$ have two neural network layers. In this proof, we use "module" to refer to a sequence of transformer blocks that achieves certain functionality.

Throughout this proof, we construct a transformer neural network that includes an input embedding module, a sequence of transformer blocks, and the output softmax layer. Instead of counting the number of neural network layers in the transformer, we keep track of the number of transformer blocks.

In this proof, we often construct neural network components that are solely based on the MHA or FF layers. These layers themselves can be regarded as special cases of the transformer block, as shown below.

Multi-Head Attention Layer as a Transformer Block. Let W_{mha} be the weight matrices of a MHA layer. To view $\mathtt{mha}(\cdot, W_{\mathrm{mha}})$ as a single transformer block, we can set W_{ff} and b_{ff} as zero matrices and vectors respectively. Then $\mathtt{ff}$ in (96) becomes a zero function. We also set $\gamma_1 = \mathbf{0}$ and $\gamma_2 = I$ in (95).

It remains to consider the normalization layer, which plays a role when row-wise ℓ_2-norm of $\mathtt{mha}(X, W_{\mathrm{mha}})$ exceeds one. To handle this, we introduce a scaling trick as follows. When the input matrix X has bounded rows and W_{mha} is bounded, we know that each $XW_i^V \in \mathbb{R}^{L \times d_v}$ has bounded rows. We let $B \geq 1$ be an upper bound of the ℓ_2-norm of the rows of $\mathtt{mha}(X, W_{\mathrm{mha}})$ for all bounded input matrix X. Then we define another set of MHA parameters $\overline{W}_{\mathrm{mha}}$ as $\{W_i^Q, W_i^K, \overline{W}_i^V\}_{i=1}^{\eta}$, where $\overline{W}_i^V = W_i^V/B$. Thus, for any input matrix X, we have $\mathtt{mha}(X, \overline{W}_{\mathrm{mha}}) = \mathtt{mha}(X, W_{\mathrm{mha}})/B$, whose row-wise ℓ_2-norm is no more than one. Therefore, for any input matrix X and any weight matrix W of a proper size, we have

$$\mathtt{NL}\big(\mathtt{mha}(X, \overline{W}_{\mathrm{mha}})\big) \cdot \overline{W} = \mathtt{mha}(X, W_{\mathrm{mha}})W, \tag{97}$$

where we set $\overline{W} = B \cdot W$. Here W is some weight matrix that is multiplied to the output of MHA layer, i.e., a weight matrix of the next layer. The equality in (97) shows that suppose our constructed neural network involves a softmax layer, we can scale the weight matrices to ensure that it is equivalent to a transformer block. Moreover, the norms of these matrices are scaled by a factor of B.

Fully Connected Layer as a Transformer Block. Similarly, consider a FF layer with parameters $\{W_{\mathrm{ff}}, b_{\mathrm{ff}}\}$. We set $\{W_i^V\}_{i=1}^{\eta}$ to be a zero matrix and thus $\mathtt{mha}(\cdot, W_{\mathrm{mha}})$ becomes a zero function. We then set $\gamma_1 = I$ and $\gamma_2 = \mathbf{0}$ in (95). Similarly, we can apply the scaling trick by multiplying $W_{\mathrm{ff},2}$ and $b_{\mathrm{ff},2}$ by $1/B$ for some parameter B. This ensures that the output of the FF layer in (96) has row-wise ℓ_2-norm bounded by one, and thus the normalization $\mathtt{NN}(\cdot)$ does not take effect. We can multiply the weight matrix in the subsequent layer by B and get the desired output.

Multi-Layer Perceptron as Transformer Blocks. The above argument can be extended to multi-layer perceptrons (MLPs), i.e., a multi-layer feed-forward neural network. We can show that an MLP can be written as a composition of multiple transformer blocks.

This is achieved by (i) setting $W_i^V = \mathbf{0}$ in the MHA layer and setting $\gamma_1 = I$ and $\gamma_2 = \mathbf{0}$ and (ii) applying the scaling trick in each transformer block.

Specifically, we define an MLP as a composition of L feed-forward layers with parameters $\{W_{\mathrm{ff}}^\ell, b_{\mathrm{ff}}^d\}_{\ell\in[L]}$. Given the input matrix $X^0 \in \mathbb{R}^{L\times r}$, the output of each layer is given by

$$X^\ell = \mathtt{ReLU}(X^{\ell-1}W_{\mathrm{ff}}^\ell + \mathbf{1}^\top b_{\mathrm{ff}}^\ell), \qquad \forall \ell \in [L] \tag{98}$$

Here W_{ff}^ℓ and b_{ff}^ℓ are the weight matrix and bias vector of a proper dimension. We have the following result showing that a L-layer MLP can be represented as a transformer with L blocks.

Proposition 39 *We consider a row-wise fully-connected network defined in* (98)*. Let $X^0 \in \mathbb{R}^{L\times r}$ denote the input of this network, and the intermediate outputs are given by* (98)*. We assume there exist positive numbers $\{B_\ell, 0 \le \ell \le L\}$ such that $\|(X^\ell)^\top\|_{2,\infty} \le B_\ell$ for all $\ell \in \{0,\dots,L\}$ with $B_\ell \ge 1$. Consider a transformer with input $Y^0 = X^0$. Let Y^ℓ denote the output of the ℓ-th transformer block for all $\ell \ge 1$. Then, we can construct a transformer with L transformer blocks such that $Y^\ell = X^\ell/B_\ell$ for all $\ell \in [L]$. Moreover, let $\{\overline{W}_{\mathrm{ff}}^\ell, \overline{b}_{\mathrm{ff}}^\ell\}$ denote the parameters of FF layer of the ℓ-th transformer block. We have*

$$\overline{W}_{\mathrm{ff},1}^\ell = B_{\ell-1}\cdot W_{\mathrm{ff}}^\ell, \quad W_{\mathrm{ff},2}^\ell = I/B_\ell, \quad \overline{b}_{\mathrm{ff},1}^\ell = b_{\mathrm{ff}}^\ell, \quad \overline{b}_{\mathrm{ff},2}^\ell = \mathbf{0}. \tag{99}$$

Suppose the weight matrices of a fully connected network with L layers have a maximum width d and maximum weight α, and the biases have maximum weight β. In that case, the magnitude of the intermediate output can increase at most exponentially with $d\cdot\alpha$. More specifically, by direct calculation, we have

$$B_\ell \le \sqrt{d}\cdot\Big(B_0\cdot(d\cdot\alpha)^\ell + \beta\cdot((d\cdot\alpha)^\ell - 1)/(d\cdot\alpha - 1)\Big) \text{ for } \ell \in [L].$$

Proof See Appendix I.4.1 for a detailed proof. Here $W_{\mathrm{ff},2}^\ell$ in (99) is proportional to an identity matrix of a proper dimension, and $b_{\mathrm{ff},2}^\ell$ is a zero vector. The details of the other parameters of the transformer can be found in the proof. ■

H.3.1 Rigorous Proof of Proposition 38

Proof Throughout this proof, we focus on the problem of approximating $\mathbb{P}(z_h^t = \cdot \mid S_h^t)$ for some fixed (t,h), which is denoted by $g_h^*(S_h^t)$. We write $L'(t,h)$ as L' for simplicity. To prove this proposition, we first introduce the transformer architecture and then establish the desired approximation error. As outlined above, the transformer has five components. We first introduce the input embedding as follows.

Input embedding. For each reasoning step in S_h^t, we define the input embedding as

$$\widetilde{z}_{h'}^{t',(0)} = \begin{cases} (z_{h'}^{t'}, h', 0) & \text{for } (t',h') < (t,h-1), \\ (z_{h-1}^t, h-1, 1) & \text{for } (t',h') = (t,h-1). \end{cases} \tag{100}$$

Here the ordering between index tuples is specified in (93). Recall that $L'(t',h')$ is the index of the reasoning step $z_{h'}^{t'}$ in S_h^t. In the embedding in (100), the first coordinate $z_{h'}^{t'}$ is the

content embedding, which stores the actual reasoning step. The second coordinate of $\widetilde{z}_{h'}^{t'}$ indicates the step-index of each reasoning step $z_{h'}^{t'}$, and the last coordinate specifies if it is the last reasoning step. This last coordinate acts as an indicator function because we want to extract the index h and approximate the target function g_h^*. The indicator function is 0 for all steps except the last reasoning step z_{h-1}^t of S_h^t, which helps to locate the step-index h. Thus, the last two coordinates are the *positional embedding* which carries the positional information. In the sequel, we let $X_{\mathrm{NN}}^{(0)} \in \mathbb{R}^{L' \times 3}$ denote the embedding matrix, whose rows are the embedding vectors defined in (100).

Using $X_{\mathrm{NN}}^{(0)}$ as the input, we present the other components of the transformer as follows. Our construction is decomposed into five steps as follows.

- In **Step 1**, we design the extraction module $\mathtt{NN}_{\mathrm{J}}$ to extract the step-index h from the last reasoning step and copy it to all previous reasoning steps in the prompt S_h^t. This module takes $X_{\mathrm{NN}}^{(0)}$ as the input and outputs a matrix in $\mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ in (103). Specifically, the first column of the output matrix corresponds to the context embedding S_h^t. The entries of the second column are all approximately equal to h, the step-index of z_h^t. The rest of the $2|\mathcal{L}|$ columns are all equal to zero vectors.

- In **Step 2**, for all $\widetilde{h} \in \{0, \ldots, H\}$, we construct the approximation modules $G_{\widetilde{h}}$ that produce an *approximation embedding* $\mathtt{embed}_{\widetilde{h}}(S_h^t)$, which is used to approximate the target distributions $g_{\widetilde{h}}^*(S_h^t)$ after a softmax transformation.

- In **Step 3**, for all $\widetilde{h} \in \{0, \ldots, H\}$, we build the selection module $F_{\widetilde{h}}$ to check if the index of the current module, i.e., $\widetilde{h}$, matches the extracted index h. This module approximately adds $\mathtt{embed}_{\widetilde{h}}(S_h^t) \cdot \mathbb{1}\{\widetilde{h} = h\}$ to a memory. As a result, F_H outputs a desired output $\mathtt{embed}_h(S_h^t)$.

- In **Step 4**, we combine the constructions introduced in the first three steps with a softmax output layer to complete the final transformer. Then we analyze the approximation error of the transformer network.

- Finally, we conclude the proof in **Step 5** by verifying that the constructed transformer network belongs to the function class $\mathcal{P}_{\mathrm{LLM}}$ by verifying that the transformer parameters satisfy (10).

Step 1: Extract and copy step index h using module $\mathtt{NN}_{\mathrm{J}}$. In this step, we construct the extraction module $\mathtt{NN}_{\mathrm{J}}$ to extract the step-index h from the input $X_{\mathrm{NN}}^{(0)}$ and copy it to each reasoning step $z_{h'}^{t'}$. Here $X_{\mathrm{NN}}^{(0)}$ is defined in (100). This step is achieved by four transformer submodules. In particular, $\mathtt{NN}_{\mathrm{J}}$ takes $X_{\mathrm{NN}}^{(0)}$ as the input and outputs $X_{\mathrm{NN}}^{(4)} = (S_h^t, \widehat{p}_h, \mathbf{0}) \in \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$. Here the first column is S_h^t which stores all the reasoning steps. The second column $\widehat{p}_h$ is close to $h \cdot \mathbf{1}_{L'}$, which copies the step-index h to every reasoning step. Here $\mathbf{1}_{L'}$ denotes an all-one vector in $\mathbb{R}^{|\mathcal{L}|}$. The last $2|\mathcal{L}|$ columns are all equal to zero vectors.

More specifically, we let $\{\mathtt{NN}_{\mathrm{J},a}\}_{a \in [4]}$ denote the four submodules of $\mathtt{NN}_{\mathrm{J}}$. We define $X_{\mathrm{NN}}^{(a)} = \mathtt{NN}_{\mathrm{J},a}(X_{\mathrm{NN}}^{(a-1)})$ for all $a \in [4]$ as the output matrices of each submodule. These

matrices are in $\mathbb{R}^{L'\times 3}$ for $a \in [3]$ and in $\mathbb{R}^{L'\times(2+2|\mathcal{L}|)}$ for $a = 4$. We let $\widetilde{z}_{h'}^{t',(a)}$, $a \in [4]$, to denote the rows of these matrices. For any (t', h') with $(t', h') < (t, h-1)$, $z_{h'}^{t'}$ is the $L'(t', h'+1)$-th element in S_h^t. Then, starting from $\widetilde{z}_{h'}^{t',(0)}$ defined in (100), the $L'(t', h'+1)$-th rows of these matrices are given by

$$\begin{aligned}
\widetilde{z}_{h'}^{t',(0)} &= (z_{h'}^{t'}, h', 0) && L'(t', h'+1)\text{-th row of } X_{\mathrm{NN}}^{(0)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},1}\\
\widetilde{z}_{h'}^{t',(1)} &= (z_{h'}^{t'}, h'+1, 0), && L'(t', h'+1)\text{-th row of } X_{\mathrm{NN}}^{(1)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},2}\\
\widetilde{z}_{h'}^{t',(2)} &= \big(z_{h'}^{t'}, f_{\mathrm{product}}(h'+1, 0), 0\big), && L'(t', h'+1)\text{-th row of } X_{\mathrm{NN}}^{(2)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},3}\\
\widetilde{z}_{h'}^{t',(3)} &= \big(z_{h'}^{t'}, f_{\mathrm{product}}(h'+1, 0), 1\big), && L'(t', h'+1)\text{-th row of } X_{\mathrm{NN}}^{(3)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},4}\\
\widetilde{z}_{h'}^{t',(4)} &= \big(z_{h'}^{t'}, p_{h'}^{t'}, \mathbf{0}\big), && L'(t', h'+1)\text{-th row of } X_{\mathrm{NN}}^{(4)}.
\end{aligned}$$

Here $f_{\mathrm{product}}(a, b) \approx ab$ is a neural network that approximately implements the product operation using transformer blocks. We specify the construction of f_{product} in Lemma 47. We use $p_{h'}^{t'} \approx h$ for each $(t', h') < (t, h-1)$ to copy the step index h to the embedding of each reasoning step $z_{h'}^{n'}$. Moreover, the last row, i.e., the $L'(t, h)$-th row of these matrices are

$$\begin{aligned}
\widetilde{z}_{h-1}^{t,(0)} &= (z_{h-1}^{t}, h-1, 1) && L'(t, h)\text{-th row of } X_{\mathrm{NN}}^{(0)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},1}\\
\widetilde{z}_{h-1}^{t,(1)} &= (z_{h-1}^{t}, h, 1), && L'(t, h)\text{-th row of } X_{\mathrm{NN}}^{(1)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},2}\\
\widetilde{z}_{h-1}^{t,(2)} &= \big(z_{h-1}^{t}, f_{\mathrm{product}}(h, 1), 1\big), && L'(t, h)\text{-th row of } X_{\mathrm{NN}}^{(2)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},3}\\
\widetilde{z}_{h-1}^{t,(3)} &= \big(z_{h-1}^{t}, f_{\mathrm{product}}(h, 1), 1/2\big), && L'(t, h)\text{-th row of } X_{\mathrm{NN}}^{(3)},\\
&\Downarrow \mathtt{NN}_{\mathrm{J},4}\\
\widetilde{z}_{h-1}^{t,(4)} &= \big(z_{h-1}^{t}, p_{h-1}^{t}, \mathbf{0}\big), && L'(t, h)\text{-th row of } X_{\mathrm{NN}}^{(4)},
\end{aligned}$$

where p_{h-1}^t is close to h.

In the rest of **Step 1**, we present lemmas proving that $\mathtt{NN}_{\mathrm{J},1}, \ldots, \mathtt{NN}_{\mathrm{J},4}$ can be realized by FF or MHA layers. The first submodule $\mathtt{NN}_{\mathrm{J},1}$ adds one to the second coordinate of each input vector. This operation can be realized by a FF layer exactly. As shown in the beginning of Appendix H.3, this can be realized by a single transformer block.

Lemma 40 (Submodule $\mathtt{NN}_{\mathrm{J},1}$) *There exists a FF layer $\mathtt{NN}_{\mathrm{J},1}$ such that*

$$\mathtt{NN}_{\mathrm{J},1}\big(X_{\mathtt{NN}}^{(0)}\big) = X_{\mathtt{NN}}^{(1)}, \text{ where } \widetilde{z}_{h'}^{t',(1)} = \begin{cases} (z_{h'}^{t'}, h'+1, 0) & \text{for } (t', h') < (t, h-1),\\ (z_{h-1}^{t}, h, 1) & \text{for } (t', h') = (t, h-1). \end{cases}$$

Moreover, the Frobenius norms of the weight matrices are bounded by $\sqrt{B^2+H^2+1}\cdot\sqrt{6}$. Thus, this function can represented as a single transformer block.

Proof See Appendix I.4.2 for details. ■

Next, we aim to substitute the second coordinate of each $\widetilde{z}_{h'}^{t',(1)}$ with the product of itself and the third coordinate using $\texttt{NN}_{\text{J},2}$. Namely, we aim to compute $(h'+1)\cdot 0 = 0$ for $(t',h') < (t,h-1)$ and $h\cdot 1 = h$ for $(t',h') = (t,h-1)$. The product operation can be approximately realized by a fully connected neural network with an arbitrarily small error. As a result, $\texttt{NN}_{\text{J},2}$ can be implemented by a composition of multiple transformer blocks.

Lemma 41 (Submodule $\texttt{NN}_{\text{J},2}$) *Let $\epsilon' \in (0,1)$ be a desired accuracy level. There exists fully connected MLP $\texttt{NN}_{\text{J},2}$ with at most $C_p\cdot(\log(H)+\log(1/\epsilon'))$ layers such that*

$$\texttt{NN}_{\text{J},2}\big(X_{\texttt{NN}}^{(1)}\big) = X_{\texttt{NN}}^{(2)},\ \textit{where}\ \widetilde{z}_{h'}^{t',(2)} = \begin{cases} \big(z_{h'}^{t'}, f_{\text{product}}(h'+1,0), 0\big) & \textit{for } (t',h') < (t,h-1), \\ \big(z_{h-1}^{t}, f_{\text{product}}(h,1), 1\big) & \textit{for } (t',h') = (t,h-1), \end{cases} \tag{101}$$

Here C_p is an absolute constant and $f_{\text{product}}:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$ is an approximation of the product operation in the sense that $|f_{\text{product}}(h,1)-h| < \epsilon'$, and $|f_{\text{product}}(h'+1,0)-0| < \epsilon'$ for each $(t',h') < (t,h-1)$. Thus, by Proposition 39, $\texttt{NN}_{\text{J},2}$ can be written as a composition of $C_p\cdot(\log(H)+\log(1/\epsilon'))$ transformer blocks up to a scaling factor. Moreover, the Frobenius norms of the weight matrices are all bounded by $\sqrt{B^2+H^2+6}\cdot\sqrt{H^4\cdot 5+4}$.

Proof See Appendix I.4.3 for details. ■

The third submodule $\texttt{NN}_{\text{J},3}$ modifies the last coordinates of the input vectors by adding a $-1/2$ or 1, which is a simple linear operation and thus can be implemented by a FF layer.

Lemma 42 (Submodule $\texttt{NN}_{\text{J},3}$) *There exists a FF layer $\texttt{NN}_{\text{J},3}$ such that*

$$\texttt{NN}_{\text{J},3}\big(X_{\texttt{NN}}^{(2)}\big) = X_{\texttt{NN}}^{(3)},\ \textit{where}\ \widetilde{z}_{h'}^{t',(3)} = \begin{cases} \big(z_{h'}^{t'}, f_{\text{product}}(h'+1,0), 1\big) & \textit{for } (t',h') < (t,h-1), \\ \big(z_{h-1}^{t}, f_{\text{product}}(h,1), 1/2\big) & \textit{for } (t',h') = (t,h-1). \end{cases}$$

Thus, as a single FF layer, $\texttt{NN}_{\text{J},3}$ can be represented by a single transformer block. The Frobenius norms of the weight matrices are bounded by $\sqrt{B^2+(H+1)^2+1}\cdot\sqrt{6}$.

Proof See Appendix I.4.4 for a detailed proof. ■

In addition to the weights constructed in Lemma 42, the parameter $W_{\text{ff},1}$ in $\texttt{NN}_{\text{J},3}$ is first multiplied with a diagonal matrix to compensate the scaling factor in Lemma 41. Finally, we use an attention layer to copy the step-index h of z_h^t to all previous steps. Moreover, we also use a residual link to ensure the first coordinate remains unchanged.

Lemma 43 (Submodule $\texttt{NN}_{\text{J},4}$) *Let ϵ' denote the error induced by $f_{\text{product}}(\cdot)$ in Lemma 41, then for any $\epsilon \in (2\epsilon', 1)$, there exists a submodule $\texttt{NN}_{\text{J},4}$ such that*

$$\texttt{NN}_{\text{J},4}\big(X_{\texttt{NN}}^{(3)}\big) = X_{\texttt{NN}}^{(4)},\ \textit{where}\ \widetilde{z}_{h'}^{t',(4)} = \begin{cases} (z_{h'}^{t'}, p_{h'}^{t'}, \mathbf{0}) & \textit{for } (t',h') < (t,h-1), \\ (z_{h-1}^{t}, p_{h-1}^{t}, \mathbf{0}) & \textit{for } (t',h') = (t,h-1), \end{cases} \tag{102}$$

where $p_{h'}^{t'}$ is the approximation of the step-index h such that $|p_{h'}^{t'} - h| < \epsilon$ for all $(t', h') \leq (t, h-1)$. Moreover, $\texttt{NN}_{\texttt{J},4}$ is a transformer block with a single-head attention (MHA with $\eta = 1$). The parameters $\{W^Q, W^K, W^V\}$ satisfying $\|W^Q\|_{\mathrm{F}} = 8\log(TH/(\epsilon - 2\epsilon'))$, $\|W^K\|_{\mathrm{F}} = 1$, and $\|W^V\|_{\mathrm{F}} = \sqrt{B^2 + (H+1)^2 + 1}$, and the Frobenius norms of the weight matrices in FF layers are bounded by $\sqrt{B^2 + (H+1)^2 + 6} \cdot \sqrt{6}$.

Proof See Appendix I.4.5 for details. ■

Setting $\epsilon = 1/4$ and $\epsilon' = 1/16$ in Lemma 43 we conclude that we can use a transformer block $\texttt{NN}_{\texttt{J},4}$ to generate the output in (102). The weights of the transformer are bounded by $8\log(8TH)$ in the Frobenius norm. Moreover, for any $(t', h') \leq (t, h-1)$, we have $|p_{h'}^{t'} - h| < \epsilon = 1/4$. Then we write the composition of the four submodules above as

$$\texttt{NN}_{\texttt{J}}(S_h^t) = (S_h^t, \widehat{p}_h, \mathbf{0}) \in \mathrm{R}^{L' \times (2+2|\mathcal{L}|)}, \text{ where } \mathbf{0} \in \mathbb{R}^{L' \times 2|\mathcal{L}|}, \tag{103}$$

where we use $\widehat{p}_h = (p_0^1, \cdots, p_{h-1}^t)^\top \in \mathbb{R}^{L' \times 1}$ to denote the vector that is the second column of $X_{\texttt{NN}}^{(4)}$. The entries of $\widehat{p}_h$ are all close to h in the sense that $\|\widehat{p}_h - h \cdot \mathbf{1}_{L'}\|_\infty < 1/4$.

To summarize, in **Step 1**, we have successfully designed a transformer module $\texttt{NN}_{\texttt{J}}$ to extract the target step-index h from the last reasoning step of the prompt S_h^t, and approximately copy it to all previous reasoning steps in the prompt. We defer the summary of parameters of $\texttt{NN}_{\texttt{J}}$ to **Step 5**. This concludes **Step 1**.

We outline how $\texttt{NN}_{\texttt{J}}(S_h^t)$ is processed by the subsequent transformer blocks. In **Step 2** and **3** we construct modules $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$. Submodule $G_{\widetilde{h}}$ approximates each target function $g_{\widetilde{h}}^*$ and submodule $F_{\widetilde{h}}$ checks if the module index $\widetilde{h}$ matches the step index h. If $\widetilde{h} = h$, it passes along the approximation produced by $G_{\widetilde{h}}$; otherwise, it discards the output. In the final network, blocks $(G_{\widetilde{h}}, F_{\widetilde{h}})_{\widetilde{h}=0}^{H}$ are chained sequentially.

