Journal of Machine Learning Research 27 (2026) 1-58 Submitted 8/24; Revised 9/25; Published 3/26

# Enhancing Accuracy in Generative Models via Knowledge Transfer

**Xinyu Tian** TIANX@UMN.EDU
*School of Statistics*
*University of Minnesota*
*Minneapolis, MN 55455, USA*

**Xiaotong Shen** * XSHEN@UMN.EDU
*School of Statistics*
*University of Minnesota*
*Minneapolis, MN 55455, USA*

**Editor:** Ali Shojaie

## Abstract

This paper investigates the accuracy of generative models and the impact of knowledge transfer on their generation precision. Specifically, we examine a generative model for a target task, fine-tuned using a pre-trained model from a source task. Building on the "Shared Embedding" concept, which bridges the source and target tasks, we introduce a novel framework for transfer learning under distribution metrics such as the Kullback-Leibler divergence. This framework underscores the importance of leveraging inherent similarities between diverse tasks despite their distinct data distributions. Our theory suggests that the shared structures can augment the generation accuracy for a target task, reliant on the capability of a source model to identify shared structures and effective knowledge transfer from source to target learning. To demonstrate the practical utility of this framework, we explore the theoretical implications for two specific generative models: diffusion and normalizing flows. The results show enhanced performance in both models over their non-transfer counterparts, indicating advancements for diffusion models and providing fresh insights into normalizing flows in transfer and non-transfer settings. These results highlight the significant contribution of knowledge transfer in boosting the generation capabilities of these models.

**Keywords:** Knowledge Transfer, Generative models, Shared Embedding, Diffusion Models, Normalizing Flows

## 1. Introduction

Generative modeling, augmented with transfer learning, has seen considerable advancements in improving learning accuracy with scarce data. This process distills knowledge from extensive, pre-trained models previously trained on large datasets from relevant studies, enabling domain adaptation for specific tasks. At its core is the dynamic between the source (pre-trained) and target (fine-tuning) learning tasks, which tend to converge towards shared, concise representations. Yet, this principle has received less attention in diffusion models (Sohl-Dickstein et al., 2015; Dhariwal and Nichol, 2021) and normalizing flows (Dinh et al., 2014, 2016). This paper presents a theoretical framework to assess the accuracy of outputs from generative models, offering theoretical support for training generative models via transfer learning. For instance, it supports pre-training and fine-tuning of text-to-image models (Rombach et al., 2022; Zhou et al., 2023) for domain adaptation

∗. Corresponding author

and a synthesis approach (Shen et al., 2023) that employs high-fidelity synthetic data to boost the effectiveness of data analytics of downstream tasks through knowledge transfer.

Accurately evaluating the fidelity of data produced by generative models is increasingly critical for downstream analyses and for maintaining users' trust in synthetic data (Liu et al., 2024). Although empirical studies show that transfer learning improves diffusion-based generators for both images and tabular data (Wang et al., 2023; Kotelnikov et al., 2023; Shen et al., 2023), its theoretical effect on generative accuracy remains underexplored. Poorly matched source tasks can even induce "negative transfer," degrading performance and jeopardizing trustworthy AI goals through misleading scientific conclusions (Zhang et al., 2022; Gibney, 2022). By contrast, transfer learning in supervised settings has been thoroughly analyzed (Frégier and Gouray, 2021; Baxter, 2000; Maurer et al., 2016; Tripuraneni et al., 2020), underscoring the need for principled study in the generative realm. Complementing diffusion and flow research, Generative Adversarial Networks (GANs) provide a mature toolkit for domain adaptation: feature-level alignment via Domain-Adversarial Training (Ganin and Lempitsky, 2015); unpaired image-to-image translation with CycleGAN (Zhu et al., 2017); and recent multi-domain or data-efficient extensions such as StarGAN (Choi et al., 2020) and ADA (Karras et al., 2020). These methods demonstrate that adversarial alignment—whether in latent or pixel space—remains an effective paradigm for cross-domain generation.