With a slight abuse of notation, we also use $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$ to refer to their outputs, respectively. The input and output of each module are listed as follows:

- $G_{\widetilde{h}} : \mathbb{R}^{L' \times (2+2|\mathcal{L}|)} \to \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ takes a matrix $(S_h^t, \widehat{p}_h, F_{\widetilde{h}-1}[3], F_{\widetilde{h}-1}[4])$ as the input. The columns of this matrix have four components, where S_h^t and $\widehat{p}_h$ are the same as in (103). The last two components $F_{\widetilde{h}-1}[3]$ and $F_{\widetilde{h}-1}[4] \in \mathbb{R}^{L' \times |\mathcal{L}|}$ are the third and fourth components of the output of the previous module $F_{\widetilde{h}-1}$. If $\widetilde{h} = 0$, the input matrix is (103), the output of the extraction module $\texttt{NN}_{\texttt{J}}$. The output of $G_{\widetilde{h}}$ $(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4])$ keeps the first two and the last components unchanged and only changes the third component. That is, $G_{\widetilde{h}}[4] = F_{\widetilde{h}-1}[4]$, and $G_{\widetilde{h}}[3]$ computes the approximation of $g_{\widetilde{h}}^*$ with the third component, which will be specified in **Step 2**.

- Similarly, $F_{\widetilde{h}} : \mathbb{R}^{L' \times (2+2|\mathcal{L}|)} \to \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ takes vector $(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4])$ as the input and produces $(S_h^t, \widehat{p}_h, F_{\widetilde{h}}[3], F_{\widetilde{h}}[4])$. Here, only the last component of columns is changed, i.e., $F_{\widetilde{h}}[3] = G_{\widetilde{h}}[3]$. The last component $F_{\widetilde{h}}[4]$ is constructed iteratively via

$$F_{\widetilde{h}}[4] \approx G_{\widetilde{h}}[3] \cdot \mathbb{1}\{\widetilde{h} = h\} + F_{\widetilde{h}-1}[4]. \tag{104}$$

  We will introduce how to use a transformer to implement (104) in **Step 3**.

Step 2: Construct approximation module $G_{\widetilde{h}}$ that approximates $g^*_{\widetilde{h}}$. In this step, we introduce the submodules $G_{\widetilde{h}} : \mathbb{R}^{L' \times (2+2|\mathcal{L}|)} \to \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ to separately approximate each target function $g^*_{\widetilde{h}}$ in (94) for all $\widetilde{h} \in \{0, \ldots, H\}$. The input and output of each $G_{\widetilde{h}}$ are given by

$$G_{\widetilde{h}}\big(S_h^t, \widehat{p}_h, F_{\widetilde{h}-1}[3], F_{\widetilde{h}-1}[4]\big) = \big(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_{\widetilde{h}}(S_h^t), F_{\widetilde{h}-1}[4]\big),$$

where $\texttt{embed}_{\widetilde{h}}(\cdot)$ is used for the approximation of the target distribution $g^*_{\widetilde{h}}(\cdot)$ in (94). More specifically, for each $\widetilde{h}$, we use $\widehat{g}^*_{\widetilde{h}}(\cdot) = \texttt{softmax}\big(\texttt{embed}_{\widetilde{h}}(\cdot)/\tau\big)$ to denote the output distribution by passing $\texttt{embed}_{\widetilde{h}}(\cdot)$ through a $\texttt{softmax}$ function with a temperature τ. We expect $\widehat{g}^*_h(S_h^t) \approx g^*_h(S_h^t)$ for all $h \in \{0, \ldots, H\}$. Intuitively, we want each module $G_{\widetilde{h}}$ to handle prompts at different steps during the testing stage. We summarize the construction of $G_{\widetilde{h}}$ in the following proposition.

Proposition 44 *Let $\epsilon_\psi, \epsilon_w \in (0,1)$ be two accuracy levels. Under Assumptions 12, 13, and 37, for any $h \in \{0, \ldots, H\}$, there exists a module $G_h \colon \mathbb{R}^{L' \times (2+2|\mathcal{L}|)} \to \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ such that $G_h(S_h^t, \widehat{p}_h, F_3, F_4) = (S_h^t, \widehat{p}_h, \texttt{embed}_h(S_h^t), F_4)$. Here G_h contains $(D_w + D_\psi + 2)$ transformer blocks with*

$$D_\psi = 2C_d B \cdot \big(\log(1/\epsilon_\psi)\big)^2 + \log B, \qquad D_w = 2C_d B \cdot \big(\log(1/\epsilon_w)\big)^2 + \log B, \tag{105}$$

where $C_d > 0$ is an absolute constant and B is the smoothness parameter appearing in Assumption 37. We define a function $\widehat{g}^*_h(S_h^t) = \texttt{softmax}(\texttt{embed}_h(S_h^t)/\tau)$ as the output distribution approximated by the network G_h. Then $\widehat{g}^*_h$ satisfies

$$\max_{S_h^t \in \mathcal{L}^*} \texttt{TV}\big(g^*_h(S_h^t), \widehat{g}^*_h(S_h^t)\big) = \mathcal{O}\big(\epsilon_w + 256^{D_w} \cdot \epsilon_\psi\big). \tag{106}$$

For each G_h, the maximum width of the FF layers is $18|\mathcal{L}| + 4$, the maximum Frobenius norm of weight matrices is

$$C_F \cdot B_0 \cdot 16^{\max\{D_\psi, D_w\}} \cdot |\mathcal{L}|^{3/2},$$

where $B_0 = \sqrt{B^2 + H^2 + |\mathcal{L}| \cdot C_A^2}$ and $C_F > 1$ is an absolute constant.

Proof See Appendix I.4.6 for a detailed proof. ■

This proposition states that for any $h \in [H]$, we can use a transformer G_h to approximate $g^*_h(S_h^t)$ accurately. The number of transformer blocks in G_h is determined by the desired accuracy levels ϵ_ψ and ϵ_w. Note that the approximation accuracy grows exponentially in D_w. This will not be a problem when D_w is small compared to D_ψ. To see this, we can rewrite the upper bound in (106) in terms of the depth of the transformer. Specifically, let $\overline{D}_\psi$ and $\overline{D}_w$ be two sufficiently large integers. Setting

$$\epsilon_\psi = \exp\Big(-\sqrt{(\overline{D}_\psi - \log B)/B}\Big) \qquad \text{and} \qquad \epsilon_w = \exp\Big(-\sqrt{(\overline{D}_w - \log B)/B}\Big) \tag{107}$$

in Proposition (44), we know that there exists $\{G_h\}_{h=0}^H$ such that

$$\max_{S_h^t \in \mathcal{L}^*} \mathtt{TV}\big(g_h^*(S_h^t), \widehat{g}_h^*(S_h^t)\big) = \mathcal{O}\bigg(\exp\Big(-\sqrt{\overline{D}_w/B}\Big) + \exp\Big(-\sqrt{\overline{D}_\psi/B} + 6 \cdot D_w\Big)\bigg) \tag{108}$$

for all $h \in \{0, \ldots, H\}$. Here we use the fact that $\log(256) < 6$. Moreover, (105) implies that the number of transformer blocks satisfies $D_\psi \le C_d \cdot \overline{D}_\psi$ and $D_w \le C_d \cdot \overline{D}_w$ for some absolute constant C_d. Note that $\overline{D}_\psi$ and $\overline{D}_w$ can be chosen arbitrarily. We can set $\overline{D}_\psi = \mathcal{O}(\overline{D}_w^2)$ so that the second term in (108) becomes negligible compared to the first term. We will determine $\overline{D}_\psi$ and $\overline{D}_w$ to obtain the final error in **Step 4**.

Step 3: Construct selection module $F_{\widetilde{h}}$. In this step, we introduce a sequence of transformer modules $\{F_{\widetilde{h}}\}_{\widetilde{h}=1}^H$. Each $F_{\widetilde{h}}$ is a mapping from $\mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ to itself, and its input and output are given by

$$F_{\widetilde{h}}(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4]) = \big(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], f_{\text{product}}\big(G_{\widetilde{h}}[3], \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big) + G_{\widetilde{h}}[4]\big). \tag{109}$$

That is, $F_{\widetilde{h}}$ takes the output of $G_{\widetilde{h}}$ as the input, and it keeps the first three components of the columns unchanged. The last component, i.e., the last $|\mathcal{L}|$ columns, are used a "memory". Note that $G_{\widetilde{h}}[4] = F_{\widetilde{h}-1}[4]$. The last component of the output of $F_{\widetilde{h}}$, $F_{\widetilde{h}}[4]$, can be written as

$$F_{\widetilde{h}}[4] = \mathbf{1}_{L'}^\top f_{\text{product}}\big(\mathtt{embed}_{\widetilde{h}}(S_h^t), \mathbb{1}\{\widetilde{h} = h\}\big) + F_{\widetilde{h}-1}[4], \tag{110}$$

where f_{product} is a transformer that approximately implements the product operation. Thus, $F_{\widetilde{h}}$ first checks if the module index $\widetilde{h}$ matches the target step index h, and then writes $\mathtt{embed}_{\widetilde{h}}(\cdot)$ to the memory if $\widetilde{h} = h$. Thus, by (110) we have

$$F_H[4] = \sum_{\widetilde{h}=0}^{H} \mathbf{1}_{L'}^\top f_{\text{product}}\big(\mathtt{embed}_{\widetilde{h}}(S_h^t), \mathbb{1}\{\widetilde{h} = h\}\big) \approx \mathbf{1}_{L'}^\top \mathtt{embed}_h(S_h^t). \tag{111}$$

To implement each $F_{\widetilde{h}}$, we starting from the input $(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4])$, denoted by $X_F^{\widetilde{h},(0)}$, we perform the following three steps:

(i) First, we use a sequence of transformer blocks to represent the indicator $\mathbb{1}\{\widetilde{h} = h\}$, and then append it to the end of $X_F^{\widetilde{h},(0)}$. Thus, we have

$$X_F^{\widetilde{h},(1)} = \big(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4], \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big). \tag{112}$$

Here the indicator is obtained by feeding $\widehat{p}_h$ to a trapezoid-shaped function.

(ii) Then we feed $X_F^{\widetilde{h},(1)}$ to the product module introduced in Lemma 47 to multiply each entry of $G_{\widetilde{h}}[3]$ with $\mathbb{1}\{\widetilde{h} = h\}$. The resulting output is

$$X_F^{\widetilde{h},(2)} = \big(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4], f_{\text{product}}\big(G_{\widetilde{h}}[3], \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big)\big). \tag{113}$$

(iii) Finally, we pass $X_F^{\widetilde{h},(2)}$ to a linear layer, which adds the last two components of $X_F^{\widetilde{h},(2)}$ and obtain

$$X_F^{\widetilde{h},(3)} = \big(S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], f_{\text{product}}\big(G_{\widetilde{h}}[4], \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big) + G_{\widetilde{h}}[4]\big).$$

Details of (i). We present the details of these three steps as follows. We first focus on how to construct the indicator $\mathbb{1}\{h = \widetilde{h}\}$. Recall that by Lemma 43 we show that $\widehat{p}_h$ satisfies $\|\widehat{p}_h - h \cdot \mathbf{1}_{L'}\|_\infty < 1/4$. Thus, each entry of $\widehat{p}_h$ is in $(h-1/4, h+1/4)$. For any $\widetilde{h}$, we want to construct a neural network $f_{\widetilde{h}} \colon \mathbb{R} \to [0,1]$ such that $f_{\widetilde{h}}(x) = 1$ if $|x - \widetilde{h}| \le 1/4$ and $f(x) = 0$ if $|x - \widetilde{h}| \ge 3/4$. Then applying $f_{\widetilde{h}}$ to each entry of $\widehat{p}_h$, we have $f_{\widetilde{h}}(\widehat{p}_h) = \mathbb{1}\{h = \widetilde{h}\} \cdot \mathbf{1}_{L'}$.

Such a $f_{\widetilde{h}}$ can be constructed by a trapezoid-shaped function, which has value one in $[h-\epsilon, h+\epsilon]$, zero when $|x-h| > 1-\epsilon$, and a linear function in between. Here we can set $\epsilon = 1/4$. See Figure 15 for an illustration of two trapezoid-shape functions. The following lemma shows that such trapezoid-shaped functions can be implemented by a FF layer.

Lemma 45 (Trapezoid module) *For any $h \in \{0, \ldots, H\}$ and any $\epsilon \in (0, 1/2)$, we define a trapezoid-shaped function $f_h \colon \mathbb{R} \to [0,1]$ as*

$$f_h(x) = \begin{cases} 1 & \text{for } |x-h| \le \epsilon, \\ -(|x-h| - \epsilon)/(1-2\epsilon) + 1 & \text{for } \epsilon < |x-h| \le 1-\epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Then there exists a neural network f that is identical to f_h. Moreover, f is a composition of two FF layers, each with no more than 10 neurons, and the entries of the weight matrices and bias vectors are bounded by $H+1$ in magnitude.

Proof See Appendix I.4.7 for details. ■

We apply this lemma with $\epsilon = 1/4$ and apply $f_{\widetilde{h}}$ to each entry of $\widehat{p}_h$ to obtain $\mathbb{1}\{h = \widetilde{h}\} \cdot \mathbf{1}_{L'}$,which becomes the last component of $X_F^{\widetilde{h},(1)}$ in (112). Moreover, to preserve the first three components of the input $X_F^{\widetilde{h},(0)}$, we apply Lemma 48, which implies that we can use a single FF layer to map $X_F^{\widetilde{h},(0)}$ to its first three components. The number of neurons in this FF layer is bounded by $2(2 + 2|\mathcal{L}|)$. Thus, we can concatenate these two networks and obtain a larger network that maps $X_F^{\widetilde{h},(0)}$ to $X_F^{\widetilde{h},(1)}$. Moreover, such a network has one FF layer, and the maximum width of the weight matrices is bounded by $10 + 2(2 + 2|\mathcal{L}|)$. As a result, the weight matrices of this feed-forward neural network are all bounded by $\sqrt{B^2 + H^2 + 2|\mathcal{L}| \cdot (C_A+1)^2} \cdot \sqrt{10 + 2(2+2|\mathcal{L}|)} \le \sqrt{B^2 + H^2 + 2|\mathcal{L}| \cdot (C_A+1)^2} \cdot (4 + 2\sqrt{|\mathcal{L}|})$ in terms of the Frobenius norm.

Details of (ii). Then, to get $X_F^{\widetilde{h},(2)}$ defined in (113), we pass $X_F^{\widetilde{h},(1)}$ to the product module in Lemma 47 to multiply $G_{\widetilde{h}}[3]$ and $\mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top$ in an elementwise fashion. Similar to the implementation of $\texttt{NN}_{\text{J},2}$ introduced in Lemma 41, the product module f_{product} here can be implemented as an MLP. More concretely, recall that Assumption 37 states that each

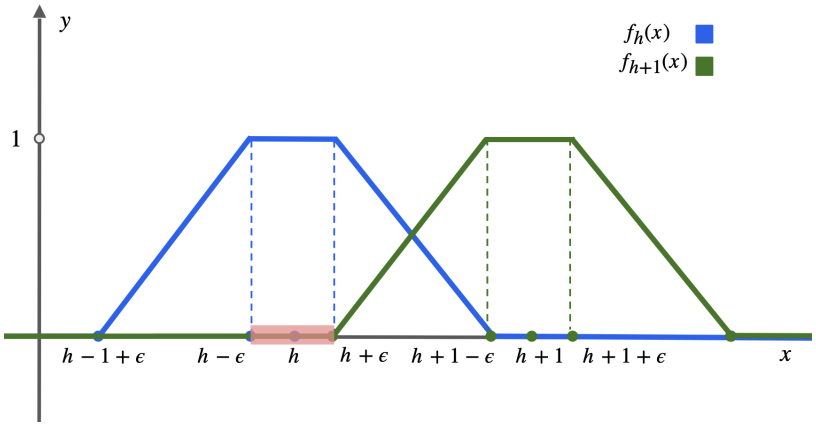


Figure 15: An illustration of two trapezoid-shaped functions f_h (blue) and f_{h+1} (green), i.e., $f_{\widetilde{h}}$ with $\widetilde{h} = h$ and $\widetilde{h} = h+1$. Observe that for any $x \in [h-\epsilon, h+\epsilon]$, $f_{\widetilde{h}}(x) = 1$ if and only if $\widetilde{h} = h$, and $f_{\widetilde{h}}(x) = 0$ for any other $\widetilde{h} \neq h$. Thus, when we apply each $f_{\widetilde{h}}$ to entries of $\widehat{p}_h$, when $\epsilon = 1/4$, we have $\mathbb{1}\{h = \widetilde{h}\}$. Each entry of $\widehat{p}_h$ falls within the highlighted region on the x-axis.

$\tau \cdot \log w^*_{h,i}$ is bounded by C_A in terms of the ℓ_∞-norm. Then, as shown in the proof of Proposition 44 in Appendix I.4.6, we have

$$\left| \big(\texttt{embed}_{\widetilde{h}}(S_h^t)\big)[i] - \tau \cdot \log w_{\widetilde{h},i} \left(\frac{1}{L'} \left(\sum_{(i,j)=(1,0)}^{(t,h-1)} \psi_{\widetilde{h}}(z_h^i) \right) \right) \right| \leq \epsilon_w + 256^{D_w} \cdot \epsilon_\psi \tag{114}$$

for all $i \in [|\mathcal{L}|]$. Here $(\texttt{embed}_{\widetilde{h}}(S_h^t))[i]$ is the i-th entry of $\texttt{embed}_{\widetilde{h}}(S_h^t)$, which is defined in the same way as in (161), but with $\widehat{f}_h^w$ and $\widehat{\psi}_h^*$ replaced by $\widehat{f}_{\widetilde{h}}^w$ and $\widehat{\psi}_{\widetilde{h}}^*$. Besides, as we will show later, we set ϵ_ψ and ϵ_w as in (107) so that the right-hand side of (114) is much smaller than one. As a result, we have $\|\texttt{embed}_{\widetilde{h}}(S_h^t)\|_\infty \leq 1 + C_A$. Thus, combining this fact with Lemma 47, we conclude that, there exists an MLP $f^{\widetilde{h}}_{\text{product}}$ such that for any $\epsilon_p \in (0,1)$,

$$\|f^{\widetilde{h}}_{\text{product}}\big(\texttt{embed}_{\widetilde{h}}(S_h^t), \mathbb{1}\{\widetilde{h} = h\}\big) - \texttt{embed}_{\widetilde{h}}(S_h^t) \cdot \mathbb{1}\{\widetilde{h} = h\}\|_\infty < \epsilon_p \tag{115}$$

and any $\widetilde{h} \in \{0, \ldots, H\}$, where $f^{\widetilde{h}}_{\text{product}}$ has at most $D_p = C_p \cdot (\log(C_A + 1) + \log(1/\epsilon_p))$ FF layers, where C_p is an absolute constant. Moreover, we remark that the module $f^{\widetilde{h}}_{\text{product}}$ is different from the one constructed in Lemma 41, and different for each $0 \leq \widetilde{h} \leq H$.

Moreover, we need to write such a product module as a composition of transformer blocks using Proposition 39. At the same time, we adopt Lemma 48 to preserve the first four components of $X_F^{\widetilde{h},(1)}$, namely $S_h^t, \widehat{p}_h, G_{\widetilde{h}}[3], G_{\widetilde{h}}[4]$ using FF layers. Specifically, we apply Proposition 39 by setting the scaling factors $B_0 = \sqrt{B^2 + H^2 + 2 \cdot |\mathcal{L}| \cdot (C_A + 1)^2}$, $\{B_\ell = \sqrt{B_0^2 + 5|\mathcal{L}|}\}_{\ell=1}^{D_p - 1}$, $B_{D_p} = \sqrt{B_0^2 + (C_A + 1)^2}$. As a result, the function that maps

$X_F^{\widetilde{h},(1)}$ to $X_F^{\widetilde{h},(2)}$ can be implemented by a composition of D_p transformer blocks. Each block has at most $5|\mathcal{L}| + 2(2 + 2|\mathcal{L}|)$ neurons in the FF layer, where $5|\mathcal{L}|$ are used for the product operation, and $2(2 + 2|\mathcal{L}|)$ are used for preserving the input of $X_F^{\widetilde{h},(1)}$. According to Lemma 47, the Frobenius norm of weight matrices are bounded by $\sqrt{B_0^2 + 5|\mathcal{L}|} \cdot \sqrt{2(2 + 2|\mathcal{L}|) + (C_A + 1)^4 \cdot 5|\mathcal{L}|} \leq C' \cdot (C_A + 1)^3 \cdot |\mathcal{L}|$ for some absolute constant C'.

Details of (iii). Finally, we pass $X_F^{\widetilde{h},(2)}$ through a linear layer to add $f_{\text{product}}\big(G_{\widetilde{h}}[3], \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big)$ with $G_{\widetilde{h}}[4]$. Moreover, we adopt Lemma 48 to preserve the first three components of $X_F^{\widetilde{h},(2)}$. Therefore, the maximum Frobenius norm of the weight matrix of this module is $\sqrt{B^2 + H^2 + 3 \cdot |\mathcal{L}| \cdot (C_A + 1)^2} \cdot \sqrt{2(2 + |\mathcal{L}|) + 4|\mathcal{L}|}$, where the first term results from the scaling trick, and the second term results form the linear operation.

Combining (i)–(iii). Combining these three steps above, we obtain the selection module $F_{\widetilde{h}}$ shown in (109) and (110). By (111), the output of F_H is given by

$$\bigg(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_H(S_h^t), \sum_{\widetilde{h}=1}^{H} f_{\text{product}}\big(\texttt{embed}_{\widetilde{h}}(S_h^t), \mathbb{1}\{\widetilde{h} = h\} \cdot \mathbf{1}_{L'}^\top\big)\bigg). \tag{116}$$

For the ease of notation, we use $\widetilde{\texttt{embed}}(S_h^t)$ to denote last component of (116), and we expect $\widetilde{\texttt{embed}}(S_h^t) \approx \texttt{embed}_h(S_h^t)$.

Finally, we calculate the depth and norms of the weight matrices of $F_{\widetilde{h}}$. According to Lemma 45, the trapezoid module has a depth of 1 and a maximum Frobenius norm of $\sqrt{B^2 + H^2 + 2 \cdot |\mathcal{L}| \cdot (C_A + 1)^2}(4 + 2\sqrt{|\mathcal{L}|})$. The linear module also has a depth of 1 and a maximum Frobenius norm of $\sqrt{B^2 + H^2 + 3 \cdot |\mathcal{L}| \cdot (C_A + 1)^2}(2 + \sqrt{6|\mathcal{L}|})$. The product module has a maximum Frobenius norm of $C' \cdot (C_A + 1)^3 \cdot |\mathcal{L}|$ and depth D_p, which will be determined in **Step 4**. Overall, each $F_{\widetilde{h}}$ module has a depth of $D_f = D_p + 2$ and maximum Frobenius norm of weight matrices as $C' \cdot (C_A + 1)^3 \cdot |\mathcal{L}|$.

Step 4: Compute the approximation error. The last component of F_H, $\widetilde{\texttt{embed}}(S_h^t)$, is then fed into a softmax layer to generate the final output. In this step, we characterize the approximation error $\|g_h^*(S_h^t) - \texttt{softmax}(\widetilde{\texttt{embed}}(S_h^t)/\tau)\|_1$. Here g_h^* in (94) refers to the target distribution, and $\texttt{softmax}(\widetilde{\texttt{embed}}(S_h^t)/\tau)$ refers to the output distribution.

First, we handle the error induced by the product operation in $F_{\widetilde{h}}$, which is established in (115). Let $\overline{D}_p$ be an integer and we set $\epsilon_p = (C_A + 1) \cdot \exp(-\overline{D}_p/C_p)$ in (115), by triangle inequality we conclude that

$$\big\|\widetilde{\texttt{embed}}(S_h^t) - \texttt{embed}_h(S_h^t)\big\|_\infty < H\epsilon_p. \tag{117}$$

Moreover, with number of transformer blocks used to implemnt f_{product}, D_p, satisfies $D_p \leq \overline{D}_p$. Since $\texttt{softmax}$ is Lipschitz continuous, as shown in Lemma 54, we conclude that

$$\begin{aligned}&\big\|\texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big) - \texttt{softmax}\big(\texttt{embed}_h(S_h^t)/\tau\big)\big\|_1 \\ &\quad\leq 2\big\|\widetilde{\texttt{embed}}(S_h^t)/\tau - \texttt{embed}_h(S_h^t)/\tau\big\|_\infty \leq 2H\epsilon_p/\tau,\end{aligned} \tag{118}$$

where the second inequality follows from (117).

Therefore, combining Proposition 44 and (118), for any prompt S_h^t with length $L' \le L$, the approximation error of the transformer is bounded by

$$\begin{aligned}
&\big\|g_h^*(S_h^t) - \texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big)\big\|_1 \\
&\quad \le \big\|g_h^*(S_h^t) - \texttt{softmax}\big(\texttt{embed}_h(S_h^t)/\tau\big)\big\|_1 \\
&\qquad + \big\|\texttt{softmax}\big(\texttt{embed}_h(S_h^t)/\tau\big) - \texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big)\big\|_1 \\
&\quad \le 2\epsilon_w + 2\cdot 256^{\overline{D}_w}\cdot\epsilon_\psi + 2H\epsilon_p/\tau. \qquad (119)
\end{aligned}$$

Here ϵ_w and ϵ_ψ are chosen as in (107) and ϵ_p is defined above. Thus, we can equivalently write the approximation error in terms of the parameters $\overline{D}_\psi$, $\overline{D}_w$, and $\overline{D}_p$ as

$$\begin{aligned}
&\big\|g_h^*(S_h^t) - \texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big)\big\|_1 \qquad (120)\\
&= \mathcal{O}\bigg(\exp\Big(-\sqrt{\overline{D}_w/B}\Big) + \exp\Big(-\sqrt{\overline{D}_\psi/B} + 6\cdot\overline{D}_w\Big) + H/\tau\cdot\exp\big(-\overline{D}_p/C_p + \log(C_A+1)\big)\bigg),
\end{aligned}$$

where we use (108), (119), and the definition of ϵ_p. Here the number of transformer blocks of each module among $\{G_j, F_j\}_{j=1}^H$ satisfy $D_w = \mathcal{O}(\overline{D}_w)$, $D_\psi \le C\overline{D}_\psi$, and $D_p \le \overline{D}_p$, where $C > 0$ is an absolute constant.

To get an explicit upper bound, we choose $\overline{D}_\psi$, $\overline{D}_w$, and $\overline{D}_p$ properly to balance the three terms in the right-hand side of (120). Specifically, we require

$$\overline{D}_\psi \ge \big(6\sqrt{B}\cdot\overline{D}_w + \sqrt{\overline{D}_w}\big)^2, \qquad (\overline{D}_p/C_p - \log(C_A+1))^2 \ge \overline{D}_w/B, \qquad (121)$$

and let $\overline{D}_g = (\overline{D}_\psi + \overline{D}_w + 2) + (\overline{D}_p + 2)$ denote a parameter that characterizes the total number of transformer blocks in each $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$ together. Note that the actual total number of transformer blocks in $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$ is $\mathcal{O}(\overline{D}_g)$. To satisfy (121), we can set

$$\overline{D}_w = \sqrt{\overline{D}_g}/(24\sqrt{B}), \qquad \overline{D}_p = \frac{C_p}{\sqrt{24}B^{3/4}}\cdot\sqrt{\overline{D}_g} + C_p\cdot\log(C_A+1), \qquad \overline{D}_\psi = \overline{D}_g - \overline{D}_w - \overline{D}_p - 4. \qquad (122)$$

Then for sufficiently large $\overline{D}_g$, we have that (120) is dominated by the first term:

$$\begin{aligned}
\big\|g_h^*(S_h^t) - \texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big)\big\|_1 &= \mathcal{O}\bigg(\exp\big(-\sqrt{\overline{D}_w/B}\big)\bigg) \\
&= \mathcal{O}\bigg(\exp\bigg(-\frac{\overline{D}_g^{1/4}}{5B}\bigg)\bigg), \qquad (123)
\end{aligned}$$

where the second inequality follows from (122), the fact that $\sqrt{24} < 5$, and the relaxation of the exponent of B for notational clarity, assuming $B \ge 1$.

Finally, we convert the ℓ_1-norm upper bound in (123) into a bound in terms of the KL divergence. Let $\mathbb{P}_{\widehat{\rho}}(\cdot \mid S_h^t)$ denote $\texttt{softmax}\big(\widetilde{\texttt{embed}}(S_h^t)/\tau\big)$, where $\widehat{\rho}$ refers to the parameter that specifies the transformer we constructed in the first three steps of the proof, which consists of a step-index extraction module $\texttt{NN}_\texttt{J}$, and H pairs of modules $\{G_{\widetilde{h}}, F_{\widetilde{h}}\}_{\widetilde{h}=0}^H$. We

first note that if $\texttt{TV}(\mathbb{P}(\cdot \,|\, S_h^t), \mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^t)) = \varepsilon$ with ε sufficiently small such that $\varepsilon < c_0/2$, where c_0 comes from Assumption 13. Under this lemma, we can bound the likelihood ratio $\mathbb{P}(z = \cdot \,|\, S_h^t)/\mathbb{P}_{\widehat{\rho}}(z = \cdot \,|\, S_h^t)$ for each $z \in \mathcal{L}$ by

$$\log \frac{c_0}{c_0 + 2\epsilon} \le \log \frac{\mathbb{P}(\cdot \,|\, S_h^t)}{\mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^t)} \le \log \frac{c_0 + 2\epsilon}{c_0} \le \frac{2\epsilon}{c_0}.$$

Therefore we conclude that, when $\overline{D}_g$ is sufficiently large, there exists a transformer with parameter $\widehat{\rho}$ such that, for any $t \in [T]$ and $h \in \{0, \dots H\}$,

$$\max_{S_h^t \in \mathcal{L}^*} \texttt{KL}\big(\mathbb{P}(\cdot \,|\, S_h^t), \mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^t)\big) = \mathcal{O}\bigg(\exp\bigg(- \frac{\overline{D}_g^{1/4}}{5B} \bigg) \bigg),$$

where the number of transformer blocks is at most $C \cdot \overline{D}_g$ for some absolute constant C.

Step 5: Verify that the constructed transformer is in $\mathcal{P}_{\mathrm{LLM}}$. Finally, we verify that the constructed transformer is in $\mathcal{P}_{\mathrm{LLM}}$. To this end, for each module of the transformer, we explicitly characterize the width and norms of the weight matrices. Recall that the transformer contains an embedding module $\texttt{NN}_{\texttt{J}}$, $\{G_{\widetilde{h}}, F_{\widetilde{h}}\}_{\widetilde{h}=0}^{H}$, and a softmax output layer.

- **Module $\texttt{NN}_{\texttt{J}}$.** The module $\texttt{NN}_{\texttt{J}}$ consists of four submodules, which are two linear modules, a product module realized via FF layers, and a MHA module. The two linear modules in Lemmas 40 and 42 have Frobenius norm of weight matrices upper bounded by $\sqrt{B^2 + (H+1)^2 + 1} \cdot \sqrt{6}$. According to Lemma 41, the product module consists of at most $C(\log(H) + \log(16))$ layers with maximum Frobenius norm $\sqrt{B^2 + H^2 + 6} \cdot \sqrt{H^4 \cdot 5 + 4}$. According to Lemma 43, the extraction module has the maximum Frobenius norm of weight matrices as $\max\{8\log(8TH), \sqrt{B^2 + (H+1)^2 + 1}\}$ for the MHA layer and $\sqrt{B^2 + (H+1)^2 + 6} \cdot \sqrt{6}$ for the FF layer. In conclusion, the module $\texttt{NN}_{\texttt{J}}$ has total depth of at most $\overline{D}_j = C_p \log(3H) + 3$ for some constant $C_p > 0$, the Frobenius norms for FF layers are upper bounded by $\sqrt{B^2 + (H+1)^2 + 6} \cdot \sqrt{H^4 \cdot 5 + 4}$ and those for MHA layers are upper bounded by $\max\{8\log(8TH), \sqrt{B^2 + (H+1)^2 + 1}\}$.
- **Module $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$.** The parameter constraint for any $G_{\widetilde{h}}$ is specified in Proposition 44. According to **Step 3**, each $F_{\widetilde{h}}$ has maximum Frobenius norm of weight matrices as $C' \cdot (C_A + 1)^3 \cdot |\mathcal{L}|$. The depth of each submodule pair $G_{\widetilde{h}}$ and $F_{\widetilde{h}}$ is specified in (122).

Let $\overline{D}$ denote a sufficient large integer. Consider any parameter class $\mathcal{P}_{\mathrm{LLM}}$ in (10) with $B_S \ge (C_A + 1) \cdot \sqrt{|\mathcal{L}|}$, $B_M \ge \max\{8\log(8TH), \sqrt{B^2 + (H+1)^2 + 1}\}$, $d_F \ge 4 + 18|\mathcal{L}|$ and

$$B_F \ge C_F \cdot \sqrt{B^2 + H^2 + C_A^2 \cdot |\mathcal{L}|} \cdot 16^{\overline{D}'} \cdot |\mathcal{L}|^{3/2},$$

where $\overline{D}' = (\overline{D} - C_p \log(3H))/(H+1)$, $C_p, C_F > 0$ are absolute constants, and C_A comes from Assumption 37. Then under Assumptions 12,13, and 37, there exists a transformer with at most $\mathcal{O}(\overline{D})$ transformer blocks and parameter $\rho^* \in \mathcal{P}_{\mathrm{LLM}}$ such that

$$\max_{S_h^t \in \mathcal{L}^*} \texttt{KL}\big(\mathbb{P}(z_h^t = \cdot \,|\, S_h^t), \mathbb{P}_{\rho^*}(z_h^t = \cdot \,|\, S_h^t)\big) = O\bigg(\exp\bigg(- \frac{\big(\overline{D} - C\log(2H))/H\big)^{1/4}}{5B} \bigg) \bigg),$$

for any $t \in [T], 0 \le h \le H$, where $C > 0$ is some absolute constant. Therefore, we conclude the proof.

■

H.4 Proof of Corollary 18

Proof Recall that Lemma 4 decomposes the CoT error $\texttt{err}_{\mathrm{CoT}}$ into $\texttt{err}_{\mathrm{pre}}$ and $\texttt{err}_{\mathrm{prompt}}$. This proof consists of two steps. We first control the expected pretraining error $\mathbb{E}_{\mathbb{P}_{\mathrm{CoT}}}[\texttt{err}_{\mathrm{pre}}]$ under the distribution $\mathbb{P}_{\mathrm{CoT}}$ with OOD queries. Then we consider the prompting error $\mathbb{E}_{\mathbb{P}_{\mathrm{CoT}}}[\texttt{err}_{\mathrm{prompt}}]$ in the second step.

Step 1: Control expected pretraining error under distribution shift. In this step, we evaluate $\mathbb{E}_{\mathbb{P}_{\mathrm{CoT}}}[\texttt{err}_{\mathrm{pre}}]$, the expected pretraining error $\texttt{err}_{\mathrm{pre}}$ defined in (9) with the expectation taken under $\texttt{prompt}_{\mathrm{CoT}}(n) \sim \mathbb{P}_{\mathrm{CoT}}$. We first decompose $\texttt{err}_{\mathrm{pre}}$ into a sum of errors incurred in each reasoning step and then take expectations with respect to $\mathbb{P}_{\mathrm{CoT}}$. We adopt the following lemma to decompose $\texttt{err}_{\mathrm{pre}}$ into a sum of KL divergences.

Lemma 46 (KL decomposition) *Recall that* $\texttt{prompt}^h_{\mathrm{CoT}}(n) = \{\Upsilon_n, z^{\mathrm{test}}_{0:h-1}\}$ *consists of* n *examples and the first* $h-1$*-th inferred steps for the testing example. Then we have*

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \qquad (124)\\ &\le \sum_{h=1}^{H} \mathbb{E}_{z^{\mathrm{test}}_{1:h-1} \sim \mathbb{P}(\cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))} \Big[\texttt{KL}\big(\mathbb{P}(z^{\mathrm{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n))\big)\Big].\end{aligned}$$

Proof See Appendix I.5.1 for detailed proof. ■

This Lemma states that we can upper bound the pretraining error $\texttt{err}_{\mathrm{pre}}$ by aggregating the pretraining error at each step of inference. We apply (90) with Lemma 51 to convert each KL divergence in (124) into TV distances. Namely, we have

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = \cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))\big) \qquad (125)\\ &\le \sum_{h=1}^{H} \mathbb{E}_{z^{\mathrm{test}}_{1:h-1} \sim \mathbb{P}(\cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))} \Big[2(3+b^*) \cdot \texttt{TV}\big(\mathbb{P}(z^{\mathrm{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\mathrm{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\mathrm{CoT}}(n))\big)\Big].\end{aligned}$$

The number b^* comes from (90), which upper bounds the log density difference $|\log \mathbb{P}(z|S) - \log \mathbb{P}_{\widehat{\rho}}(z|S)$ for any $z \in \mathcal{L}, S \in \mathcal{L}^*$.

We take the expectation of (125) with respect to $\texttt{prompt}_{\mathrm{CoT}}(n) \sim \mathbb{P}_{\mathrm{CoT}}$. Notice that different parts of $\texttt{prompt}^h_{\mathrm{CoT}}(n)$ have different distributions: $z^{\mathrm{test}}_{1:(h-1)} \sim \mathbb{P}(\cdot \,|\, \texttt{prompt}_{\mathrm{CoT}}(n))$,

$z_0^{\text{test}} \sim \mu(\cdot \mid \Upsilon_n)$, $\Upsilon_n \sim \mathbb{P}(\cdot \mid \theta^*)$. We take these expectations sequentially:

$$\begin{aligned}
&\sum_{h=1}^{H} \mathbb{E}_{\theta^*} \mathbb{E}_{\mu} \mathbb{E}_{z_{1:(h-1)}^{\text{test}} \sim \mathbb{P}(\cdot \mid \texttt{prompt}_{\text{CoT}}(n))} \Big[\texttt{KL}\big(\mathbb{P}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)), \mathbb{P}_{\widehat{\rho}}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)) \big) \Big] \\
&\quad \le \sum_{h=1}^{H} 2(3+b^*) \cdot \int_{\mathcal{L}^*} \Big(\texttt{TV}\big(\mathbb{P}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)), \mathbb{P}_{\widehat{\rho}}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)) \big) \\
&\qquad \cdot \pi(\theta^*)^{-1} \cdot \mathbb{P}(\Upsilon_n) \cdot \kappa \cdot \mathbb{P}(z_0^{\text{test}} = \cdot \mid \Upsilon_n) \cdot \mathbb{P}(z_{1:(h-1)}^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}(n)) \Big) \mathrm{d}\texttt{prompt}_{\text{CoT}}^h(n) \\
&\quad = 2(3+b^*)\kappa\pi(\theta^*)^{-1} \sum_{h=1}^{H} \mathbb{E}_{\mathbb{P}} \Big[\texttt{TV}\big(\mathbb{P}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)), \mathbb{P}_{\widehat{\rho}}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)) \big) \Big]
\end{aligned} \tag{126}$$

with probability at least $1-\delta$. In the first inequality, we integrate over $\texttt{prompt}_{\text{CoT}}^h(n) \in \mathcal{L}^*$. This inequality is a result of (125), which transforms the KL distances into TV distances, and a change of distributions from $\mathbb{P}_{\text{CoT}}$ to $\mathbb{P}$. Notice that we have $\mathbb{P}(\Upsilon_n \mid \theta^*)\pi(\theta^*) \le \mathbb{P}(\Upsilon_n)$ due to the discreteness of Θ. Also note that, by Assumption 17, $\mu(z_0^{\text{test}} = \cdot \mid \Upsilon_n)$ can be bounded by $\mathbb{P}(z_0^{\text{test}} = \cdot \mid \Upsilon_n)$ by introducing an additional factor κ. Consequently, (126) shifts the expectation under $\mathbb{P}_{\text{CoT}}$ towards $\mathbb{P}$ by scaling some constants.

Next, we upper bound (126) using the analysis of pretraining error in Proposition 14. Recall that we introduce the notation $\mathbb{E}_{S \sim \mathcal{D}}$ in (11), which involves an expectation over N i.i.d. documents, each has T examples. Since $\texttt{prompt}_{\text{CoT}}(n)$ only has n examples and $T \ge n+1$ and the N documents in $\mathcal{D}_{N,T}$ are i.i.d., we have

$$\begin{aligned}
&\sum_{h=1}^{H} \mathbb{E}_{\mathbb{P}} \Big[\texttt{TV}\big(\mathbb{P}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)), \mathbb{P}_{\widehat{\rho}}(z_h^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}^h(n)) \big) \Big] \\
&\quad \le \sum_{t=1}^{T} \sum_{h=1}^{H} \Big[\texttt{TV}\big(\mathbb{P}(z_h^{t,1} = \cdot \mid S_h^{t,1}), \mathbb{P}_{\widehat{\rho}}(\mathbb{P}(z_h^{t,1} = \cdot \mid S_h^{t,1})) \big) \Big] \le T \cdot (H+1) \cdot \Delta_{\text{pre}}(N,T,\delta),
\end{aligned} \tag{127}$$

with probability at least $1-\delta$. Here the first inequality holds because the left-hand side is only a single term in the right-hand side summation with $t=n$, and the second inequality follows from Proposition 14, Combining (126) and (127), we upper bound the expectation of pretraining error as

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\text{CoT}}} \big[\texttt{err}_{\text{pre}}(\mathbb{P}, \mathbb{P}_{\widehat{\rho}}; \texttt{prompt}_{\text{CoT}}(n)) \big] \\
&\quad = \mathbb{E}_{\mathbb{P}_{\text{CoT}}} \Big[\texttt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\text{test}} = \cdot \mid \texttt{prompt}_{\text{CoT}}(n)) \big) \Big] \\
&\quad \le 2(3+b^*)\kappa \cdot \pi(\theta^*)^{-1} \cdot T(H+1) \cdot \Delta_{\text{pre}}(N,T,\delta),
\end{aligned} \tag{128}$$

with probability at least $1-\delta$. The randomness comes from the pretrained model $\mathbb{P}_{\widehat{\rho}}$. This concludes **Step 1**.

Step 2: Control expected prompting error under distribution shift. In this step, we evaluate the expected prompting error $\texttt{err}_{\text{prompt}}$ (8) with the expectation taken under $\texttt{prompt}_{\text{CoT}}(n) \sim \mathbb{P}_{\text{CoT}}$. Similar to **Step 1**, we first take the expectation of the shifted

testing query $z_0^{\text{test}} \sim \mu(\cdot \mid \texttt{prompt}_{\text{CoT}}(n))$, followed by the expectation of the demonstrations $\Upsilon_n \sim \mathbb{P}(\cdot \,|\, \theta^*)$.

We first compute the expected KL divergence with respect to the query $z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)$:

$$\begin{aligned}
&\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \Big[\texttt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)) \big) \Big] \\
&\quad \le \mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\log \left(1 + \frac{\sum_{\theta \in \Theta^{\complement}} \mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta) \pi(\theta)}{\mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta^*) \pi(\theta^*)} \right) \right] \\
&\quad \le \log \left(1 + \mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\frac{\sum_{\theta \in \Theta^{\complement}} \mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta) \pi(\theta)}{\mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta^*) \pi(\theta^*)} \right] \right). \qquad (129)
\end{aligned}$$

The first inequality follows from Proposition 28, and the second is due to Jensen's inequality. Next, we rewrite the expected likelihood ratio in (129) as

$$\begin{aligned}
&\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\frac{\sum_{\theta \in \Theta^{\complement}} \mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta) \pi(\theta)}{\mathbb{P}(\Upsilon_n, z_0^{\text{test}} \,|\, \theta^*) \pi(\theta^*)} \right] \\
&\quad = \sum_{\theta \in \Theta^{\complement}} \frac{\mathbb{P}(\Upsilon_n \,|\, \theta)}{\mathbb{P}(\Upsilon_n \,|\, \theta^*)} \cdot \mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\frac{\mathbb{P}(z_0^{\text{test}} \,|\, \theta) \pi(\theta)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*) \pi(\theta^*)} \right], \qquad (130)
\end{aligned}$$

where the equality follows from the independence between Υ_n and z_0^{test} conditioning on any task θ, which enables us to exchange $\sum_{\theta \in \Theta^{\complement}}$ with the expectation with respect to $\mu(\cdot \,|\, \Upsilon_n)$. According to Lemma 29 and Assumption 6, we have $\mathbb{P}(\Upsilon_n \,|\, \theta)/\mathbb{P}(\Upsilon_n \,|\, \theta^*) \le \exp(-2n\lambda + 2\log(\bar{\xi}^{-1}))$ with probability at least $1 - \bar{\xi}$ for any $\theta \in \Theta^{\complement}$. Setting $\bar{\xi} = \xi/|\Theta|$ and taking a union bound, we therefore have

$$\text{RHS of (130)} \le \big|\Theta^{\complement}\big|^2 \xi^{-2} \cdot \exp(-2n\lambda) \cdot \sum_{\theta \in \Theta^{\complement}} \mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\frac{\mathbb{P}(z_0^{\text{test}} \,|\, \theta) \pi(\theta)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*) \pi(\theta^*)} \right], \qquad (131)$$

which holds with probability at least $1 - \xi$ with respect to the randomness comes from $\Upsilon_n \sim \mathbb{P}(\cdot \,|\, \theta^*)$.

Next we upper bound each term in the summation in (131). By changing the probability measure and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)} \left[\frac{\mathbb{P}(z_0^{\text{test}} \,|\, \theta)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*)} \right] \\
&\quad = \mathbb{E}_{z_0^{\text{test}} \sim \mathbb{P}(\cdot \,|\, \theta^*)} \left[\frac{\mu(z_0^{\text{test}} \,|\, \Upsilon_n) \cdot \mathbb{P}(z_0^{\text{test}} \,|\, \theta)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*) \cdot \mathbb{P}(z_0^{\text{test}} \,|\, \theta^*)} \right] \\
&\quad \le \sqrt{\mathbb{E}_{z_0^{\text{test}} \sim \mathbb{P}(\cdot \,|\, \theta^*)} \left[\left(\frac{\mathbb{P}(z_0^{\text{test}} \,|\, \theta)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*)} \right)^2 \right]} \sqrt{\mathbb{E}_{z_0^{\text{test}} \sim \mathbb{P}(\cdot \,|\, \theta^*)} \left[\left(\frac{\mu(z_0^{\text{test}} \,|\, \Upsilon_n)}{\mathbb{P}(z_0^{\text{test}} \,|\, \theta^*)} \right)^2 \right]} \\
&\quad = \sqrt{\big(\chi^2\big(\mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta), \mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta^*) \big) + 1 \big) \cdot \kappa^2}, \qquad (132)
\end{aligned}$$

where the second line follows from the Cauchy–Shwarz inequality. The final line follows because

$$\mathbb{E}_{z_0^{\text{test}} \sim \mathbb{P}(\cdot \mid \theta^*)} \left[\left(\frac{\mu(z_0^{\text{test}} \mid \Upsilon_n)}{\mathbb{P}(z_0^{\text{test}} \mid \theta^*)} \right)^2 \right] \le \kappa^2,$$

which results from Assumption 17. Besides, we define $C(\theta^*)$ as

$$C(\theta^*) = \sup_{\theta \in \Theta^{\complement}} \sqrt{\chi^2\big(\mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta), \mathbb{P}(z_0^{\text{test}} = \cdot \,|\, \theta^*)\big) + 1}. \tag{133}$$

Thus, combining (131), (132), (133), with probability at least $1 - \xi$, we have

$$\text{RHS of (131)} \le C(n) \cdot \xi^{-2}, \text{ where } C(n) = \frac{\pi(\Theta^{\complement})}{\pi(\theta^*)} \cdot \kappa \big|\Theta^{\complement}\big|^2 \cdot C(\theta^*) \cdot \exp(-2n\lambda), \tag{134}$$

and $C(\theta^*)$ is defined in (133).