We now review the relevant literature on the accuracy of two advanced generative models, diffusion models and normalizing flows. In diffusion models, Oko et al. (2023) derives convergence rates for unconditional generation for smooth densities, while Chen et al. (2023b) investigates distribution recovery over a low-dimensional linear subspace. Although a conditional diffusion model has shown effectiveness (Batzolis et al., 2021), its theoretical foundation remains underexplored. Recently, Fu et al. (2024) investigated conditional diffusion models under a smooth density assumption. By comparison, the study of generation accuracy for flows remains sparse, with limited exceptions on universal approximation (Koehler et al., 2021).

This paper develops a comprehensive theoretical framework for transfer learning that addresses the accuracy of target generation. This generation accuracy, measured by the excess risk, induces several valuable metrics such as the Kullback-Leibler (KL) divergence to assess the distribution closeness. To the best of our knowledge, this study is the first to outline the bounds of generation accuracy in the context of transfer learning. The contributions of this paper are as follows:

**1). Generation accuracy theory.** We introduce the concept of the "Shared Embedding" condition (SEC) to quantify the similarities between the latent representations of source and target learning. The SEC distinguishes between conditional and unconditional generation by featuring nonlinear dimension reduction for the former while capturing shared latent representations through embeddings for the latter. Our theoretical framework establishes generation error bounds for conditional and unconditional models. These bounds incorporate factors such as complexity measures and approximation errors while leveraging the transferability principle via shared structures. This theory offers statistical guarantees for the efficacy of generative models through knowledge transfer while demonstrating that such models can achieve rapid convergence rates for the target task under metrics stronger than commonly used total variation TV-norm. Achieving this involves leveraging the common structures for dimension reduction.

**2). Diffusion models and normalizing flows via transfer learning.** We leverage the general theoretical framework to unveil new insights into the precision of both conditional and unconditional generation. This exploration examines conditional generation with the KL divergence and the TV-norm for smooth target distributions and unconditional generation with the dimension-scaling

Wasserstein distance, specifically in diffusion and coupling flows, as detailed in Theorems 1-8. Our focus is on the prevalent practices with smooth distributions through continuous embeddings. The analysis reveals that utilizing transfer learning strategies—grounded in the shared embedding structures within the lower-dimensional manifold that bridges the source and target learning—holds the potential to elevate performance over non-transfer methodologies.

**3). Non-transfer diffusion models and normalizing flows.** This paper investigates non-transfer generative models, an area attracting considerable interest. Our results demonstrate that diffusion models structured with the SEC framework achieve a faster KL rate than their non-transfer analogs in the TV-norm for conditional generation with a smooth density (Fu et al., 2024), where Fu et al. (2024) aligns with the minimax rate in Oko et al. (2023) without dimension reduction capabilities, albeit with a logarithmic factor. In unconditional generation, our method exhibits a faster rate under the Wasserstein distance relative to that under the TV-norm (Oko et al., 2023). Crucially, our analysis of coupling flows reveals its competitiveness compared to diffusion models in both conditional and unconditional generation; see Section 4 for details. These results enrich our understanding of these models' complexities and strengths.

This article comprises seven sections. Section 2 outlines the transfer learning framework for generative tasks. Section 3 applies the supplementary theory to diffusion models, deriving new results to illustrate knowledge transfer. Section 4 introduces a novel finding for normalizing flows. Section 5 presents the core proof strategy that establishes accuracy guarantees for generative models enhanced by transfer learning. Section 6 illustrates the core theory through numerical examples. Finally, Section 7 concludes the article. The Appendix contains technical details and experiment details.

## 2. Enhancing generation accuracy and knowledge transfer

Within the framework of synthesizing random samples that approximate a target distribution, the transfer learning approach leverages a pre-trained generative model trained on a source domain. This method fine-tunes the target generative model using the source model and training data from the target distribution, thereby enabling sample generation that accurately reflects the target distribution.

As a starting point, we adopt a basic independence assumption between the source and target datasets.