Applying (134) to (129) gives us the following tail probability bound:

$$\mathbb{P}\bigg(\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n))\big)\Big] > \log\big(1 + C(n)\xi^{-2}\big)\bigg) < \xi, \tag{135}$$

where $\xi \in (0,1)$, and the randomness comes from $\Upsilon_n \sim \mathbb{P}(\cdot \,|\, \theta^*)$. By replacing $x = \log(1 + C(n) \cdot \xi^{-2})$ in (135), for any $x \in [0, \infty)$, we have

$$\begin{aligned}
&\mathbb{P}\bigg(\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n))\big)\Big] > x\bigg) \\
&\quad \le \big(C(n)/(\exp(x) - 1)\big)^{1/2}.
\end{aligned} \tag{136}$$

We provide an upper bound of the expected KL divergence by integrating the tail probability in (136) as follows,

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\text{CoT}}}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n))\big)\Big] \\
&= \int_0^{\infty} \mathbb{P}\bigg(\mathbb{E}_{z_0^{\text{test}} \sim \mu(\cdot \,|\, \Upsilon_n)}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n))\big)\Big] > x\bigg) \mathrm{d}x.
\end{aligned}$$

Note that we can split the integration of x over $[0, \infty)$ into two regions: $[0, \log(1 + C(n))$ and $[\log(1 + C(n), \infty)$, where the probability in (136) is bounded by one in $[0, \log(1 + C(n))$. Therefore, we have

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\text{CoT}}}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n))\big)\Big] \\
&\quad \le \log\big(1 + C(n)\big) + \int_{\log(1 + C(n))}^{\infty} \big(C(n)/(\exp(x) - 1)\big)^{1/2} \mathrm{d}x \\
&\quad = \log\big(1 + C(n)\big) + C(n)^{1/2} \cdot \Big(\pi - 2\arctan\big(C(n)^{1/2}\big)\Big) = \mathcal{O}\big(C(n)^{1/2}\big) \\
&\quad = \mathcal{O}\bigg(\bigg(\frac{\pi(\Theta^{\complement})}{\pi(\theta^*)} \cdot C(\theta^*) \cdot \kappa\bigg)^{1/2} \cdot \big|\Theta^{\complement}\big| \cdot \exp(-n\lambda)\bigg).
\end{aligned} \tag{137}$$

Here in the third line we plug in the closed-form

$$\int \big(\exp(x) - 1\big)^{-1/2} \mathrm{d}x = 2\arctan\big(\sqrt{\exp(x) - 1}\big).$$

When n is large, $C(n)$ is sufficiently small. The second equality follows from the first-order Taylor approximations $\log(1+u) \approx u$ and $\arctan(u) \approx u$ when u is close to zero.

The convergence rate in the last line is dominated by the rate of $C(n)^{1/2}$. Therefore, we control the rate of expected prompting error defined in (8) by applying (137):

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\mathrm{CoT}}}\Big[\mathtt{err}_{\mathrm{prompt}}\big(\mathbb{P}, \theta^*, \mathtt{prompt}_{\mathrm{CoT}}(n)\big)\Big] \\
&\quad = \mathcal{O}\bigg(Hb^*\bigg(\frac{\pi(\Theta^{\complement})}{\pi(\theta^*)}C(\theta^*)\kappa\bigg)^{1/4} \cdot \big|\Theta^{\complement}\big|^{1/2} \cdot \exp(-n\lambda/2)\bigg). \qquad (138)
\end{aligned}$$

Combining (128) and (138), we have that under the Assumptions 6, 17, with probability at least $1-\delta$,

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}_{\mathrm{CoT}}}\Big[\mathtt{KL}\big(\mathbb{P}(y^{\mathrm{test}} = \cdot \,|\, z_0^{\mathrm{test}}, \theta^*), \mathbb{P}_{\widehat{\rho}}(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n))\big)\Big] \\
&\quad = \mathcal{O}\bigg(Hb^*\bigg(\frac{\pi(\Theta^{\complement})}{\pi(\theta^*)} \cdot C(\theta^*) \cdot \kappa\bigg)^{1/4} \cdot \big|\Theta^{\complement}\big|^{1/2} \cdot \exp(-n\lambda/2) \\
&\qquad + \kappa T H \pi(\theta^*)^{-1}(1+b^*) \cdot \Delta_{\mathrm{pre}}(N, T, \delta)\bigg),
\end{aligned}$$

where the first term corresponds to the prompting error $\mathtt{err}_{\mathrm{prompt}}$ (8), and the second corresponds to the pre-training error $\mathtt{err}_{\mathrm{pre}}$ (9). The randomness comes from the pretrained model $\mathbb{P}_{\widehat{\rho}}$. Therefore, we conclude the proof. ■

Appendix I. Supplemental Materials

This section consists of three subsections. The first Subsection I.1 gives a more detailed description of the pretraining process. The Subsections I.2 and I.3 prove the lemmas and propositions used in Sections E and H.

I.1 Additional Details about Pretraining

In this section, we provide a detailed description of the pretraining process of autoregressive LLM. Specifically, we focus on pretraining with data sampled from the generalized model described in (20). An autoregressive LLM is a transformer that maps a reasoning step sequence $S \in \mathcal{L}^*$ to a probability distribution for predicting the next reasoning step $z \in \mathcal{L}$.

Transformer Architecture. We focus on a transformer with D transformer blocks stacked sequentially followed by a final softmax layer. Let $X^0 \in \mathbb{R}^{L\times r}$ denote the initial input embedding for the entire network, which contains both the content embedding and the positional encoding. The d-th block takes in $X^{d-1} \in \mathbb{R}^{L\times r}$, produces $X^d \in \mathbb{R}^{L\times r}$, and feeds it to the next module until arriving at the last one. Each transformer block consists of four components: a MHA and a FF layer. Each component has a residual connection around it, followed by layer normalization, which prepares the raw output of the current layer to be forwarded as input toward the next layer. See Figure 4 for an illustration of the architecture.

Input Embedding. Specifically, the input of the transformer is a sequence of reasoning steps of length L, with each step taking values in $\mathcal{L}$. Since the attention mechanism is permutation invariant but the sequential order matters in CoT reasoning, to encode an order, the transformer incorporates positional embeddings that map the positional information of each reasoning step into the Euclidean space. In addition, the values in $\mathcal{L}$ are also mapped to a vector space. Thus, the transformer first maps the input sequence of length L into a sequence of L vectors in $\mathbb{R}^r$, which involves both content and positional embedding.

Multi-Head Attention (MHA). We let $X^0 \in \mathbb{R}^{L\times r}$ denote the output after the embedding module, which is passed to D transformer blocks. Each block consists of a MHA layer, a FF layer, and two normalization layers. For any $d \in [D]$, the parameters of the d-th transformer block are $\rho^d = (W^d_{\text{mha}}, W^d_{\text{ff}}, \gamma^d_1, \gamma^d_2)$. The weight matrices of the MHA layer are $W^d_{\text{mha}} = (W_i^{Q,d}, W_i^{K,d}, W_i^{V,d})_{i=1}^{\eta}$, where η is the number of heads. Here, $W_i^{Q,d} \in \mathbb{R}^{r\times d_q}$, $W_i^{K,d} \in \mathbb{R}^{r\times d_k}$, and $W_i^{V,d} \in \mathbb{R}^{r\times d_v}$ convert the input X^{d-1} into queries, keys, and values, respectively. We set $d_v = r$ to ensure the MHA output is also in $\mathbb{R}^r$. Specifically, a MHA layer with η heads output a vector sequence given by (18). We denote the output of the d-th attention layer by $\overline{X}^d = \texttt{mha}(X^{d-1}, W^d_{\text{mha}}) \in \mathbb{R}^{L\times r}$. In particular, for any $\ell \in [L]$, the ℓ-th vector of $\overline{X}^d$ is given by

$$\overline{X}^d[\ell] = \sum_{i=1}^{\eta} \texttt{attn}(q_i^\ell, K_i, V_i), \tag{139}$$

where the query, key, and value of the i-th head are $q_i^\ell = (W_i^{Q,d})^\top X^{d-1}[\ell]$, $K_i = X^{d-1}W_i^{K,d}$, and $V = X^{d-1}W_i^{V,d}$. Here `attn` in (139) is the softmax attention defined in (17).

First Residual Link and Normalization. The raw output of MHA layer is then passed through a residual link with diagonal weight matrix $\gamma_1^d \in \mathbb{R}^{r\times r}$ and a normalization layer $\texttt{NL}(\cdot)$, resulting in the intermediate output

$$Y^d = \texttt{NL}(\overline{X}^d + X^{d-1}\gamma_1^d) = \texttt{NL}\big(\texttt{mha}(X^{d-1}, W_{\text{mha}}^d) + X^{d-1}\gamma_1^d\big) \in \mathbb{R}^{L\times r}.$$

Here the multiplication of γ_1^d should be understood as a columnwise operation for all $r' \in [r]$. Note that each of the r vector of X^{d-1} is in $\mathbb{R}^L$, which is mapped to another vector in $\mathbb{R}^L$ by scaling with $\gamma_1^d[r']$. For the ease of analysis, we adopt the normalization function that maps each row of the input into the unit ℓ_2-ball as follows.

$$[\texttt{NL}(X)]_{i,:} = \begin{cases} X_{i,:} & \text{if } \|X_{i,:}\|_2 \le 1 \\ X_{i,:}/\|X_{i,:}\|_2 & \text{otherwise.} \end{cases} \tag{140}$$

Another popular normalization function is layer normalization (Xiong et al., 2020), which standardizes the vectors of $\overline{X}^d + X^{d-1}\gamma_1^d$ by subtracting the mean and dividing by the variance.

Feed Forward (FF) Layer. The FF layer is parameterized by $W_{\text{ff}}^d = (W_{\text{ff},1}^d, W_{\text{ff},2}^d)$, where $W_{\text{ff},1}^d \in \mathbb{R}^{r\times d_F}$ and $W_{\text{ff},2}^d \in \mathbb{R}^{d_F\times r}$, where d_F is the number of neurons of the FF layer. In particular, Y^d is passed through the FF layer and the output is another sequence of vectors in $\mathbb{R}^{L\times d}$. To get the output, for any $\ell \in [L]$, we pass $Y^d[\ell]$ into a two layer neural network and obtain $\texttt{ReLU}\big((Y^d[\ell])^\top (W_{\text{ff},1}^d)\big)W_{\text{ff},2}^d$, where $\texttt{ReLU}(\cdot)$ is the ReLU activation function and we regard $Y^d[\ell]$ as a column vector in $\mathbb{R}^r$. Here we omit the intercepts to simplify the presentation. The output of FF layer, denoted by $\texttt{ff}(Y^d, W_{\text{ff}}^d)$, concatenates all these L output vectors.

Second Residual Link and Normalization. The output of FF layer is then passed through a second residual link with weight $\gamma_2^d \in \mathbb{R}^{r\times r}$ and a normalization layer $\texttt{NL}(\cdot)$. This concludes the d-th transformer block and the resulting output is given by

$$X^d = \texttt{NL}\big(\texttt{ff}(Y^d, W_{\text{ff}}^d) + Y^d\gamma_2^d\big) \in \mathbb{R}^{L\times r}.$$

Softmax Output Layer After processing through all D transformer blocks, the output X^D is fed into a softmax layer to generate the probability distribution of the next reasoning step. This softmax layer is parameterized by $\rho^{D+1} = (W_{\text{softmax}}, \tau)$, where $0 < \tau \le 1$ is the temperature parameter and $W_{\text{softmax}} \in \mathbb{R}^{r\times|\mathcal{L}|}$ is the weight matrix. The softmax layer takes $X^D \in \mathbb{R}^{L\times r}$ and produces the output distribution

$$O^{D+1} = \texttt{softmax}\big(\mathbf{1}^\top X^D W_{\text{softmax}}/(L\cdot\tau)\big) \in \mathbb{R}^{|\mathcal{L}|}, \tag{141}$$

where $\mathbf{1} \in \mathbb{R}^L$ is an all-one vector. Here $\texttt{softmax}(\cdot)$ is the softmax function which maps a vector in $\mathbb{R}^{|\mathcal{L}|}$ to a distribution over $\mathcal{L}$.

We concatenate all ρ^d, $d \in [D+1]$, to form ρ, which parameterizes the whole pretraining network. We assume the parameters are bounded, i.e., we consider transformers in the following parameter space:

$$\mathcal{P}_{\text{LLM}} = \Big\{\rho : \|\gamma_1^d\|_\infty, \|\gamma_2^d\|_\infty \le 1, \|W_i^{Q,d}\|_F, \|W_i^{K,d}\|_F, \|W_i^{V,d}\|_F \le B_M, \|W_{\text{ff},1}^d\|_F, \|W_{\text{ff},2}^d\|_F \le B_F,$$
$$\|W_{\text{softmax}}\|_{1,2} \le B_S, \forall d \in [D], i \in [\eta], D \ge C\log(2H), \eta \ge 1\Big\},$$

where $C > 0$ is a constant, B_M, B_F, B_S are the upper bounds on the norms. We assume these bounds to be larger than 1.

Pretraining Dataset under the Generalized Model. We describe the pretraining dataset generated according to the generalized model in (20). We denote the pretraining dataset using $\mathcal{D}_{N,T}$, which consists of N independent trajectories with T examples in each trajectory. For each trajectory $\ell \in [N]$, we first sample a task $\theta_\ell^* \overset{i.i.d}{\sim} \pi$. Conditioning on this task, we sequentially generate T examples $\{s^{k,\ell}\}_{k=1}^T$ according to the model (20), i.e., we iteratively generate the next reasoning step $z_h^{t,\ell} \sim \mathbb{P}(\cdot \,|\, S_h^{t,\ell})$, where we use $S_h^{t,\ell} = (\Upsilon_{t-1,\ell}, \{z_j^{t,\ell}\}_{j=0}^{h-1})$ to denote the sequence with all previous reasoning steps of the ℓ-th trajectory. Since LLMs make prediction autoregressively, we divide each trajectory into $T(H+1)$ pieces and collect all N independent trajectories and use $\mathcal{D}_{N,T} = \{(S_h^{t,\ell}, z_h^{t,\ell})\}_{h=0,t=1,\ell=1}^{H,T,N}$ to denote the pretraining dataset.

Maximum Likelihood Estimation (MLE). We obtain the pretrained LLM by minimizing the negative likelihood loss computed based on $\mathcal{D}_{N,T}$,

$$\widehat{\rho} = \operatorname*{argmin}_{\rho \in \mathcal{P}_{\mathrm{LLM}}} -\frac{1}{NT(H+1)} \sum_{\ell=1}^{N} \sum_{t=1}^{T} \sum_{h=0}^{H} \log \mathbb{P}_\rho\big(z_h^{t,\ell} \,|\, S_h^{t,\ell}\big) \tag{142}$$

and set $\mathbb{P}_{\mathrm{LLM}} = \mathbb{P}_{\widehat{\rho}}$. Here $\mathbb{P}_\rho$ denotes the conditional distribution specified by the transformer with parameter ρ. We neglect the optimization issue and assume that the MLE in (142) can be obtained. We note that when the transformer class is sufficiently expressive, we expect that $\mathbb{P}_{\mathrm{LLM}}$ learns the conditional distribution of $z_h^{t,\ell}$ given $S_h^{t,\ell}$.

I.2 Proofs of the Auxiliary Results in Appendix E.2

Proof of Proposition 28

Proof [Proof of Proposition 28] Using any distribution q over Θ, we bound the loglikelihood $\log \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big)$ by

$$\begin{aligned}
&\log \mathbb{P}\big(y^{\mathrm{test}} = \cdot \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \\
&\quad \geq \mathbb{E}_{\theta \sim q(\theta)} \bigg[\log \frac{\mathbb{P}\big(y^{\mathrm{test}}, \theta \,|\, \mathtt{prompt}_{\mathrm{CoT}}(n)\big)}{q(\theta)} \bigg] \\
&\quad = \mathbb{E}_q \bigg[\log \frac{\mathbb{P}\big(y^{\mathrm{test}} \,|\, \theta, \mathtt{prompt}_{\mathrm{CoT}}(n)\big) \pi(\theta) \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n) \,|\, \theta\big)}{q(\theta) \mathbb{P}\big(\mathtt{prompt}_{\mathrm{CoT}}(n)\big)} \bigg],
\end{aligned} \tag{143}$$

where we take expectation with respect to an arbitrary distribution q over Θ. The inequality follows from the evidence lower bound, and the equality follows from decomposing the numerator. Conditioning on any $\mathtt{prompt}_{\mathrm{CoT}}(n)$ consisting of n examples generated from the true distribution and a new testing input z_0^{test}, we compute the KL divergence with respect to the final output y^{test}. For simplicity, we write $\mathbb{E}_{y^{\mathrm{test}} \sim \mathbb{P}(\cdot \,|\, z_0^{\mathrm{test}}, \theta^*)}$ as $\mathbb{E}_{\theta^*}$. Then, we

have

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)\Big) \qquad (144)\\
&\quad \le \mathbb{E}_{\theta^*} \log \mathbb{P}(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta^*) - \mathbb{E}_{\theta^*}\mathbb{E}_q\left[\log \frac{\mathbb{P}\big(y^{\text{test}} \,|\, \theta, \mathtt{prompt}_{\text{CoT}}(n)\big)\pi(\theta)\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)}{q(\theta)\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n)\big)}\right]\\
&\quad = \mathbb{E}_{\theta^*}\mathbb{E}_{\theta\sim q}\left[\log \frac{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n)\big)q(\theta)}{\mathbb{P}(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta)\pi(\theta)}\right] - \mathbb{E}_{\theta^*}\mathbb{E}_{\theta\sim q}\left[\log \frac{\mathbb{P}\big(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta\big)}{\mathbb{P}(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta^*)}\right].
\end{aligned}$$

The inequality follows from the definition of KL divergence and the lower bound in (143). The equality follows from the fact that $\mathbb{P}(y^{\text{test}} \,|\, \theta, \mathtt{prompt}_{\text{CoT}}(n)) = \mathbb{P}(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta)$ under the model in (2) and rearranging terms.

Now we set q as $q(\theta) \propto \mathbf{1}\{\theta \in \Theta_{\text{eq}}(\theta^*)\} \cdot \pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)$, where $\pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)$ is the posterior distribution over Θ after observing the prompt. Note that this q assigns zero probability to any θ outside the equivalence class $\Theta_{\text{eq}}(\theta^*)$. By taking an expectation with respect to this q, we have

$$\mathbb{E}_{\theta^*}\mathbb{E}_{\theta\sim q}\big[\log \mathbb{P}(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta) - \log \mathbb{P}(y^{\text{test}} \,|\, z_0^{\text{test}}, \theta^*)\big] = 0$$

by the construction of the equivalence classes in Definition (5). Thus the KL divergence in (144) is further bounded as follows:

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}\big(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*\big), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)\Big)\\
&\quad \le \mathbb{E}_{\theta^*}\mathbb{E}_{\theta\sim q}\left[\log \frac{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n)\big)q(\theta)}{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big) \cdot \pi(\theta)}\right]\\
&\quad = \mathbb{E}_{\theta^*}\mathbb{E}_{\theta\sim q}\left[\log \frac{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n)\big) \cdot \pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)}{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)\pi(\theta) \cdot \int_{\Theta_{\text{eq}}(\theta^*)} \pi\big(\theta' \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)\mathrm{d}\theta'}\right]\\
&\quad = -\log\Big(\int_{\Theta_{\text{eq}}(\theta^*)} \pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)\mathrm{d}\theta\Big). \qquad (145)
\end{aligned}$$

Here in the first equality, we plug in the closed form of $q(\theta)$, and in the second equality, we use the fact that

$$\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n)\big) \cdot \pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big) = \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big) \cdot \pi(\theta)$$

to cancel terms. We can interpret the last term in (145) as an integrated version of posterior contraction. Intuitively, a better CoT prompt yields a higher posterior concentration on $\Theta_{\text{eq}}(\theta^*)$, leading to a smaller upper bound. Finally, we plug in the closed-form expression of $\pi(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n))$, i.e.,

$$\pi\big(\theta \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big) = \frac{\mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)\pi(\theta)}{\int_\Theta \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)\pi(\theta)\mathrm{d}\theta},$$

in (145) and obtain that

$$\begin{aligned}
&\mathtt{KL}\Big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, z_0^{\text{test}}, \theta^*), \mathbb{P}\big(y^{\text{test}} = \cdot \,|\, \mathtt{prompt}_{\text{CoT}}(n)\big)\Big) \\
&\quad \le \log \frac{\int_{\Theta} \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)\pi(\theta)\mathrm{d}\theta}{\int_{\Theta_{\text{eq}}(\theta^*)} \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,|\, \theta\big)\pi(\theta)\mathrm{d}\theta} \\
&\quad = \log\left(1 + \frac{\int_{\Theta^{\complement}} \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)\mathrm{d}\theta}{\int_{\Theta_{\text{eq}}(\theta^*)} \mathbb{P}\big(\mathtt{prompt}_{\text{CoT}}(n) \,\big|\, \theta\big)\pi(\theta)\mathrm{d}\theta}\right).
\end{aligned}$$

Here the equality follows from the definition of $\Theta^{\complement}$. Therefore, we conclude the proof. ■

Proof of Lemma 29

Proof [Proof of Lemma 29] Recall that we define $J_i \subseteq [H-1]$ for each $i \in [n]$, and we use $S^i_{J_i}$ to denote a truncated version of the i-th trajectory S^i corresponding to the indices specified by J_i. Namely, $S^i_{J_i} = \{z^i_0\} \cup \{z_j\}_{j \in J_i} \cup \{z^i_H\}$. We begin by applying the previous Lemma 50:

$$\begin{aligned}
\frac{1}{2}\log\frac{\mathbb{P}(\{S^i_{J_i}\}_{i=1}^n \,|\, \theta)}{\mathbb{P}(\{S^i_{J_i}\}_{i=1}^n \,|\, \theta^*)} &= \sum_{i=1}^n \frac{1}{2}\log\frac{\mathbb{P}(S^i_{J_i} \,|\, \theta)}{\mathbb{P}(S^i_{J_i} \,|\, \theta^*)} \\
&\le \sum_{i=1}^n \log\left[\mathbb{E}_{\theta^*}\left(\frac{\mathbb{P}(S_{J_i} \,|\, \theta)}{\mathbb{P}(S_{J_i} \,|\, \theta^*)}\right)^{\frac{1}{2}}\right] + \log(\delta^{-1}) \\
&\le \sum_{i=1}^n \left[\mathbb{E}_{\theta^*}\left(\frac{\mathbb{P}(S_{J_i} \,|\, \theta)}{\mathbb{P}(S_{J_i} \,|\, \theta^*)}\right)^{\frac{1}{2}} - 1\right] + \log(\delta^{-1}) \\
&= -\sum_{i=1}^n \mathrm{H}^2\big(\mathbb{P}(S_{J_i} \,|\, \theta^*), \mathbb{P}(S_{J_i} \,|\, \theta)\big) + \log(\delta^{-1}),
\end{aligned}$$

with probability at least $1-\delta$. The first inequality follows from Lemma 50, and the second inequality follows from the fact that $x - 1 \ge \log(x)$. Putting both sides of the inequality into the exponential function, we have

$$\frac{\mathbb{P}(\{S^i_{J_i}\}_{i=1}^n \,|\, \theta)}{\mathbb{P}(\{S^i_{J_i}\}_{i=1}^n \,|\, \theta^*)} \le \exp\Big(-2\sum_{i=1}^n \mathrm{H}^2\big(\mathbb{P}(S_{J_i} \,|\, \theta^*), \mathbb{P}(S_{J_i} \,|\, \theta)\big) + 2\log(\delta^{-1})\Big),$$

with probability at least $1-\delta$. Therefore, we conclude the proof. ■

I.3 Proofs of the Auxiliary Lemmas in Appendix H.1

Proof of Lemma 34

Proof

We first note that for any $z \in \mathcal{L}$ and $S \in \mathcal{L}^*$, we have

$$\mathbb{P}_{\widehat{\rho}}(z \mid S) \geq 1/\big(1 + |\mathcal{L}| \exp(B_S/\tau)\big), \tag{146}$$

which follows from the softmax layer in (141). Combining (146) with Assumption 13, we obtain the upper bound of the following log density bound:

$$\Big|\log \mathbb{P}(z \mid S) - \log \mathbb{P}_{\widehat{\rho}}(z \mid S)\Big| \leq b^* = \log \max\{c_0^{-1}, 1 + |\mathcal{L}| \exp(B_S/\tau)\}. \tag{147}$$

Inequality (147) gives the specific form of the upper bound b^* mentioned in Assumption 3.

Next, we apply concentration to each fixed (t, h) with $t \in [T]$ and $0 \leq h \leq H$. By Hoeffding's inequality and (147) we have

$$\mathbb{P}\bigg(\sum_{\ell=1}^{N}\bigg(\log \frac{\mathbb{P}(z_h^{t,\ell} \mid S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell} \mid S_h^{t,\ell})} - \mathbb{E}_{S_h^{t,\ell}} \mathtt{KL}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_{\rho'}(\cdot \mid S_h^{t,\ell})\big)\bigg) > t\bigg) \leq 2\exp\bigg(\frac{-t^2}{2N(b^*)^2}\bigg). \tag{148}$$

Then (148) implies that, with probability at least $1 - \delta$, we have

$$\frac{1}{N}\sum_{\ell=1}^{N}\bigg(\log \frac{\mathbb{P}(z_h^{t,\ell} \mid S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell} \mid S_h^{t,\ell})} - \mathbb{E}_{S_h^{t,\ell}} \mathtt{KL}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_{\rho'}(\cdot \mid S_h^{t,\ell})\big)\bigg) \leq b^* \sqrt{\frac{1}{2N}} \log \frac{1}{\delta}. \tag{149}$$

Applying union bound to (149) for all (t, h) with $t \in [T], 0 \leq h \leq H$, we have that

$$\begin{aligned}
&\frac{1}{NT(H+1)} \cdot \sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\bigg(\log \frac{\mathbb{P}(z_h^{t,\ell} \mid S_h^{t,\ell})}{\mathbb{P}_{\rho'}(z_h^{t,\ell} \mid S_h^{t,\ell})} - \mathbb{E}_{S_h^{t,\ell}} \mathtt{KL}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_{\rho'}(\cdot \mid S_h^{t,\ell})\big)\bigg) \\
&\quad \leq b^* \sqrt{\frac{1}{2N}} \log \frac{T(H+1)}{\delta}.
\end{aligned}$$

Therefore, we conclude the proof of this lemma. ■

Proof of Lemma 36

Proof Fix any (t, h) with $t \in [T]$ and $0 \leq h \leq H$, we invoke Proposition 53 by setting $X_\ell = S_h^{t,\ell}$ and $f(S_h^{t,\ell}) = \mathtt{TV}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_\rho(\cdot \mid S_h^{t,\ell})\big)$, which gives us that with probability at least $1 - \delta$,

$$\begin{aligned}
&\frac{1}{N}\Big|\mathbb{E}_{\rho \sim P}\sum_{\ell=1}^{N}\Big[\mathbb{E}_{S_h^{t,\ell}}\big[\mathtt{TV}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_\rho(\cdot \mid S_h^{t,\ell})\big)\big] - \mathtt{TV}\big(\mathbb{P}(\cdot \mid S_h^{t,\ell}), \mathbb{P}_\rho(\cdot \mid S_h^{t,\ell})\big)\Big]\Big| \\
&\quad \leq \sqrt{\frac{1}{2\log 2 \cdot N}}\bigg[\mathtt{KL}(P \,\|\, Q) + \log \frac{4}{\delta}\bigg],
\end{aligned} \tag{150}$$

where P refers to the distribution defined in Lemma 33, and Q is the uniform distribution over $\mathcal{P}_{\mathrm{LLM}}$. The right-hand side of (150) follows from Proposition 53 by setting $b = 1$ because the TV distance is always between 0 and 1.

Next, we note that by the construction of P in (87), for $\rho \sim P$, we have

$$\begin{aligned}\texttt{TV}\big(\mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^{t,\ell}), \mathbb{P}_{\rho}(\cdot \,|\, S_h^{t,\ell})\big) &\le \sqrt{\frac{1}{2}\texttt{KL}\big(\mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^{t,\ell}), \mathbb{P}_{\rho}(\cdot \,|\, S_h^{t,\ell})\big)} \\ &= \mathcal{O}\big(1/\sqrt{NTH}\big), \end{aligned} \tag{151}$$

for any S_h^t. The first line follows from the Pinsker's inequality, and the second line follows directly from (88).

Combining Lemma 35, (150) and (151) with union bound across all $(t, h), t \in [T], 0 \le h \le H$, we obtain

$$\begin{aligned}&\frac{1}{NT(H+1)}\sum_{\ell=1}^{N}\sum_{t=1}^{T}\sum_{h=0}^{H}\Big(\mathbb{E}_{S_h^{t,\ell}}\Big[\texttt{TV}\big(\mathbb{P}(\cdot \,|\, S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^{t,\ell})\big)\Big] - \texttt{TV}\big(\mathbb{P}(\cdot \,|\, S_h^{t,\ell}), \mathbb{P}_{\widehat{\rho}}(\cdot \,|\, S_h^{t,\ell})\big)\Big) \\ &\quad = \mathcal{O}\bigg(\frac{1}{\sqrt{N}}\Big(\bar{D}\log(1 + NTH\bar{B}) + \log\frac{TH}{\delta}\Big)\bigg).\end{aligned}$$

Therefore, we conclude the proof. ■

I.4 Proofs of the Auxiliary Lemmas in Appendix H.2

I.4.1 Proof of Proposition 39

Proof In this proof, we aim to show that a sequence of L transformer blocks defined in (95) can exactly represent the MLP up to a scaling. Recall that the intermediate layers of the MLP are denoted by $\{X^d\}_{d\in[L]}$. In the following, we let $Y^0 = X^0$, and let Y^d denote the output of the d-th transformer block for all $d \in [L]$. Our goal is to construct the transformer blocks such that Y^L is exactly equal to X^L up to a constant factor.

Our construction is based on two key ideas. First, as shown in Appendix H.3, in each transformer block, we can set the MHA layer to a zero function and only keep the FF layer and normalization layer in (95). Second, to avoid the influence of the normalization layers, we adopt the scaling trick by scaling the parameters of the FF layer to ensure the output matrix is bounded by one in terms of the row-wise ℓ_2-norm. This normalizing scalar is then multiplied by the weight matrix of the next layer to ensure the output stays the same.

More rigorously, we define $B_0 = B$, which is an upper bound of $\|(X^0)^\top\|_{2,\infty}$. Recall that we assume the intermediate values of the MLP satisfy $\|(X^d)^\top\|_{2,\infty} \le B_d$ for all $d \in [L]$, where $\|(X^d)^\top\|_{2,\infty}$ is the row-wise maximum ℓ_2-norm of X^d. We will construct a transformer such that $Y^d = X^d/B_d$ for all $d \in [L]$. We prove this argument via induction.

We will verify the base case $Y^1 = X^1/B_1$ later. For any $d \ge 2$, suppose we have $Y^{d-1} = X^{d-1}/B_{d-1}$ and let us consider the d-th transformer block and the d-th layer of the MLP. Note that X^d is constructed by X^{d-1} via (98) and Y^d is derived from Y^{d-1} via

$$Y^d = \texttt{NL}\big(\texttt{ff}(Z, \overline{W}_{\text{ff}}, \bar{b}_{\text{ff}}) + Z^d\bar{\gamma}_2\big), \qquad Z^d = \texttt{NL}\big(\texttt{mha}(Y^{d-1}, \overline{W}_{\text{mha}}) + Y^{d-1}\bar{\gamma}_1\big) \tag{152}$$

for some weight matrices W_{ff}, $\bar{b}_{\text{ff}}$, $\overline{W}_{\text{mha}}$, $\bar{\gamma}_1$, and $\bar{\gamma}_2$ to be determined. We set the number of heads of the MHA layer to be $\eta = 1$ and set the weight matrix of the values, $\overline{W}^V$, as a

zero matrix. Moreover, we set $\overline{\gamma}_1 = I$ in (152). Thus, we have

$$Z^d = \mathtt{NL}\big(\mathtt{mha}(Y^{d-1}, \overline{W}_{\mathrm{mha}}) + Y^{d-1}\overline{\gamma}_1\big) = \mathtt{NL}(X^{d-1}/B_{d-1}) = X^{d-1}/B_{d-1}. \tag{153}$$

Here the second equality follows from the induction assumption and the last equality follows from the fact that the ℓ_2-norm of each row of X^{d-1}/B_{d-1} is bounded by one.

Now we set

$$\overline{W}_{\mathrm{ff}} = \{W^d_{\mathrm{ff}} \cdot B_{d-1}, I/B_d\}, \qquad \overline{b}^d_{\mathrm{ff}} = \{b^d_{\mathrm{ff}}, 0\}, \qquad \text{and} \qquad \overline{\gamma}_1 = I$$

in (152). That is, $\overline{W}_{\mathrm{ff},1}$ is proportional to W^d_{ff} of the MLP, $\overline{W}_{\mathrm{ff},2}$ is proportional to an identity matrix, $\overline{b}_{\mathrm{ff},1}$ is the same as b^d_{ff} of the MLP, $\overline{b}_{\mathrm{ff},2}$ is a zero vector, and $\overline{\gamma}_1$ as an identity matrix. Then, by direct calculation we have

$$\mathtt{ff}(Z^d, \overline{W}_{\mathrm{ff}}, \overline{b}_{\mathrm{ff}}) = \mathtt{ReLU}(X^{d-1}W^d_{\mathrm{ff}} + \mathbf{1}^\top b^d_{\mathrm{ff}})/B_d = X^d/B_d.$$

As a result, in (152) we have

$$Y^d = \mathtt{NL}\big(\mathtt{ff}(X^{d-1}/B_{d-1}, \overline{W}_{\mathrm{ff}}, \overline{b}_{\mathrm{ff}})\big) = \mathtt{NL}(X^d/B_d) = X^d/B_d, \tag{154}$$

where the first equality follows from (153) and the last equality follows from the fact that $B_d \geq \|(X^d)^\top\|_{2,\infty}$. Also see Figure 16 for an illustration of each transformer block.

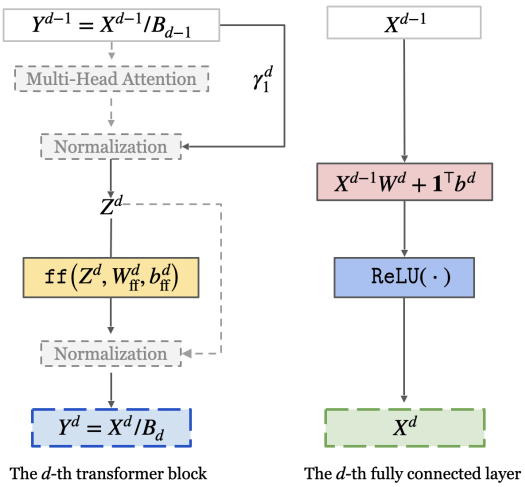


Figure 16: An illustration of the d-th transformer block, compared with the d-th fully connected layer. (a) The left figure shows the d-th transformer block, where the MHA block is omitted by setting the value weight matrix to zero. Since the input is scaled down before each normalization layer, these two layers act as identity maps. As a result, the only "active" component in this transformer block is the feed-forward layer. (b) The right figure depicts the d-th fully connected layer. The output of the d-th transformer block matches the output of this fully connected layer up to a scaling factor B_d.

It remains to verify the base case $Y^1 = X^1/B_1$. Recall that the transformer and the MLP have the same input. Thus we have $Y^0 = X^0$. For $d = 1$, in (152) we set the value matrix of MHA to be zero and set $\overline{\gamma}_1 = I/B_0$. Since each row of X^0/B_0 is in the unit ball with respect to the ℓ_2-norm, we have

$$Z^1 = \texttt{NL}(Y^0\overline{\gamma}_1) = \texttt{NL}(X^0/B_0) = X^0/B_0.$$

Thus we recover (153) for $d = 1$. Then, similar to the derivations above, we can obtain (154) for the base case. Therefore we conclude that $Y^d = X^d/B_d$ for all $d \in [L]$.

In summary, we construct a transformer with L transformer blocks such that the final output Y^L satisfies $Y^L = X^L/B_L$. The weight matrices of these transformer blocks are given by $\{\overline{W}_1^{Q,d}, \overline{W}_1^{K,d}, \overline{W}_1^{V,d}, \overline{W}_{\text{ff},1}^d, \overline{W}_{\text{ff},2}^d, \overline{b}_{\text{ff},1}^d, \overline{b}_{\text{ff},2}^d, \overline{\gamma}_1^d, \overline{\gamma}_2^d, \}_{d=1}^L$, where

$$\overline{W}_{\text{ff},1}^d = B_{d-1} \cdot W_{\text{ff}}^d, \quad \overline{W}_{\text{ff},2}^d = I/B_d, \quad \overline{b}_{\text{ff},1}^d = b_{\text{ff}}^d, \quad b_{\text{ff},2}^d = 0, \quad \overline{W}_{\text{mha}}^{V,d} = 0, \quad \overline{\gamma}_2^d = 0$$

for all $d \in [L]$. Moreover, we have $\overline{\gamma}_1^1 = I/B_0$ and $\overline{\gamma}_1^d = I$ for all $d \geq 2$. Finally, to recover X^L, it suffices to multiply $B_L \cdot I$ to Y^L. Therefore, we conclude the proof of this proposition. ■

I.4.2 Proof of Lemma 40

Proof Consider the FF layer defined in (96). We set the bias terms as $b_{\text{ff},1}^{\texttt{NN},(1)} = (0, 0, 1, -1, 0, 0)$ and $b_{\text{ff},2}^{\texttt{NN},(1)} = \mathbf{0}$. We set the weight matrices as $W_{\text{ff},1}^{\texttt{NN},(1)}$ and $W_{\text{ff},2}^{\texttt{NN},(1)}$ as

$$W_{\text{ff},1}^{\texttt{NN},(1)} = \big(W_{\text{ff},2}^{\texttt{NN},(1)}\big)^\top = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix} \in \mathbb{R}^{3\times 6}.$$

Note that fact that $x = \texttt{ReLU}(x) - \texttt{ReLU}(-x)$ for any $x \in \mathbb{R}$. For any vector of the form $v = (a, b, 0)^\top$ in $\mathbb{R}^3$, by the direct computation we have

$$(W_{\text{ff},2}^{\texttt{NN},(1)})^\top \texttt{ReLU}\big((W_{\text{ff},1}^{\texttt{NN},(1)})^\top v + (b_{\text{ff},1}^{\texttt{NN},(1)})^\top\big) = (a, b + 1, 0)^\top.$$

Thus, we have

$$X_{\texttt{NN}}^{(1)} = \texttt{ff}(X_{\texttt{NN}}^{(0)}, W_{\text{ff}}^{\texttt{NN},(1)}, b_{\text{ff}}^{\texttt{NN},(1)}) = \texttt{ReLU}(X_{\texttt{NN}}^{(0)} W_{\text{ff},1}^{\texttt{NN},(1)} + \mathbf{1}^T b_{\text{ff},1}^{\texttt{NN},(1)}) W_{\text{ff},2}^{\texttt{NN},(1)},$$

which means that $\texttt{NN}_{\texttt{J},1}$ is realized by a FF layer. Here $\mathbf{1} \in \mathbb{R}^{L'}$ is the all-one vector. Next, we compute the scaling factors in Proposition 39 to bypass the normalization layer in the FF layer. Note that both the input and output magnitude is bounded by $\sqrt{B^2 + H^2 + 1}$. Absorbing the scaling factor into the weight matrix, we have

$$\max\{\|W_{\text{ff},1}^{\texttt{NN},(1)}\|_F, \|W_{\text{ff},2}^{\texttt{NN},(1)}\|_F, \|b_{\text{ff},1}^{\texttt{NN},(1)}\|_2, \|b_{\text{ff},2}^{\texttt{NN},(1)}\|_2\} \leq \sqrt{B^2 + H^2 + 1} \cdot \sqrt{6}.$$

Furthermore, as introduced at the beginning of Appendix H.2, we can represent a feed-forward layer using a transformer block. Thus, we conclude the proof. ■

I.4.3 Proof of Lemma 41

Proof We first show that the product operation can be well approximated by a fully connected neural network, which is stated in the following lemma.

Lemma 47 (Product Operation as Neural Network) *Let $r \geq 1$ be an integer. There exists a constant $C_p > 0$ such that for any M and ϵ, there exists a multi-layer perceptron $f_{\text{product}} : \mathbb{R}^r \times \mathbb{R}^1 \to \mathbb{R}^r$ with D layers such that for any $x_1, \cdots, x_r, y \in [-M, M]$,*

$$\|f_{\text{product}}(x_{1:r}, y) - (x_1 y, \cdots, x_r y)\|_\infty \leq \epsilon.$$

Moreover, the depth D satisfies $D \leq C_p \log(M) + \log(1/\epsilon))$, and the maximum number of the hidden neurons is bounded by $5r$. Furthermore, the parameters $\{W^i_{\text{ff},1}, b^i\}_{i\in[D]}$ satisfies $\max\{\|W^i_{\text{ff},1}\|_\infty, \|b^i\|_\infty\} \leq 1$ for all $i \in [D-1]$, and $\|W^D_{\text{ff},1}\|_\infty \leq M^2$.

Proof See Appendix I.5.2 for details. ■

The output (101) can be achieved by applying the product operation construction from Lemma 47 with $r = 1$, ensuring that $|f_{\text{product}}(h, 1) - h| < \epsilon'$ and $|f_{\text{product}}(h' + 1, 0) - 0| < \epsilon'$ for all $(t', h') < (t, h - 1)$.

By setting $r = 1$, this lemma says that we can realize the product operation f_{product} in (101) as the output of a fully connected network with an error at most ϵ'. Moreover, the depth of f_{product} is at most $C_p(\log(H) + \log(1/\epsilon'))$ for some constant C, and the maximum number of hidden neurons is bounded by 5.

To implement this fully connected network using a transformer, we employ Proposition 39 by setting scaling factors as $B_0 = \sqrt{H^2 + 1}$ and $B_d = \sqrt{5}$ for all $d \in [D-1]$, and $B_D = H^2$. So far we have shown that we can use D transformer blocks to realize the product operation $f_{\text{product}}(x, y) \approx xy$. However, note that the target input of module $\texttt{NN}_{\text{J},2}$ is $X^{(1)}_{\text{NN}} \in \mathbb{R}^{L'\times 3}$, and the target output is $X^{(2)}_{\text{NN}} \in \mathbb{R}^{L'\times 3}$, where the product operation $f_{\text{product}}(x, y)$ only substitutes the second column of the output. Thus, we have to preserve the first and last columns of the input.

To realize the output in (101), we concatenate another MLP, denoted by $f_{1,3}$ to f_{product}. MLP $f_{1,3} \colon \mathbb{R}^{L'\times 3} \to \mathbb{R}^{L'\times 2}$ extracts the first and last columns of the input matrix. As we show in Lemma 48, such an MLP exists and $f_{1,3}$ in fact can be written as a single FF layer. The Frobenius norms of the weight matrices are bounded by 2. The intuition behind Lemma 48 is that we can write and x as $\texttt{ReLU}(x) - \texttt{ReLU}(-x)$, which enables us to preserve desired columns of the input matrix using a FF layer.

Finally, to concatenate $f_{1,3}$ with f_{product}, notice that these two MLPs might have different numbers of layers. This does not cause trouble because by Lemma 48, we can further stack FF layers on top of $f_{1,3}$ that represents identity mappings. This will enable us to write $f_{1,3}$ as an MLP that has the same depth as f_{product}. Then we can concatenate the weights of these two MLPs in a layer-wise fashion. Thus, the Frobenius norm of the weight matrices of the concatenated MLP is bounded by $\sqrt{B^2 + H^2 + 6} \cdot \sqrt{H^4 \cdot 5 + 4}$, where the first term in the multiplication comes from the scaling factors and the second term follows from the weight matrices. This concludes the proof of this lemma. ■

I.4.4 Proof of Lemma 42

Proof This linear operation can be realized without error by setting which can be realized by setting bias term $b^{\mathtt{NN},(2)} = (0,0,0,0,1,-1)$ and $b_{\mathrm{ff},2}^{\mathtt{NN},(2)} = \mathbf{0}$. We set the weight matrices as

$$W_{\mathrm{ff},1}^{\mathtt{NN},(2)} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1/2 & 1/2 \end{pmatrix},$$
$$W_{\mathrm{ff},2}^{\mathtt{NN},(2)} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}^{\top}.$$

For any vector of the form $v = (a,b,c)^{\top}$ in $\mathbb{R}^3$, by direct computation, a FF layer with parameters $\{W_{\mathrm{ff}}^{\mathtt{NN},(2)}, b_{\mathrm{ff}}^{\mathtt{NN},(2)}\}$ maps it to $(a,b,c')^{\top}$, where

$$c' = \mathtt{ReLU}(1-c/2) + \mathtt{ReLU}(-1+c/2).$$

Thus, $c' = 1$ if $c = 0$ and $c' = 1/2$ if $c = 1$. Therefore, we have

$$X_{\mathtt{NN}}^{(3)} = \mathtt{ff}(X_{\mathtt{NN}}^{(2)}, W_{\mathrm{ff}}^{\mathtt{NN},(2)}, b_{\mathrm{ff}}^{\mathtt{NN},(2)}),$$

which means that $\mathtt{NN}_{\mathrm{J},3}$ can be realized by a single FF layer. Moreover, the weight matrices and bias vectors are all bounded by $\sqrt{B^2+(H+1)^2+1}\cdot\sqrt{6}$ in terms of the Frobenius norm. Noticing that a FF layer can be represented by a single transformer block, we conclude the proof. ■

I.4.5 Proof of Lemma 43

Proof In this proof, we first construct an attention layer that takes $\widetilde{z}_{h'}^{t',(2)} \in \mathbb{R}^3$ as the input and outputs $(0, p_{h'}^{t'}, 0)$ for any index (t',h'). Here $p_{h'}^{t'} \approx h$ for any (t',h'). Next, we explain how to maintain the first coordinate of $\widetilde{z}_{h'}^{t',(2)}$, i.e., $z_{h'}^{t'}$. Finally, we show that this network module can be implemented by a single transformer block.

To construct the attention layer, we introduce an auxiliary parameter α to control the precision of $p_{h'}^{t'}$. Recall that the sequence S_h^t has length $L' = L'(t,h) = (t-1)\cdot(H+1)+h$ and its last element is z_{h-1}^t. For any (t',h'), $\widetilde{z}_{h'}^{t',(2)}$ is the $L'(t',h')$-th vector of the input $X_{\mathtt{NN}}^{(3)}$. We define the attention matrices $W_{\mathtt{mha}} = \{W^Q, W^K, W^V\}$, where $W^Q = (0,0,-\alpha)^{\top} \in \mathbb{R}^{3\times 1}, W^K = (0,0,1)^{\top} \in \mathbb{R}^{3\times 1}$ and

$$W^V = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Here $\alpha > 0$ is a parameter to be determined later. With the input $X_{\texttt{NN}}^{(3)} \in \mathbb{R}^{L' \times 3}$, the queries, keys, and values are given by

$$\begin{aligned} Q &= X_{\texttt{NN}}^{(3)} W^Q = -(\alpha, \cdots, \alpha, \alpha/2)^\top \in \mathbb{R}^{L'}, \\ K &= X_{\texttt{NN}}^{(3)} W^K = (1, \cdots, 1, 1/2)^\top \in \mathbb{R}^{L'}, \\ V &= X_{\texttt{NN}}^{(3)} W^V = \begin{pmatrix} 0 & \cdots & 0 \\ f_{\text{product}}(h'+1, 0) & \cdots & f_{\text{product}}(h, 1) \\ 0 & \cdots & 0 \end{pmatrix}^\top \in \mathbb{R}^{L' \times 3}, \end{aligned} \tag{155}$$

where we f_{product} approximates the product operation in the sense that $|f_{\text{product}}(x, y) - xy| \le \epsilon'$ for any $x, y \in [-H, H]$. Using (155), we compute that the softmax attention score $a(i, j)$ based on the i-th query and j-th key for all $i, j \in [L']$:

$$a(i,j) = \begin{cases} \dfrac{e^{-\alpha/2}}{e^{-\alpha/2} + (L'-1) \cdot e^{-\alpha}} & \text{for } i \neq L' \text{ and } j = L', \\ \dfrac{e^{-\alpha}}{e^{-\alpha/2} + (L'-1) \cdot e^{-\alpha}} & \text{for } i \neq L' \text{ and } j \neq L', \\ \dfrac{e^{-\alpha/4}}{e^{-\alpha/4} + (L'-1) \cdot e^{-\alpha/2}} & \text{for } i = L' \text{ and } j = L', \\ \dfrac{e^{-\alpha/2}}{e^{-\alpha/4} + (L'-1) \cdot e^{-\alpha/2}} & \text{for } i = L \text{ and } j \neq L'. \end{cases}$$

Using these attention scores to aggregate the value vectors, for any position (t', h'), we obtain the output of the attention layer:

$$(0, p_{h'}^{t'}, 0) = \sum_{j=1}^{L'} a\big(L'(t', h'), j\big) \cdot v_j, \tag{156}$$

where v_j is the j-th row of the value matrix V, and $L'(t', h')$ is the index of $z_{h'}^{t'}$ in sequence S_h^t. By the construction of V in (155), the first and last coordinates of the attention output are both zero, and we let $p_{h'}^{t'}$ in (156) to denote the nonzero coordinate.

It remains to show that $|p_{h'}^{t'} - h|$ is small for all (t', h'). For any $\epsilon > 2\epsilon'$, we can choose α sufficiently large such that $(L'-1) \cdot \max\{e^{-\alpha/2}, e^{-\alpha/4}\} + 2\epsilon' < \epsilon$. We separately consider the cases where $(t', h') < (t, h-1)$ and $(t', h') = (t, h-1)$ as follows. For the first case, we have

$$\begin{aligned} |p_{h'}^{t'} - h| &= \bigg| \frac{e^{-\alpha/2}}{e^{-\alpha/2} + (L'-1) \cdot e^{-\alpha}} \cdot f_{\text{product}}(h, 1) - h \\ &\qquad + \frac{e^{-\alpha}}{e^{-\alpha/2} + (L'-1) \cdot e^{-\alpha}} \sum_{(t', h') < (t, h-1)} f_{\text{product}}(h'+1, 0) \bigg| \\ &\le (L'-1) \cdot e^{-\alpha/2} + 2\epsilon' < \epsilon. \end{aligned} \tag{157}$$

To see the second inequality, we note that using the fact that $|f_{\text{product}}(h, 1) - h| \le \epsilon'$ and the fact that

$$\frac{e^{-\alpha/2}}{e^{-\alpha/2} + (L'-1) \cdot e^{-\alpha}} = \frac{1}{1 + (L'-1) \cdot e^{-\alpha/2}},$$

we have by direct computation that

$$\left|\frac{e^{-\alpha/2}}{e^{-\alpha/2}+(L'-1)\cdot e^{-\alpha}}\cdot f_{\mathrm{product}}(h,1)-h\right| \le \frac{|f_{\mathrm{product}}(h,1)-h|}{1+(L'-1)\cdot e^{-\alpha/2}}+\frac{(L'-1)\cdot e^{-\alpha/2}}{1+(L'-1)\cdot e^{-\alpha/2}},$$

which is bounded by $(L'-1)\cdot e^{-\alpha/2}+\epsilon'$. Moreover, the second summation in (157) is bounded by ϵ' because

$$\left|\frac{e^{-\alpha}}{e^{-\alpha/2}+(L'-1)\cdot e^{-\alpha}}\sum_{(t',h')<(t,h-1)} f_{\mathrm{product}}(h'+1,0)\right| \le |f_{\mathrm{product}}(h'+1,0)| \le \epsilon'.$$

Combining the above two inequalities yields (157). Similarly, for $(t',h')=(t,h-1)$, we use the same argument to obtain that

$$\begin{aligned}|p_{h-1}^{n+1}-h| = \Bigg|&\frac{e^{-\alpha/4}}{e^{-\alpha/4}+(L'-1)\cdot e^{-\alpha/2}}\cdot f_{\mathrm{product}}(h,1)-h\\ &+\frac{e^{-\alpha/2}}{e^{-\alpha/4}+(L'-1)\cdot e^{-\alpha/2}}\sum_{(t',h')<(t,h-1)} f_{\mathrm{product}}(h'+1,0)\Bigg|\\ &\le (L'-1)\cdot e^{-\alpha/4}+2\epsilon' < \epsilon.\end{aligned}$$

In conclusion, when selecting α such that $(L'-1)\cdot\max\{e^{-\alpha/2},e^{-\alpha/4}\}<\epsilon-2\epsilon'$, we have $|p_{h'}^{t'}-h|<\epsilon$ for all (t',h'). Since $L'\le T\cdot(H+1)$, it suffices to choose

$$\alpha = 8\cdot\log\big(TH/(\epsilon-2\epsilon')\big).$$

We conclude that we can construct an attention layer that takes $\widetilde{z}_{h',(2)}^{t'}\in\mathbb{R}^3$ as input and outputs $(0,p_{h'}^{t'},0)$, such that $|p_{h'}^{t'}-h|<\epsilon$ for any $(t',h')\le(t,h-1)$. Moreover, the norms of weight matrices satisfy

$$\|W^Q\|_{\mathrm{F}} = 8\log\big(TH/(\epsilon-2\epsilon')\big), \qquad \|W^K\|_{\mathrm{F}} = \|W^V\|_{\mathrm{F}} = 1.$$

Note that this single-head attention is a special of MHA layer with $\eta=1$.

Finally, to show that such a layer can be implemented by a single transformer block defined in (95), we use a residual link by setting $\gamma_1=\mathtt{diag}(1,0,0)\in\mathbb{R}^{3\times3}$ in (95). This enables us to pass along the first coordinate $z_{h'}^{t'}$. Additionally, the FF module can append zeros to the input by taking $W_{\mathrm{ff},2}^{\mathtt{NN},(1)}$ as

$$W_{\mathrm{ff},1}^{\mathtt{NN},(1)} = \begin{pmatrix}1&-1&0&0&0&0\\0&0&1&-1&0&0\\0&0&0&0&1&-1\end{pmatrix}\in\mathbb{R}^{3\times6}, \quad \big(W_{\mathrm{ff},2}^{\mathtt{NN},(1)}\big)^\top = \begin{pmatrix} W_{\mathrm{ff},1}^{\mathtt{NN},(1)}\\ \mathbf{0}\end{pmatrix}\in\mathbb{R}^{(2|\mathcal{L}|+2)\times6},$$

and $b_{\mathtt{ff},1}^{\mathtt{NN},(1)}=b_{\mathtt{ff},2}^{\mathtt{NN},(1)}=0$. As a result, for any (t',h'), the output of the transformer block is given by

$$z_{h'}^{t',(4)} = \mathtt{NL}\big((z_{h'}^{t'},p_{h'}^{h'},\mathbf{0})\big)\in\mathbb{R}^{2+2|\mathcal{L}|}.$$

Note that the ℓ_2-norm of vector $(z_{h'}^{t'}, p_{h'}^{h'}, \mathbf{0})$ is bounded by a constant because $|p_{h'}^{h'}| \leq H$ and $\mathcal{L}$ is regarded as a compact subset of $\mathbb{R}$. Thus, we can additionally apply the scaling trick introduced in Appendix H.3 to bypass the normalization layer $\texttt{NL}(\cdot)$. To implement this scaling trick, we need only to scale W^V by a constant factor $\sqrt{B^2 + (H+1)^2 + 1}$, which affects the magnitude of the transformer weight matrices by a constant factor. Now we conclude the proof. ■

I.4.6 Proof of Proposition 44

Proof

This proof is structured in three steps. In **Step 1**, we provide a high-level overview of the network $\texttt{embed}_h(\cdot)$, aiming for $\texttt{softmax}(\texttt{embed}_h(S_h^t)/\tau) \approx g_h^*(S_h^t)$, where $g_h^*(\cdot)$ is the target distribution. In **Step 2**, we provide a detailed construction of $\texttt{embed}_h(\cdot)$ by approximating ψ_h^* and $\tau \log w_{h,i}^*$ for each $i \in [|\mathcal{L}|]$. Finally in **Step 3**, we apply the $\texttt{embed}_h(\cdot)$ approximation to construct each module G_h, which takes $(S_h^t, \widehat{p}_h, F_3, F_4)$ as the input and produces $(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t), F_4)$ as the output. We show how to modify the weight matrices in $\texttt{embed}_h(\cdot)$ to construct G_h. This technique is applied repeatedly in the proof found in Appendix H.3, with similar approaches being used in related cases.

Step 1: High-level structure of each $\texttt{embed}_h(\cdot)$. The module G_h takes input $(S_h^t, \widehat{p}_h, F_3, F_4) \in \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ as the input and outputs $(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t), F_4) \in \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$. Note that ψ_h^* is a univariate function and $\omega_h^* \colon \mathbb{R} \to \mathbb{R}^{|\mathcal{L}|}$. Note that the key functionality of the module G_h is to use S_h^t to produce $\texttt{embed}_h$. For any $h \in [H]$, we want to construct networks $\widehat{w}_h^*$ and $\widehat{\psi}_h^*$ approximating functions w_h^* and ψ_h^* in (94) separately. More specifically, we want to construct a network $\widehat{g}_h^*$ such that

$$
\begin{aligned}
g_h^*(S_h^t) &= w_h^*\bigg(\frac{1}{L'} \sum_{(i,j)=(1,0)}^{(t,h-1)} \psi_h^*(z_j^i)\bigg) \\
&\approx \texttt{softmax}\bigg(\widehat{f}_h^w\bigg(\frac{1}{L'} \sum_{(i,j)=(1,0)}^{(t,h-1)} \widehat{\psi}_h^*(z_j^i)\bigg) \Big/ \tau\bigg) = \widehat{g}_h^*(S_h^t), \qquad (158)
\end{aligned}
$$

where $\sum_{(i,j)=(1,0)}^{(t,h-1)}$ means that we sum over all reasoning steps before z_h^t. Here $\widehat{f}_h^w$ in (158) approximates the function $\tau \cdot \log w_h^*$. In **Step 2**, we apply Lemma 55 to separately bound the error induced by $\widehat{f}_h^w$ and $\widehat{\psi}_h^*$ using the universal approximation property of the fully-connected networks. Finally, in **Step 3** we combine everything and construct the G_h as a composition of transformer blocks.

In the sequel, we introduced the rationale behind the construction of $\widehat{\psi}_h^*$, and $\widehat{f}_h^w$ describe how to implement them using transformer blocks.

- **Approximate ψ_h^* using an MLP.** We construct $\widehat{\psi}_h^* \colon \mathbb{R} \to \mathbb{R}$ as a fully-connected MLP with D_ψ layers and each layer has no more than 16 neurons, where D_ψ will be specified later. The construction directly follows from Lemma 55, which is a neural network approximation result established in Elbrächter et al. (2021). As shown in

Proposition 39, such a fully connected network can be regarded as a composition of D_ψ transformer blocks. In particular, as shown in the proof of Proposition 39, in each transformer block as in (95), we can set the value matrices in the MHA layers to be zero, set γ_1 as an identity matrix, and set γ_2 to a zero matrix. This reduces the transformer block to a feed-forward layer, combined with normalization. We can apply the scaling trick introduced in Appendix H.3 to bypass the normalization layer. This enables us to represent each layer of the MLP using a transformer block.

- **Realize the average module.** After having $\widehat{\phi}_h^*$, we need to compute

$$1/L' \cdot \sum_{(i,j)=(1,0)}^{(t,h-1)} \widehat{\psi}_h^*(z_j^i)$$

using a transformer block. This can be achieved by having a single-head attention layer with $W^Q = W^K = 0$, and $W^V = 1$. To see this, observe that when $W^Q = W^K = 0$, all the attention scores become $1/L'$ and thus the attention layer becomes an average.

- **Approximate w_h^* using an MLP.** Note that w_h^* takes values in $\mathbb{R}^{|\mathcal{L}|}$. We let $w_{h,i}^*$ denote its i-th entry for all $i \in [|\mathcal{L}|]$. We leverage Lemma 55 to approximate each $\tau \cdot \log w_{h,i}^*$ using a fully connected MLP $\widehat{f}_{h,i}^w$ with D_ω layers, where each layer has at most 16 neurons. Here D_ω will be specified later. Similar to $\widehat{\psi}_h$, such an MLP can be implemented by a composition of D_w transformer blocks.

Step 2: Approximate ψ_h^* and w_h^* using MLPs. In this step, we employ the universal approximation properties of fully connected networks to construct MLPs that approximate ψ_h^* and $w_{h,i}^*$ for all $i \in [|\mathcal{L}|]$. The technical tool we leverage is Lemma 55, obtained from Elbrächter et al. (2021), which shows that MLP functions can approximate sufficiently smooth functions.

Specifically, under Assumption 37, by Lemma 55, for any desired accuracy levels ϵ_ψ and ϵ_w, there exist MLPs $\widehat{\psi}_h^*$ and $\{\widehat{f}_{h,i}^w\}_{i\in[|\mathcal{L}|]}$ such that

$$\|\widehat{\psi}_h^* - \psi_h^*\|_\infty < \epsilon_\psi, \qquad \|\widehat{f}_{h,i}^w - \tau \cdot \log w_{h,i}^*\|_\infty < \epsilon_w \text{ for all } i \in [|\mathcal{L}|], \tag{159}$$

where $\widehat{\psi}_h^*$ has D_ψ layers and each $\widehat{f}_{h,i}^w$ has at most D_w layers. Here we have

$$D_\psi = 2C_d B \cdot \big(\log(1/\epsilon_\psi)\big)^2 + \log B, \qquad D_w = 2C_d B\big(\log(1/\epsilon_w)\big)^2 + \log B, \tag{160}$$

where $C_d > 0$ is an absolute constant and B is the parameter appearing in Assumption 37. Moreover, each layer has at most 16 neurons and all the neural network weights are bounded by one in magnitude, i.e., each entry of the weight matrices is bounded in $[-1, 1]$. By this construction, define an embedding vector $\texttt{embed}_h(S_h^t) \in \mathbb{R}^{|\mathcal{L}|}$ as

$$\texttt{embed}_h(S_h^t) = \widehat{f}_h^w\bigg(\frac{1}{L'}\bigg(\sum_{i=1}^{t-1}\sum_{j=0}^{H}\widehat{\psi}_h^*(z_j^i) + \sum_{j'=0}^{h-1}\widehat{\psi}_h^*(z_{j'}^t)\bigg)\bigg), \tag{161}$$

where $\widehat{f}_h^w$ denotes the vector-valued mapping whose entries are $\{\widehat{f}_{h,i}^w\}_{i\in[|\mathcal{L}|]}$.

Next, we feed $\texttt{embed}_h(S_h^t)$ into the softmax layer and obtain an estimator of $g_h^*(S_h^t)$. For any prompt S_h^t with length $L' \le L$, the ℓ_1-approximation error is bounded by

$$\begin{aligned}
&\left\|g_h^*(S_h^t) - \texttt{softmax}\big(\texttt{embed}_h(S_h^t)/\tau\big)\right\|_1 \\
&\quad\le \left\|g_h^*(S_h^t) - \texttt{softmax}\left(\widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\psi_h^*(z_j^i)\right)\right)\right\|_1 \\
&\qquad + \left\|\texttt{softmax}\left(\widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\psi_h^*(z_j^i)\right)\right) - \texttt{softmax}\left(\widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\widehat{\psi}_h^*(z_j^i)\right)\right)\right\|_1 \\
&\quad\le 2\epsilon_w + 2\cdot 256^{D_w}\cdot\epsilon_\psi, \qquad (162)
\end{aligned}$$

where 256 appears because it is the total number of parameters in each layer of $\widehat{f}_{h,i}^w$. Here, the first inequality follows from the triangle inequality. In the second inequality, we employ Lemma 54, which states that $\texttt{softmax}(\cdot)$ is Lipschitz continuous with parameter 2 in terms of the ℓ_1-ℓ_∞ norm pair. To bound the first term, we combine Lemma 54 and (159), which shows that the first term is no more than $2\epsilon_w$. To bound the second term, we note that the fact that each $f_{h,i}^i$, as a D_w-layer MLP, is a Lipschitz continuous function in terms of the ℓ_∞-norm. The Lipschitz parameter is bounded by 256^{D_w} because the vectorized ℓ_1-norm of the weight matrix in each layer is bounded by 256, which is a result of Lemma 55. Then we combine Lemma 54, Lipschitzness of $\widehat{f}_h^w$, and (159) to obtain

$$\begin{aligned}
&\left\|\texttt{softmax}\left(\widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\psi_h^*(z_j^i)\right)\right) - \texttt{softmax}\left(\widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\widehat{\psi}_h^*(z_j^i)\right)\right)\right\|_1 \\
&\quad\le 2\cdot\max_{i\in[|\mathcal{L}|]}\left|\widehat{f}_{h,i}^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\psi_h^*(z_j^i)\right) - \widehat{f}_h^w\left(\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\widehat{\psi}_h^*(z_j^i)\right)\right| \\
&\quad\le 2\cdot 256^{D_w}\cdot\left\|\frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\psi_h^*(z_j^i) - \frac{1}{L'}\sum_{(i,j)=(1,0)}^{(t,h-1)}\widehat{\psi}_h^*(z_j^i)\right\|_\infty \\
&\quad< 2\cdot 256^{D_w}\cdot\epsilon_\psi,
\end{aligned}$$

In summary, for any given ϵ_w and ϵ_ψ, there exist MLPs $\widehat{\psi}_h^*$ and $\{\widehat{f}_{h,i}^w\}_{i\in[|\mathcal{L}|]}$ such that

$$\left\|g_h^*(S_h^t) - \texttt{softmax}\big(\texttt{embed}_h(S_h^t)/\tau\big)\right\|_1 \le 2\epsilon_w + 2\cdot 256^{D_w}\cdot\epsilon_\psi.$$

These MLPs have at most D_ψ and D_w layers respectively, where D_ψ and D_w are defined in (160). In each layer, there are 16 neurons and the weights are all in $[-1, 1]$.

In conclusion, the above analysis can be extended to bound the error

$$\left\|g_{\widetilde{h}}^*(S_h^t) - \texttt{softmax}(\texttt{embed}_{\widetilde{h}}(S_h^t)/\tau)\right\|_1$$

for all $\widetilde{h} \in \{0, \ldots, H\}$ using the same upper bound. This means that $\texttt{softmax}(\texttt{embed}_{\widetilde{h}}(S_h^t)/\tau)$ serves as an estimator for $g_{\widetilde{h}}^*(\cdot)$ with uniform precision across all $\widetilde{h}$. This generalization is

possible because the analysis in (162) is based solely on the Lipschitz continuity of $\texttt{softmax}$ and the approximation errors established in (159). Therefore, the same error bounds apply when substituting any $\widetilde{h}$ for h in $g_h^*(\cdot)$ and $\texttt{embed}_h(\cdot)$, ensuring that the error analysis is valid for any $\widetilde{h} \in \{0, \dots, H\}$.

Step 3: Construct the transformer module G_h. In the previous step, we construct $\texttt{embed}_h(S_h^t)$ that approximates $g_h^*(S_h^t)$. However, the actual input of the transformer module G_h is $(S_h^t, \widehat{p}_h, F_3, F_4) \in \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$ and the expected output is $(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t), F_4) \in \mathbb{R}^{L' \times (2+2|\mathcal{L}|)}$. In this final step, we explicitly construct the transformer module G_h that preserves $S_h^t, \widehat{p}_h, F_4$ and substitute $\mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t)$ in the third column.

To achieve such a goal, we need to first show that $\texttt{embed}_h(\cdot)$ can be implemented by transformer blocks. Then we need to show that these transformer blocks can be put in a larger transformer with the desired input-output relationship. To achieve the first goal, we apply Proposition 39 separately to the approximation modules $\widehat{\psi}_h^*$ and $(\widehat{f}_{h,1}^w, \cdots, \widehat{f}_{h,|\mathcal{L}|}^w)$, and connect them with the average module that is realized by a single-head attention layer. In particular, we apply the scaling trick in Proposition 39 to bypass the normalization layers in the transformer blocks. We specify the expression of these scaling at the end of our proof.

Our next step is to adjust weight matrices in each transformer block to preserve $S_h^t, \widehat{p}_h, F_4$ and substitute $\texttt{embed}_h(S_h^t)$ in the third column. To this end, we introduce the notion of a residual ReLU module, which is an FF layer that only keeps some desired columns of the input matrix. Then we can concatenate $\texttt{embed}_h(\cdot)$ with a residual ReLU module to achieve the desired functionality.

Lemma 48 (Residual ReLU module) *Let $X \in \mathbb{R}^{m \times r}$ denote the input, and $\mathcal{J} \subseteq [r]$ denote a set of indices. Then there exists a FF layer with weight matrices $W_{\text{ff},1} \in \mathbb{R}^{r \times (2|\mathcal{J}|)}$ and $W_{\text{ff},2} \in \mathbb{R}^{(2|\mathcal{J}|) \times |\mathcal{J}|}$ such that the output matrix only keeps those columns with indices $i \in \mathcal{J}$. Specifically, we have*

$$\texttt{ReLU}(XW_{\text{ff},1})W_{\text{ff},2} = X_{:,i \in \mathcal{J}} \in \mathbb{R}^{m \times |\mathcal{J}|}.$$

Moreover, these weight matrices satisfy $\|W_{\text{ff},1}\|_{\text{F}} = \sqrt{2|\mathcal{J}|}$ and $\|W_{\text{ff},2}\|_{\text{F}} = \sqrt{2|\mathcal{J}|}$.

Proof See Appendix I.5.3 for details. ■

To show that we can fuse $\texttt{embed}_h(\cdot)$ with a residual ReLU module that preserves the submatrix $(S_h^t, \widehat{p}_h, F_4) \in \mathbb{R}^{L' \times (2+|\mathcal{L}|)}$ through each FF layer, it suffices to show that a residual ReLU module can work together with a feed-forward layer and a MHA layer. The reason is that $\texttt{embed}_h(\cdot)$ is a composition of D_ψ FF layers, an attention layer, and D_w FF layers. If each layer of $\texttt{embed}_h(\cdot)$ can be added to a larger network which keeps $(S_h^t, \widehat{p}_h, F_4) \in \mathbb{R}^{L' \times (2+|\mathcal{L}|)}$ unchanged, then we can apply this argument to all layers of $\texttt{embed}_h(\cdot)$ and obtain the desired network. Thus, in the following, we focus only on a FF layer and a MHA layer.

Notice that permutation of the columns can be achieved by a linear FF layer. It suffices to put columns corresponding to $\texttt{embed}_h(\cdot)$ to the first $|\mathcal{L}|$ columns. That is, we can study whether the transformation

$$(X, S_h^t, \widehat{p}_h, F_4) \longrightarrow (X', S_h^t, \widehat{p}_h, F_4),$$

can be achieved by a transformer block, where $X \in \mathbb{R}^{L' \times d}$ for some d, and X' is obtained by X through an FF or MHA layer. For ease of presentation, we denote $(S_h^t, \widehat{p}_h, F_4)$ by a matrix $Y \in \mathbb{R}^{L' \times d'}$ and study this problem with abstraction, where d' is the number of columns in Y. Then we consider $X' = \texttt{ff}((X,Y), W_{\text{ff}}, b_{\text{ff}})$ or $X' = \texttt{mha}((X,Y), W_{\text{mha}})$.

First, we assume $X' = \texttt{ff}((X,Y), W_{\text{ff}}, b_{\text{ff}})$, where $W_{\text{ff}} = (W_{\text{ff},1}, W_{\text{ff},2})$ and $b_{\text{ff}} = (b_{\text{ff},1}, b_{\text{ff},2})$. Using Lemma 48, we will construct weights $\overline{W}_{\text{ff}}, \overline{b}_{\text{ff}}$ such that $(X', Y) = \texttt{ff}((X,Y), \overline{W}_{\text{ff}}, \overline{b}_{\text{ff}})$. In particular, we apply Lemma 48 to (X,Y) with $\mathcal{J} = \{d+1, \ldots, d+d'\}$, where $\mathcal{J}$ refers to the column indices corresponding to Y. Then there exist weight matrices $W_{\text{ff},1}^Y$ and $W_{\text{ff},2}^Y$ such that

$$Y = \texttt{ReLU}\big((X,Y)W_{\text{ff},1}^Y\big)W_{\text{ff},2}^Y.$$

Notice that $W_{\text{ff},1}^Y$ has size $(d+d') \times (2d')$ and $W_{\text{ff},2}^Y$ has size $2d' \times d'$. Whereas $W_{\text{ff},1}$ has d rows. Now we define

$$\overline{W}_{\text{ff},1} = \begin{pmatrix} W_{\text{ff},1} & W_{\text{ff},1}^Y \end{pmatrix}, \qquad \overline{W}_{\text{ff},2} = \begin{pmatrix} W_{\text{ff},2} & 0 \\ 0 & W_{\text{ff},2}^Y \end{pmatrix}, \qquad \overline{b}_{\text{ff},1} = (b_{\text{ff},1}, \mathbf{0}), \qquad \overline{b}_{\text{ff},2} = (b_{\text{ff},2}, \mathbf{0}).$$

Here in $\overline{W}_{\text{ff},1}$ we add d' all-zero rows below $W_{\text{ff},1}$ to construct a valid matrix. As defined in (96), we can directly calculate the FF layer with parameters $\overline{W}_{\text{ff}}$ and $\overline{b}_{\text{ff}}$ and have

$$\begin{aligned}
\texttt{ff}\big((X,Y), W_{\text{ff}}, b_{\text{ff}}\big) &= \texttt{ReLU}\big((X,Y)\overline{W}_{\text{ff},1} + \mathbf{1}^\top \overline{b}_{\text{ff},1}\big)\overline{W}_{\text{ff},2} + \mathbf{1}^\top \overline{b}_{\text{ff},2} \\
&= \big(\texttt{ReLU}((X,Y)W_{\text{ff},1} + \mathbf{1}^\top b_{\text{ff},1}), \quad \texttt{ReLU}((X,Y)W_{\text{ff},1}^Y)\big)\,\overline{W}_{\text{ff},2} + \mathbf{1}^\top \overline{b}_{\text{ff},2} \\
&= (X', Y).
\end{aligned}$$

Therefore, we construct an FF layer such that we change X to X' and keep Y unchanged.

It remains to consider the case where $X' = \texttt{mha}((X,Y), W_{\text{mha}})$. We show that (X', Y) can be implemented by a transformer block starting from (X,Y). When $X' = \texttt{mha}(X, W_{\text{mha}})$, we can augment the three matrices of W_{mha} by adding zeros such that $(X', \mathbf{0}) = \texttt{mha}((X,Y), \overline{W}_{\text{mha}})$, where $\overline{W}_{\text{mha}}$ is obtained from W_{mha} by adding zeros, and $\mathbf{0}$ is a zero matrix that has the shape as Y. Then, with a generalized residual link, we have

$$(X', Y) = \texttt{mha}((X,Y), \overline{W}_{\text{mha}}) + (X', Y)\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix},$$

where the 2×2 block matrix plays the same role as γ_1 in (95).

Therefore, we conclude that an FF and MHA layer that maps X to X' can be augmented to a layer that maps (X,Y) to (X', Y). Now we apply this argument recursively for $\texttt{embed}_h(\cdot)$. The input matrix is $(F_3, S_h^t, \widehat{p}_h, F_4)$ and the desired output is $(\mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t), S_h^t, \widehat{p}_h, F_4)$. In particular, $\texttt{embed}_h(S_h^t)$ is an MLP of S_h^t that consists of $D_\psi + D_w$ FF layers in total and a MHA layer. Thus, we can apply the above argument with $X = F_3$, $Y = (S_h^t, \widehat{p}_h, F_4)$ and X' being the intermediate outputs of $\texttt{embed}_h(\cdot)$. We conclude that such a mapping can be implemented by a transformer with $D_\psi + D_w + 1$ blocks.

Finally, we need to permute $(\mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t), S_h^t, \widehat{p}_h, F_4)$ to $(S_h^t, \widehat{p}_h, \mathbf{1}_{L'}^\top \texttt{embed}_h(S_h^t, F_4)$, which can be achieved by another linear layer. Therefore, the desired G_h can be implemented by $D_\psi + D_w + 2$ transformer blocks.

We compute the scaling factors from Proposition 39 when implementing each approximation module $\widehat{f}^w_{h,i}$ and $\widehat{f}^w_h = \{\widehat{f}^w_{h,i}\}_{i=1}^{|\mathcal{L}|}$ using transformers blocks. Furthermore, we conclude this proof by commenting on the width and norm of weight matrices of G_h.

In the construction of G_h, we first note that since we approximate each coordinate of the output distribution individually using $\widehat{f}^w_{h,i}$ for $i \in [|\mathcal{L}|]$, we horizontally stack the weight matrices for each $\widehat{f}^w_{h,i}$ at corresponding layers. Therefore, we derive an upper bound of the hidden layer size as $d_F \leq 16|\mathcal{L}| + 4 + 2|\mathcal{L}|$, where $16|\mathcal{L}|$ follows from the transformer implementation of $\widehat{f}^w_h$, and $2(2 + |\mathcal{L}|)$ follows from the preservation of columns $S^t_h, \widehat{p}_h, F_4$.

To implement $\widehat{\psi}^*_h$ using transformer blocks while preserving the inputs, we apply Proposition 39 by setting the scaling factors $\{B^\psi_\ell\}_{\ell=0}^{D_\psi}$ as

$$B^\psi_0 = B^\psi_{D_\psi} = \sqrt{B^2 + H^2 + 2|\mathcal{L}|(C_A+1)^2},$$
$$B^\psi_\ell = \sqrt{d_F \cdot ((B^\psi_0 + 1)\cdot 16^\ell)^2 + B^2 + H^2 + |\mathcal{L}|(C_A+1)^2}, \text{ for } \ell \in [D_\psi - 1],$$

where B^ψ_0 normalizes the input $(S^t_h, \widehat{p}_h, F_3, F_4)$ row-wisely, each B^ψ_ℓ keeps the intermediate outputs in a unit ball. Finally, $B^\psi_{D_\psi}$ follows since C_A upper bounds the magnitude of ψ^*_h by Assumption 37, which controls the magnitude of each row in $\widehat{\psi}^*_h(S^t_h)$. This scaling is absorbed into the average module realized via a MHA layer. Next, we consider the transformer implementation of $\widehat{f}^w_h$. Similar to the implementation of $\widehat{\psi}^*_h$, we apply Proposition 39 by setting the scaling factors $\{B^w_\ell\}_{\ell=0}^{D_w}$ as

$$B^w_0 = B^w_{D_w} = \sqrt{B^2 + H^2 + 2|\mathcal{L}|(C_A+1)^2},$$
$$B^w_\ell = \sqrt{d_F \cdot ((B^w_0 + 1)\cdot 16^\ell)^2 + B^2 + H^2 + |\mathcal{L}|(C_A+1)^2}, \text{ for } \ell \in [D_w - 1].$$

Finally, we compute the maximum network weight for the module G_h as

$$\max_{0\leq i\leq D_\psi, 0\leq j\leq D_w} \{B^\psi_i, B^w_j\} \cdot \sqrt{d_F^2 + 2(2+|\mathcal{L}|)} \leq C_F \cdot B_0 \cdot 16^{\max\{D_\psi, D_w\}} \cdot |\mathcal{L}|^{3/2},$$

where $B_0 = \sqrt{B^2 + H^2 + |\mathcal{L}| \cdot C_A^2}$ and $C_F > 1$ is an absolute constant.

■

I.4.7 Proof of Lemma 45

Proof

Recall that for each $h \in [H]$ and $\epsilon \in (0, 1/2)$, we want to construct an MLP that implements the trapezoid-shaped function

$$f_h(x) = \begin{cases} 1 & \text{for } |x-h| \leq \epsilon, \\ 1 - (|x-h| - \epsilon)/(1-2\epsilon) & \text{for } \epsilon < |x-h| \leq 1-\epsilon, \\ 0 & \text{otherwise.} \end{cases}$$

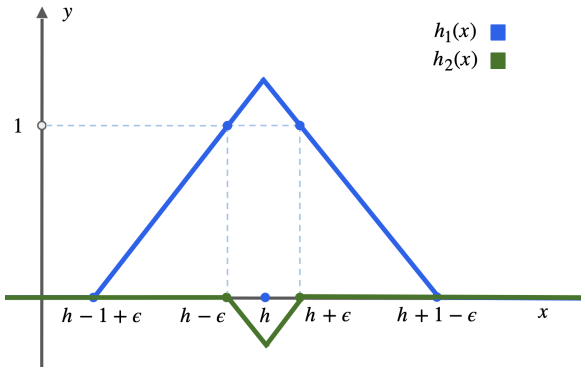


Figure 17: An illustration of the trapezoid functions $f_h(x)$ as the sum of two functions $h_1(x)$ and $h_2(x)$. Intuitively, $h_1(x)$ is a large upward-pointing triangle, and $h_2(x)$ is a smaller downward-pointing triangle that mirrors $h_1(x)$ within the interval $x \in [-\epsilon, \epsilon]$. The downward slope of $h_2(x)$ cancels out the upward slope of $h_1(x)$ within this interval, resulting in a flat region. Therefore, the sum of the two functions creates a trapezoid shape.

This function can be expressed as the sum of two triangular-shaped functions $h_1(x) + h_2(x)$, where we define h_1 and h_2 as

$$h_1(x) = \begin{cases} 1 - (|x-h| - \epsilon)/(1-2\epsilon) & \text{for } |x-h| \le 1-\epsilon, \\ 0 & \text{else,} \end{cases}$$
$$h_2(x) = \begin{cases} (|x-h| - \epsilon)/(1-2\epsilon) & \text{for } |x-h| \le \epsilon, \\ 0 & \text{else.} \end{cases}$$

Here h_1 is nonzero when $|x-h| \le 1-\epsilon$ with $h_1(h) = (1-\epsilon)/(1-2\epsilon)$, $h_1(h+1-\epsilon) = h_1(h-1+\epsilon) = 0$. Thus h_1 is a triangle pointing upwards. Similarly h_2 is a triangle pointing downwards with $h_2(h) = -\epsilon/(1-2\epsilon)$ and $h_2(h-\epsilon) = h_2(h+\epsilon) = 0$. See Figure 17 for an illustration of f_h, h_1, and h_2.

Furthermore, both h_1 and h_2 are piecewise linear functions with four linear pieces, and thus can be written as a sum of four ReLU functions. In particular, we can write h_1 as

$$h_1(x) = \frac{1}{1-2\epsilon}\Big(\texttt{ReLU}(x-(h+1-\epsilon)) + \texttt{ReLU}(h-1+\epsilon-x) \\ - \texttt{ReLU}(x-h) - \texttt{ReLU}(h-x) + 1 - \epsilon\Big),$$

which can be verified by direct calculation. Thus, this function can be written as a single feed-forward layer with parameters

$$W^1_{\text{ff},1} = (1,-1,1,-1,0), \quad b^1_{\text{ff},1} = (-(h+1-\epsilon), h-1+\epsilon, -h, h, 1-\epsilon),$$
$$W^1_{\text{ff},2} = (1,1,-1,-1,1)^\top/(1-2\epsilon), \quad b^1_{\text{ff},2} = \mathbf{0}.$$

Similarly, we can write h_2 as

$$h_2(x) = \frac{1}{1-2\epsilon}\Big(\texttt{ReLU}(x-(h+\epsilon)) + \texttt{ReLU}(h-\epsilon-x) - \texttt{ReLU}(x-h) - \texttt{ReLU}(h-x) - \epsilon\Big),$$

which can be written as a feed-forward layer with parameters

$$W^2_{\text{ff},1} = (1,-1,1,-1,0), \quad b^2_{\text{ff},1} = (-(h+\epsilon), h-\epsilon, -h, h, -\epsilon),$$
$$W^2_{\text{ff},2} = (1,1,-1,-1,1)^\top/(1-2\epsilon), \quad b^2_{\text{ff},2} = \mathbf{0}.$$

Finally, by directly concatenating the corresponding weight matrices for $h_1(\cdot)$ and $h_2(\cdot)$, we can implement the function f_h using a single feedforward (FF) layer. The width of the weight matrix in this FF layer is at most 10, and the magnitude of the weights is bounded by $H+1$. Thus, we conclude the proof. ■

I.5 Proofs of the Remaining Auxiliary Lemmas

In the following, we prove the remaining auxiliary lemmas, which include Lemma 46 used in the proof of Corollary 18, and Lemmas 47 and 48 used in the proofs in Appendix I.4.

I.5.1 Proof of Lemma 46

Proof

Let $\texttt{prompt}_{\text{CoT}}(n)$ denote a fixed prompt, then according to the chain rule of KL divergence, we have that

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(z^{\text{test}}_{1:H} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\text{test}}_{1:H} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))\big) \\ &= \sum_{h=1}^{H} \mathbb{E}_{z^{\text{test}}_{1:h-1}\sim\mathbb{P}(\cdot\,|\,\texttt{prompt}_{\text{CoT}}(n))}\Big[\texttt{KL}\big(\mathbb{P}(z^{\text{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\text{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\text{CoT}}(n))\big)\Big].\end{aligned} \tag{163}$$

Here the chain rule of KL divergence states that

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(X_2 = \cdot, X_3 = \cdot \,|\, X_1), \overline{\mathbb{P}}(X_2 = \cdot, X_3 = \cdot \,|\, X_1)\big) \\ &\quad = \texttt{KL}\big(\mathbb{P}(X_2 = \cdot, \,|\, X_1), \overline{\mathbb{P}}(X_2 = \cdot, \,|\, X_1)\big) \\ &\qquad + \mathbb{E}_{X_2\sim\mathbb{P}(\cdot\,|\,X_1)}\big[\texttt{KL}\big(\mathbb{P}(X_3 = \cdot \,|\, X_1, X_2), \overline{\mathbb{P}}(X_3 = \cdot, \,|\, X_1, X_2)\big)\big]\end{aligned}$$

holds for any three random variables (X_1, X_2, X_3) with two joint distributions $\mathbb{P}$ and $\overline{\mathbb{P}}$. Then according to data processing inequality, we have that

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(z^{\text{test}}_{1:H} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\text{test}}_{1:H} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))\big) \\ &\quad \geq \texttt{KL}\big(\mathbb{P}(z^{\text{test}}_H = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\text{test}}_H = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))\big).\end{aligned} \tag{164}$$

Notice that $y^{\text{test}} = z^{\text{test}}_H$. Combining (163) with (164), we have that

$$\begin{aligned}&\texttt{KL}\big(\mathbb{P}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(y^{\text{test}} = \cdot \,|\, \texttt{prompt}_{\text{CoT}}(n))\big) \\ &\leq \sum_{h=1}^{H} \mathbb{E}_{z^{\text{test}}_{1:h-1}\sim\mathbb{P}(\cdot\,|\,\texttt{prompt}_{\text{CoT}}(n))}\Big[\texttt{KL}\big(\mathbb{P}(z^{\text{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\text{CoT}}(n)), \mathbb{P}_{\widehat{\rho}}(z^{\text{test}}_h = \cdot \,|\, \texttt{prompt}^h_{\text{CoT}}(n))\big)\Big].\end{aligned}$$

Therefore, we conclude the proof. ■

I.5.2 Proof of Lemma 47

Proof In this proof, we extend Proposition III.3 in Elbrächter et al. (2021) to construct a sequence of FF modules such that $f : \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}^r$, where we want $f(x_{1:r}, y) \approx (x_1 y, \cdots, x_r y)$. For simplicity, we denote the input as $X^0 = (x_1, \cdots, x_r, y) \in \mathbb{R}^{r+1}$. By leveraging the construction by Elbrächter et al. (2021), we define a set of matrices $\{A_i, b_i\}_{i=1}^{m+2}$ as follows.

First, we define $A_{0,1} = (1, -1, 1, -1)/(2M)$ and $A_{0,2} = (1, -1, -1, 1)/(2M)$. Then for each $i \in [r]$, we define $A_{1,i} \in \mathbb{R}^{(r+1)\times 4}$ by setting the i-th row as $A_{0,1}$, $(r+1)$-th row as $A_{0,2}$ and fill the rest with zero. For example, $A_{1,1} = (A_{0,1}, \mathbf{0}_{(r-1)\times 4}, A_{0,2})^\top \in \mathbb{R}^{(r+1)\times 4}$. We stack the matrices $\{A_{1,i}\}_{i=1}^r$ horizontally to form $A_1 = (A_{1,1}, \cdots, A_{1,r}) \in \mathbb{R}^{(r+1)\times 4r}$. Then we define the bias $b_1 = \mathbf{0} \in \mathbb{R}^{4r}$.

To define $\{A_i, b_i\}_{i=2}^{m+2}$, we first define $\{A'_\ell, b'_\ell\}_{\ell=2}^{m+1}$ by letting

$$A'_2 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 1 \\ 0 & 0 & -1 & 1 & 1 \end{pmatrix} \in \mathbb{R}^{4\times 5}, \qquad b'_2 = \begin{pmatrix} 0 & -1/2 & 0 & 0 & -1/2 \end{pmatrix},$$

$$A'_\ell = \frac{1}{2M} \begin{pmatrix} 1/2 & 1/2 & -1/2 & 0 & 0 \\ -1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 1/2 & 1/2 \\ 0 & 0 & -1 & -1 & -1 \end{pmatrix} \in \mathbb{R}^{5\times 5}, \qquad b'_\ell = \begin{pmatrix} 0 & -2^{3-2\ell} & 0 & 0 & -2^{3-2\ell} \end{pmatrix},$$

for $3 \le \ell \le m+1$. Finally, we define $A'_{m+2} = (-1/2, 1, 1, 1/2, -1)^\top \in \mathbb{R}^{5\times 1}$ and $b'_{m+2} = 0 \in \mathbb{R}$. Next, for any $i \in \{2, \ldots, m+2\}$, we define A_i and b_i as

$$A_i = \mathtt{diag}(\underbrace{A'_i, \cdots, A'_i}_{r}), \qquad b_i = \big(\underbrace{b'_i, \cdots, b'_i}_{r}\big),$$

where each A_i is obtained by constructing a block-diagonal matrix with A'_i being the diagonal blocks, and each b_i is obtained by stacking b'_i horizontally for r times. Therefore we have $A_2 \in \mathbb{R}^{4r\times 5r}$, $A_{m+2} \in \mathbb{R}^{5r\times r}$, and $A_\ell \in \mathbb{R}^{5r\times 5r}$ for $3 \le \ell \le m+1$. Besides, we have $b_i \in \mathbb{R}^{1\times 5r}$ for each $2 \le \ell \le m+1$, and $b_{m+2} = \mathbf{0} \in \mathbb{R}^r$. Note that in the construction of Proposition III.3 by Elbrächter et al. (2021), a scalar multiplication module is used to restore the normalization $1/2M$ introduced by the first weight matrix A_1. In our approach, we instead scale the last weight matrix A_{m+2} by M^2, thereby eliminating the need for a separate scalar multiplication module.

Note that by setting $r = 1$, we recover the exact construction by Elbrächter et al. (2021) in Proposition III.3. We use $f_1 : \mathbb{R} \to \mathbb{R}$ to denote such a network, which is an MLP with parameters $\{A'_i, b'_i\}_{i=1}^{m+2}$, where $A'_1 = (A_{0,1}, A_{0,2})^\top \in \mathbb{R}^{2\times 4}$, $b'_1 = \mathbf{0} \in \mathbb{R}^4$, and $\{A'_i, b'_i\}_{i=2}^{m+2}$ are defined above. By the construction of the weight matrices $\{A'_i, b'_i\}_{i=1}^{m+2}$, the MLP with these parameters yields a vector-valued mapping $f\colon \mathbb{R}^r \times \mathbb{R} \to \mathbb{R}$ such that

$$f(x, y) = \big(f_1(x_1, y), \cdots, f_1(x_r, y)\big), \qquad \forall x \in \mathbb{R}^r, y \in \mathbb{R}. \tag{165}$$

As shown in Proposition III.3 in Elbrächter et al. (2021), when the depth of f_1, i.e., $m+2$, is bounded by $C_p(\log M + \log(1/\epsilon))$ for some constant $C_p > 0$, f_1 is a good approximator of the product operation in the sense that $|f_1(a,b) - ab| < \epsilon$ for any $a, b \in [-M, M]$.

Therefore, f constructed in (165) using weight matrices $\{A_i, b_i\}_{i=1}^{m+2}$ satisfies

$$\|f(x,y) - (x_1 y, \cdots, x_r y)\|_\infty = \max_{i\in[r]} |f(x_i, y) - x_i y| < \epsilon.$$

The depth of f is no more than $C(\log M + \log(1/\epsilon))$, and the maximum dimension of the hidden neurons is $5r$. The maximum magnitude of the intermediate weight matrices is bounded by 1, i.e., $\max\{\|A_i\|_\infty, \|b_i\|_\infty\} \le 1$ for all $i \in [m+1]$. Additionally, $\|A_{m+2}\|_\infty \le M^2$ due to the direct scaling, which replaces the scalar multiplication module.

To bound the Frobenius norms of weight matrices and bias vectors in f, by direct computation, we have

$$\max_{\ell\in[m+2]} \{\|A_\ell\|_{\mathrm{F}}, \|b_\ell\|_{\mathrm{F}}\} = \max\{\|A_2\|_{\mathrm{F}}, \|A_{m+2}\|_{\mathrm{F}}\} = \max\{\sqrt{12r}, H^2 \cdot \sqrt{5r}\}.$$

Finally, we compute the row-wise ℓ_2-norm for each intermediate output

$$X^\ell = \mathtt{ReLU}\big(X^{\ell-1} A_d + \mathbf{1}^\top b_\ell\big), \qquad \forall \ell \in [m+2],$$

where the initial input is given by $X^0 = (x_1, \cdots, x_r, y) \in \mathbb{R}^{r+1}$. Given that each entry of $X^{\ell-1}$ lies within the interval $[0,1]$, and noting that by construction $\|\mathbf{1} A_\ell + b_\ell\|_\infty \le 1$, for $\ell \ge 3$, we conclude that each entry of X^ℓ is also within $[0,1]$. We can calculate X^1 as:

$$X^1 = \frac{1}{2M} \cdot \mathtt{ReLU}\left(x_1 + y, -(x_1+y), x_1 - y, -(x_1 - y), \ldots, -(x_r - y)\right).$$

Since $x_1, \ldots, x_r, y \in [-M, M]$, each coordinate of X^1 is in $[0,1]$. Direct computation shows that

$$\begin{aligned} X^2 = \mathtt{ReLU}\big(&\frac{1}{2M} \cdot (|x_1+y|, |x_1+y|, |x_1+y| - |x_1 - y|, |x_1 - y|, |x_1 - y|, \cdots, |x_r - y|) \\ &- (0, 1/2, 0, 0, 1/2, \cdots, 1/2)\big), \end{aligned}$$

thus each coordinate of X^2 is in $[0,1]$. By induction, this implies that for each intermediate output X^ℓ (with $\ell \in [m+1]$), every element remains within $[0,1]$. In conclusion, we have that

$$\begin{aligned} \|X^0\|_{2,\infty} &= \|(x_1, \cdots, x_r, y)\|_2 \le M\sqrt{r+1}, \\ \|X^\ell\|_{2,\infty} &\le \sqrt{5r}, \text{ for any } \ell \in [m+1], \\ \|X^{m+2}\|_{2,\infty} &\le H^2. \end{aligned}$$

The first line follows from the direct calculation, and the second line holds because the maximum hidden embedding size is $5r$, thus a row in X^ℓ has length at most $5r$. These upper bounds on the ℓ_2-norm of X^ℓ will be used when implementing this fully connected network under a transformer. The total number of layers of this fully connected network is $C_p \cdot (\log M + \log(1/\epsilon))$, where C_p is an absolute constant. Now we conclude the proof. ■

I.5.3 Proof of Lemma 48

Proof In this proof, we first construct a pair of weight matrices $(W'_{\text{ff},1}, W'_{\text{ff},2})$ such that the output matrix keeps columns in $\mathcal{J}$ and set the other columns to a zero vector. Thus, the output matrix is in $\mathbb{R}^{m\times r}$. Then we modify $(W'_{\text{ff},1}, W'_{\text{ff},2})$ to form another pair of weight matrices $(W_{\text{ff},1}, W_{\text{ff},2})$ such that the FF layer truncates the zero columns to generate the desired output.

Since $a = \texttt{ReLU}(a) - \texttt{ReLU}(-a)$ for any $a \in \mathbb{R}$, by defining $W_1 = W_2^\top = (1, -1) \in \mathbb{R}^{1\times 2}$, we have $\texttt{ReLU}(aW_1)W_2 = a$. Setting $W'_1 = W'_2 = 0 \in \mathbb{R}$, we send a to zero by $\texttt{ReLU}(aW'_1)W'_2 = 0$. For each $i \in [r]$, define $W_{1,i} = \mathbb{1}\{i \in \mathcal{J}\} \cdot W_1 + \mathbb{1}\{i \notin \mathcal{J}\} \cdot W'_1$ and $W_{2,i} = \mathbb{1}\{i \in \mathcal{J}\} \cdot W_2 + \mathbb{1}\{i \notin \mathcal{J}\} \cdot W'_2$. For any $i \in [r]$, using $\{W_{1,i}, W_{2,i}\}$ as the weight matrices of a FF layer to process the i-th column $X_{:,i}$, the output is $\mathbb{1}\{i \in \mathcal{J}\} \cdot X_{:,i}$.

Now we put these matrices in the diagonal blocks of $W'_{\text{ff},1}$ and $W'_{\text{ff},2}$ to form

$$W'_{\text{ff},1} = \texttt{diag}(W_{1,i}, \cdots, W_{1,r}) \in \mathbb{R}^{r\times(r+|\mathcal{J}|)}, \qquad W'_{\text{ff},2} = \texttt{diag}(W_{2,i}, \cdots, W_{2,r}) \in \mathbb{R}^{(r+|\mathcal{J}|)\times r}.$$

By direct calculation, we have

$$\texttt{ReLU}(XW'_{\text{ff},1})W'_{\text{ff},2} = \big(X_{:,1} \cdot \mathbb{1}\{1 \in \mathcal{J}\} + \mathbf{0}_{m\times 1}, \ldots, X_{:,r} \cdot \mathbb{1}\{r \in \mathcal{J}\} + \mathbf{0}_{m\times 1}\big) \in \mathbb{R}^{m\times r}.$$

This output has the same shape as the input X.

This output keeps the dimension as $m \times r$. To get the final result, we define $W_{\text{ff},1} \in \mathbb{R}^{r\times 2|\mathcal{J}|}$ by removing all all-zero columns from $W'_{\text{ff},1}$, and $W_{\text{ff},2} \in \mathbb{R}^{2|\mathcal{J}|\times|\mathcal{J}|}$ by removing all all-zero rows and columns from $W'_{\text{ff},2}$. These are the submatrices of $W'_{\text{ff},1}$ and $W'_{\text{ff},2}$ used to process columns $X_{:,i}$'s with $i \in \mathcal{J}$. As a result, $W_{\text{ff},1}$ and $W_{\text{ff},2}$ have only $2|\mathcal{J}|$ nonzero entries, taking values in $\{-1, 0, 1\}$. Moreover, we have

$$\texttt{ReLU}(XW_{\text{ff},1})W_{\text{ff},2} = X_{:,i\in\mathcal{J}} \in \mathbb{R}^{m\times|\mathcal{J}|},$$

and the norms of these weight matrices are $\|W_{\text{ff},1}\|_{\text{F}} = \sqrt{2|\mathcal{J}|}$ and $\|W_{\text{ff},2}\|_{\text{F}} = \sqrt{2|\mathcal{J}|}$. ■

Appendix J. Technical Lemmas

Finally, in this appendix, we lay out the helper lemmas used in the proofs in previous appendices. These lemmas are directly obtained from existing works and we provide the references to their proofs.

Lemma 49 (Proposition 2 in Caponnetto and De Vito (2007)) *Let (Ω, ν) be a probability space and ξ be a random variable on Ω taking value in a real separable Hilbert space $\mathcal{H}$. We assume that there exist constants $B, \sigma > 0$ such that*

$$\big\|\xi(w)\big\|_{\mathcal{H}} \leq B/2, \text{ a.s.}, \quad \mathbb{E}\big[\|\xi\|_{\mathcal{H}}^2\big] \leq \sigma^2.$$

Then, it holds with probability at least $1-\delta$ that

$$\left\| L^{-1} \sum_{i=1}^{L} \xi(\omega_i) - \mathbb{E}[\xi] \right\| \leq 2\left(\frac{B}{L} + \frac{\sigma}{\sqrt{L}}\right) \log \frac{2}{\delta}.$$

Lemma 50 (Theorem A.4 in Foster et al. (2021)) *For any sequence of real random variables $\{X_i\}_{1 \leq i \leq n}$ that adapts to a filtration $\{\mathscr{F}_i\}_{1 \leq i \leq n}$, then for any $m \leq n$, with probability at least $1-\delta$,*

$$\sum_{i=1}^{n} X_i \leq \sum_{i=1}^{n} \log \mathbb{E}_{i-1}\big(e^{X_i}\big) + \log(\delta^{-1}).$$

Lemma 51 (Lemma I.10 in Zhang et al. (2023a)) *Let $b = \sup_x \log(p(x)/q(x))$. We have that*

$$\mathtt{KL}(p \,\|\, q) \leq 2(3+b) \cdot \mathtt{TV}(p, q).$$

Lemma 52 (Proposition E.1 in Zhang et al. (2023a)) *Let $\mathfrak{K}(a, b) = \exp(\gamma \cdot a^T b)$ denote exponential kernel with constant $\gamma > 0$, where $a, b \in \mathbb{R}^d$. We use $\mathbf{S}^{d-1}$ to denote a $d-1$-dimensional unit sphere. Then we have that*

$$\int_{\mathbf{S}^{d-1}} a \mathfrak{K}(a, b)\, da = C_1(\gamma) b,$$

for some constant $C_1(\gamma) = \int_{\mathbf{S}^{d-1}} (a^\top b) \exp(\gamma \cdot a^\top b) \mathrm{d}a > 0$ and all $b \in \mathbf{S}^{d-1}$. The constant C_1 does not depend on b due to symmetry on the unit sphere.

Proposition 53 (Proposition F.2 in Zhang et al. (2023a)) *Let $\mathcal{F}$ be the collection of functions of $f : \mathbb{R}^n \to \mathbb{R}$, and we assume that $|f| \leq b$ for any function $f \in \mathcal{F}$. Let $X_1, \cdots, X_N$ be N i.i.d. random variables. Let Q be a probability distributionover $\mathcal{F}$. With probability at least $1-\delta$, we have*

$$\left| \mathbb{E}_{f \sim P}\Big[\mathbb{E}_{X_1}\big[f(X_1)\big] - f(X) \Big] \right| \leq \sqrt{\frac{b^2}{2 \log 2 \cdot N}} \left[\mathtt{KL}(P \,\|\, Q) + \log \frac{4}{\delta} \right],$$

simultaneously for any distribution P on $\mathcal{F}$.

Lemma 54 (Corollary A.7 in Edelman et al. (2022)) *For any two vectors $x, y \in \mathbb{R}^r$,*

$$\|\texttt{softmax}(x) - \texttt{softmax}(y)\|_1 \leq 2\|x - y\|_\infty.$$

Lemma 55 (Lemma A.6 in Elbrächter et al. (2021)) *For $a, b \in \mathbb{R}$ with $a < b$, define*

$$\mathcal{S}_{[a,b]} = \left\{ f \in \mathcal{S}^\infty([a,b], \mathbb{R}) \mid \left\| f^{(n)}(x) \right\| \leq n! \text{ for all } n \in \mathbb{N} \right\}.$$

There exists a constant $C > 0$ such that for all $a, b \in \mathbb{R}$ with $a < b$, $f \in \mathcal{S}_{[a,b]}$, and $\varepsilon \in (0, 1/2)$, there is a fully connected network Ψ_f such that

$$\|f - \Psi_f\|_\infty \leq \varepsilon,$$

where the depth of the network is upper bounded by

$$C \cdot \max\{2, b-a\}(\log \varepsilon^{-1})^2 + \log(\lceil \max\{|a|, |b|\} \rceil) + \log(\lceil 1/(b-a) \rceil),$$

the width of the network is upper bounded by 16, and the maximum weight of the weight matrices is bounded by one. In particular, applying Propsotion 39, we compute the magnitude of each intermediate output as follows. For any input $x \in [a, b]$, let X^ℓ denote the output of the ℓ-th layer of the neural network, we have $X^\ell \leq 4\big(b + 1) \cdot 16^\ell$.

Appendix K. Supplementary Information about Prompts

This section provides supplementary examples for the area code experiment in Section 4.1 and the CityEquation experiment in Section 5.2.

K.1 Details of the Area Code Experiment in Section 4.1

The following includes the experimental details of testing the ChatGPT(gpt-3.5-turbo-16k with the temperature set to zero) on the area code task using vanilla ICL and CoT methods, respectively. The output of ChatGPT is colored in red.

The area code experiment with ICL prompt

Input prompt:

Q: The US=?
A: The answer is 2.

Q: France=?
A: The answer is 66.

Q: Japan=?
A:

LLM output:

The answer is 100.

As we can see, the answer provided by ChatGPT is wrong: it should be 162 instead of 100

The area code experiment with CoT prompt

Input prompt:

Q: The US=?
A: The US has area code 1, so the answer is 2.

Q: France=?
A: France has area code 33, so the answer is 66.

Q: Japan=?
A:

LLM output:

Japan has area code 81, so the answer is 162.

The output follows the same pattern shown in the demonstrations: it starts with stating the area code and computes the final answer.

K.2 Details of the CityEquation Experiment

This section provides the experimental details for the city arithmetic experiment discussed in Section 5.2.

Example prompts

We provide example prompts (2-shot) for vanilla ICL, informative CoT, Partially informative CoT-(a), Partially informative CoT-(c), and uninformative CoT methods, respectively. To highlight the differences among these five methods, we color code the useful intermediate steps in blue and the irrelevant intermediate steps in red for better comparison.

Example prompts for Vanilla ICL

Q: "Mumbai + Sydney"
A: The answer is 224.
Q: "New York + Seoul"
A: The answer is 53.
Q: "Paris + Beijing"
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with "The answer is".

Example prompts for Informative CoT

Q: "Mumbai + Sydney"
A: Using the longitudes of cities, the equation "Mumbai + Sydney" translates as "Mumbai" = 73, "Sydney" = 151. Here the longitudes of the western hemisphere are negative numbers. And we round the coordinates to the nearest integer. This gives the result. The answer is 224.
Q: "New York + Seoul"
A: Using the longitudes of cities, the equation "New York + Seoul" translates as "New York" = -74, "Seoul" = 127. Here the longitudes of the western hemisphere are negative numbers. And we round the coordinates to the nearest integer. This gives the result. The answer is 53.
Q: "Paris + Beijing"
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with "The answer is'/.

Example prompts for Partially informative CoT-(a)

Q: "Mumbai + Sydney"
A: Mumbai has longitude: 73. The answer is 224.
Q: "New York + Seoul"
A: New York has longitude: -74. The answer is 53.
Q: Paris + Beijing
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with "The answer is".

Example prompts for Partially informative CoT-(b)

Q: "Mumbai + Sydney"
A: The greater Mumbai area has around 20 million residents, predominantly of South Asian ethnicity. The median age is 31 years. Sydney has a population of around 5.3 million people, with a breakdown of 58% White, 34.2% Asian, and 2.6% Aboriginal/Torres Strait Islander. The remaining percentages include others. The median age is 36 years. The answer is 224.
Q: "New York + Seoul"
A: With a population of approximately 8.4 million people, New York is diverse. Around 32.1% identify as White, 29.1% as Hispanic, 24.3% as Black, 14.1% as Asian, with the rest being a mixture of Native American, Pacific Islander, and other ethnicities. The median age is 36 years. Seoul has a population of around 9.7 million, predominantly Korean, with a median age of 41 years. The answer is 53.
Q: "Paris + Beijing"
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with 'The answer is'.

Example prompts for Partially informative CoT-(c)

Q: "Mumbai + Sydney"
A: The greater Mumbai area has around 20 million residents, predominantly of South Asian ethnicity. The median age is 31 years. Sydney has a population of around 5.3 million people, with a breakdown of 58% White, 34.2% Asian, and 2.6% Aboriginal/Torres Strait Islander. The remaining percentages include others. The median age is 36 years. Mumbai has longitude: 73. The answer is 224.
Q: "New York + Seoul"
A: With a population of approximately 8.4 million people, New York is diverse. Around 32.1% identify as White, 29.1% as Hispanic, 24.3% as Black, 14.1% as Asian, with the rest being a mixture of Native American, Pacific Islander, and other ethnicities. The median age is 36 years. Seoul has a population of around 9.7 million, predominantly Korean, with a median age of 41 years.New York has longitude: -74. The answer is 53.
Q: "Paris + Beijing"
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with 'The answer is'.

Example prompts for Partially informative CoT-(d)

Q: "Mumbai + Sydney"
A: The greater Mumbai area has around 20 million residents, predominantly of South Asian ethnicity. The median age is 31 years. Sydney has a population of around 5.3 million people, with a breakdown of 58% White, 34.2% Asian, and 2.6% Aboriginal/Torres Strait Islander. The remaining percentages include others. The median age is 36 years. Mumbai has longitude: 73. Sydney has longitude: 151. The answer is 224.
Q: "New York + Seoul"
A: With a population of approximately 8.4 million people, New York is diverse. Around 32.1% identify as White, 29.1% as Hispanic, 24.3% as Black, 14.1% as Asian, with the rest being a mixture of Native American, Pacific Islander, and other ethnicities. The median age is 36 years. Seoul has a population of around 9.7 million, predominantly Korean, with a median age of 41 years. New York has longitude: -74. Seoul has longitude: 127. The answer is 53.
Q: "Paris + Beijing"
A: Concisely explain your steps and write your answer as an integer in the last sentence starting with "The answer is".

Failure cases

We present examples of failure cases for vanilla ICL, informative CoT, Partially informative CoT-(a), Partially informative CoT-(c), and uninformative CoT methods. The ground truth is written in blue, while the incorrect answer produced by GPT-4 is in red.

Failed example for Vanilla ICL

Testing query: Q: "Sao Paulo - Cairo",
Ground truth: -78,
LLM output: "The questions seem to be asking for the difference in time zones between two cities. Sao Paulo, Brazil is in the Brasilia Time Zone, which is GMT -3. Cairo, Egypt is in the Eastern European Time Zone, which is GMT +2. To find the difference, subtract the time zone of Sao Paulo from that of Cairo: -3 - (+2) = -5. The answer is -5."

Failed example for Informative CoT

Testing query: Q: "Moscow + New York",
Ground truth: -36,
LLM output: "Using the longitudes of cities, the equation "Moscow + New York" translates as "Moscow" = 37, "New York" = -74. Here the longitudes of the western hemisphere are negative numbers. And we round the coordinates to the nearest integer. This gives the result. The answer is -37."

Failed example for Partially informative CoT-(a)

Testing query: Q: "Sao Paulo - New York",
Ground truth: 27,
LLM output: "Let's consider the geographic coordinates that specify their east-west or north-south positions on the Earth's surface. Sao Paulo is located at approximately 46 degrees west, and New York is located at approximately 74 degrees west. If we subtract the longitude of Sao Paulo from that of New York, we get -28. The answer is -28."

Failed example for Partially informative CoT-(b)

Testing query: Q: "Sydney - New York",
Ground truth: 225,
LLM output: "Sydney has a population of around 5.3 million people, with a breakdown of 58% White, 34.2% Asian, and 2.6% Aboriginal/Torres Strait Islander. The remaining percentages include others. The median age is 36 years. New York City has a population of approximately 8.4 million people, with a diverse population that includes White (42.7%), Hispanic (29.1%), Black (24.3%), and Asian (14%) individuals. The median age is 36 years. The answer is -2."

Failed example for Partially informative CoT-(c)

Testing query: Q: "Mexico City + Beijing",
Ground truth: 17,
LLM output: "Mexico City has a population of approximately 21.7 million in its metropolitan area. The majority of residents are of Mestizo (mixed Indigenous and European) descent, with a significant number of Indigenous peoples, and a smaller number of European and other ethnicities. The median age is 27 years. Beijing, with a population of around 21.5 million, is predominantly Han Chinese, with a median age of 37.6 years. Mexico City has longitude: -99. The answer is -77."

Failed example for Partially informative CoT-(d)

Testing query: Q: "Moscow - Tokyo",
Ground truth: -102,
LLM output: "Moscow, the capital of Russia, has a population of approximately 12.5 million people, predominantly of Russian ethnicity. The median age is 40 years. Tokyo, the capital of Japan, has a population of around 14 million in its 23 special wards, predominantly of Japanese ethnicity. The median age in Tokyo is 45.5 years. Moscow has longitude: 37. Tokyo has longitude: 140. Subtracting Moscow's longitude from Tokyo's gives us 140 - 37. The answer is 103."

Upon closer inspection of GPT-4 outputs, incorrect results from informative CoT are solely due to rounding. That is, when rounding the coordinates to the nearest integer, GPT-4 makes an error. In particular, in the failed example shown above, the longitude of Moscow is 37.6, which should be rounded to 38 instead of 37. However, such a rounding error is the only error source. Therefore, with informative CoT, GPT-4 in fact understands that extracting the longitudes is the key to solving the CityEquation task.

Vanilla ICL prompts produce incorrect reasoning steps like using time zones, indicating a propensity to misinterpret prompts without clear guidance. Thus, it is challenging for vanilla ICL to realize that longitudes are the key to solving the CityEquations talk.

Furthermore, the errors incurred by Partially Informative CoT-(a) typically involve rounding and sign issues, particularly in reasoning steps related to longitudes. For example, in the failed case mentioned above, the longitude of New York is given as -74 instead of 74.

This suggests that while the prompts enable GPT-4 to associate the problem with the cities' longitudes, it sometimes struggles to handle the signs correctly. Additionally, the failure of Partially Informative CoT-(b) often results from the use of irrelevant information about the cities, such as demographic data, in the computation. In the failure example of Partially Informative CoT-(c) mentioned earlier, GPT-4 lists demographic data for both cities and the longitude of Mexico City, using both to compute the answer. Partially Informative CoT-(d) includes demographic data and longitudes for both cities in the intermediate reasoning steps. However, the final answer is based solely on the computation involving the longitudes. In the failure case above, the arithmetic formula incorrectly switches the minuend and subtrahend and rounds the longitude of Moscow to 37 inaccurately. This suggests that while GPT-4 correctly identifies that the longitudes are the only useful information for solving the task, it struggles with using this information correctly. By comparing Partially Informative CoT-(b) through CoT-(d), we observe that adding more relevant information to the CoT prompts improves GPT-4's performance on the CityEquation task.

Appendix L. Illustrative Example: Short vs. Long Chain-of-Thought

To further clarify the distinction discussed in the Conclusion, we present a concrete comparison between the Short CoT and Long CoT regimes. The key differentiator is not just the length, but the necessity of search, backtracking, and adaptive scaling required by the problem complexity.

Regime 1: Standard Short CoT (Fixed H)

- **Task Type:** Simple Arithmetic / Symbolic Manipulation.
- **Example Input:** "Alice has 1 apple, Bob has 2. How many in total?"
- **Reasoning Chain:**
  1. Alice has 1 apple.
  2. Bob has 2 apples.
  3. We add the two numbers together and get $1 + 2 = 3$.
- **Characteristics:** The reasoning path is linear and direct. The intrinsic task complexity (V) is low, allowing a small, fixed number of steps (H) to reliably map the input to the output without branching or error correction.

Regime 2: Long CoT (Inference-Time Scaling)

- **Task Type:** Complex Logical Reasoning / Olympiad Mathematics.
- **Example Input:** "Find all integers n such that $n^2 + n + 1$ is a perfect cube."
- **Reasoning Chain (Abbreviated):**
  1. *Exploration:* Let's test small values. $n = 0 \to 1$ (Yes). $n = 1 \to 3$ (No).
  2. *Hypothesis Generation:* Maybe $n = 0$ is the only solution. I should try to bound the expression between consecutive cubes.
  3. *Attempt 1:* Compare to $(n + 1)^3$. This is too large. Let's compare to n^3.
  4. *Verification/Self-Correction:* Wait, for $n > 0$, $n^2 + n + 1$ is strictly between n^3 and $(n + 1)^3$. This proves no solution exists for $n > 0$.
  5. *Backtracking:* I need to check negative integers separately...
  6. The chain continues...
- **Characteristics:** The chain is non-linear and involves exploration, backtracking, and self-correction. The high task complexity (V) necessitates a large H, where the model effectively trades test-time compute for accuracy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. Unified pre-training for program understanding and generation. *arXiv preprint arXiv:2103.06333*, 2021.

Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Pierre Alquier. User-friendly introduction to pac-bayes bounds. *arXiv preprint arXiv:2110.11216*, 2021.

Anthropic. The claude 3 model family: Opus, sonnet, haiku. *View in Article*, 2:42, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

Charles R. Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 1973.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 2007.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Tao He, Haotian Wang, Weihua Peng, Ming Liu, Bing Qin, and Ting Liu. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*, 2023.

Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36, 2024.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pages 5793–5831. PMLR, 2022.

Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. Deep neural network approximation theory. *IEEE Transactions on Information Theory*, 67(5):2581–2623, 2021.

Fabian Falck, Ziyu Wang, and Chris Holmes. Is in-context learning in large language models bayesian? a martingale perspective. *arXiv preprint arXiv:2406.00793*, 2024.

Guhao Feng, Yuntian Gu, Bohang Zhang, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *arXiv preprint arXiv:2305.15408*, 2023.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Deqing Fu, Tian-Qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn higher-order optimization methods for in-context learning: A study with linear models. *arXiv preprint arXiv:2310.17086*, 2023.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Michael Hahn and Navin Goyal. A theory of emergent in-context learning as implicit structure induction. *arXiv preprint arXiv:2303.07971*, 2023.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Jianliang He, Siyu Chen, Fengzhuo Zhang, and Zhuoran Yang. From words to actions: Unveiling the theoretical underpinnings of llm-driven autonomous systems. *arXiv preprint arXiv:2405.19883*, 2024.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial (with comments by m. clyde, david draper and ei george, and a rejoinder by the authors. *Statistical science*, 14(4):382–417, 1999.

Yifan Hou, Jiaoda Li, Yu Fei, Alessandro Stolfo, Wangchunshu Zhou, Guangtao Zeng, Antoine Bosselut, and Mrinmaya Sachan. Towards a mechanistic interpretation of multi-step reasoning capabilities of language models. *arXiv preprint arXiv:2310.14491*, 2023.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023a.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. *arXiv preprint arXiv:2310.05249*, 2023b.

Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

Hyuhng Joon Kim, Hyunsoo Cho, Junyeob Kim, Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. Self-generated in-context learning: Leveraging auto-regressive language models as a demonstration generator. *arXiv preprint arXiv:2206.08082*, 2022.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Akshay Krishnamurthy, Keegan Harris, Dylan J Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? *arXiv preprint arXiv:2403.15371*, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Yingcong Li, Muhammed Emrullah Ildiz, Dimitris Papailiopoulos, and Samet Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023a.

Yingcong Li, Kartik Sreenivasan, Angeliki Giannou, Dimitris Papailiopoulos, and Samet Oymak. Dissecting chain-of-thought: A study on compositional in-context learning of mlps. *arXiv preprint arXiv:2305.18869*, 2023b.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Aman Madaan and Amir Yazdanbakhsh. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*, 2022.

Arvind Mahankali, Tatsunori B Hashimoto, and Tengyu Ma. One step of gradient descent is provably the optimal in-context learner with one layer of linear self-attention. *arXiv preprint arXiv:2307.03576*, 2023.

David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 230–234, 1998.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904*, 2023.

Mary Phuong and Marcus Hutter. Formal algorithms for transformers. *arXiv preprint arXiv:2207.09238*, 2022.

Ben Prystawski, Michael Li, and Noah Goodman. Why think step by step? reasoning emerges from the locality of experience. *Advances in Neural Information Processing Systems*, 36, 2024.

Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*, 2021.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*, 2022.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*, 2022.

Taylor Sorensen, Joshua Robinson, Christopher Michael Rytting, Alexander Glenn Shaw, Kyle Jeffrey Rogers, Alexia Pauline Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. An information-theoretic approach to prompt engineering without ground truth labels. *arXiv preprint arXiv:2203.11364*, 2022.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, page 127063, 2023.

Xiaojuan Tang, Zilong Zheng, Jiaqi Li, Fanxu Meng, Song-Chun Zhu, Yitao Liang, and Muhan Zhang. Large language models are in-context semantic reasoners rather than symbolic reasoners. *arXiv preprint arXiv:2305.14825*, 2023.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Rasul Tutunov, Antoine Grosnit, Juliusz Ziomek, Jun Wang, and Haitham Bou-Ammar. Why can large language models generate correct chain-of-thoughts? *arXiv preprint arXiv:2310.13571*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

B Wang, S Min, X Deng, J Shen, Y Wu, L Zettlemoyer, and H Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. arxiv 2023. *arXiv preprint arXiv:2212.10001*, 2023a.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. *arXiv preprint arXiv:2305.14160*, 2023b.

Xinyi Wang, Wanrong Zhu, and William Yang Wang. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. *arXiv preprint arXiv:2301.11916*, 2023c.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Yu-An Wang and Yun-Nung Chen. What do position embeddings learn? an empirical study of pre-trained language model positional encoding. *arXiv preprint arXiv:2010.04903*, 2020.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Noam Wies, Yoav Levine, and Amnon Shashua. The learnability of in-context learning. *arXiv preprint arXiv:2303.07895*, 2023.

Jingfeng Wu, Difan Zou, Zixiang Chen, Vladimir Braverman, Quanquan Gu, and Peter L Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023a.

Skyler Wu, Eric Meng Shen, Charumathi Badrinath, Jiaqi Ma, and Himabindu Lakkaraju. Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions. *arXiv preprint arXiv:2307.13339*, 2023b.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601*, 2023.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Neural Information Processing Systems*, 2017.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Fengzhuo Zhang, Boyi Liu, Kaixin Wang, Vincent Tan, Zhuoran Yang, and Zhaoran Wang. Relational reasoning via set transformers: Provable efficiency and applications to marl. *Advances in Neural Information Processing Systems*, 35:35825–35838, 2022a.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023a.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*, 2022b.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, et al. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents. *arXiv preprint arXiv:2311.11797*, 2023b.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023c.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. Rethinking pre-training and self-training. *Advances in neural information processing systems*, 33:3833–3845, 2020.