**Assumption 1** *The source and target data are assumed to be independent.*

To facilitate transfer learning between source and target tasks, we next introduce the procedures and necessary conditions for both conditional and unconditional generation.

### 2.1 Conditional generation

In the target task, we train a conditional generator for $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$ using a target training sample $\mathcal{D}_t = \{\boldsymbol{x}_t^i, \boldsymbol{z}_t^i\}_{i=1}^{n_t}$, whereas source training occurs separately with an independent source training sample $\mathcal{D}_s = \{\boldsymbol{x}_s^i, \boldsymbol{z}_s^i\}_{i=1}^{n_s}$.

**SEC for conditional generation.** We introduce the "Shared Embedding" condition for conditional generation. Denote the target and source covariate vectors by $\boldsymbol{X}_t$ and $\boldsymbol{X}_s$, which are allowed to differ in dimensionality. To sample from the conditional distribution of the target covariates given an auxiliary vector $\boldsymbol{Z}_t$, denoted by $P_{\boldsymbol{X}_t|\boldsymbol{Z}_t}$, we transfer information from the source task to improve

target-side estimation. Decompose the auxiliary vectors as

$$\boldsymbol{Z}_t = (\boldsymbol{Z}, \boldsymbol{Z}_{t^c}), \quad \boldsymbol{Z}_s = (\boldsymbol{Z}, \boldsymbol{Z}_{s^c}),$$

where the common block $\boldsymbol{Z} \in \mathbb{R}^{d_c}$ is shared across tasks and $\boldsymbol{Z}_{j^c}$ contains the task-specific remainder; $j \in \{s, t\}$, and $d_c$ is the dimension of $\boldsymbol{Z}$.

**Shared Embedding Condition (SEC).** Assume there exists a latent representation $h(\boldsymbol{Z})$ that is common to both tasks such that the conditional laws factor through task-specific decoders $P_t$ and $P_s$:

$$P_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\cdot|\boldsymbol{z}_t) = P_t(\cdot, h_t(\boldsymbol{z}_t)), \quad P_{\boldsymbol{x}_s|\boldsymbol{z}_s}(\cdot|\boldsymbol{z}_s) = P_s(\cdot, h_s(\boldsymbol{z}_s)). \tag{1}$$

where $h_j(\boldsymbol{z}_j) = (h(\boldsymbol{z}), \boldsymbol{z}_{j^c})$ and $P_j$ is a suitable probability function; $j \in \{s, t\}$. For an explicit illustration of SEC, see Figure 1, which highlights the shared-embedding architecture within target and source diffusion models. This design parallels practical fine-tuning strategies in text-to-image pipelines, where the SEC is a reasonable assumption: the semantic representation of text is largely transferable across tasks, and the model architecture reflects this by freezing the text-embedding module while adapting the diffusion backbone.

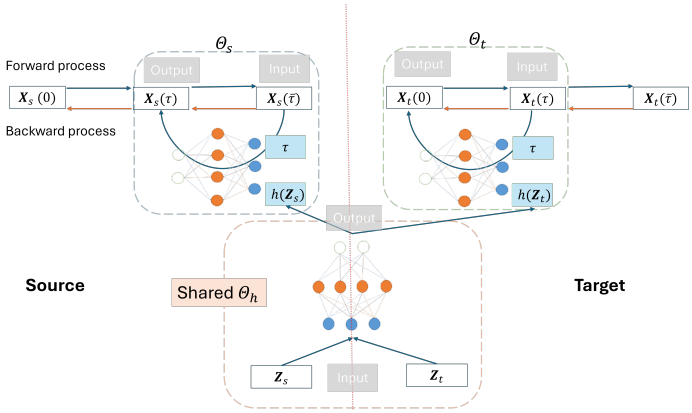



Figure 1: Shared architecture for conditional diffusion generation. A common backbone $\Theta_h$ is first trained on the source data and subsequently fed into the target diffusion model $\Theta_t$ to transfer knowledge.

The SEC in (1) presents a dimension reduction framework, indicating that $P_{\boldsymbol{x}_j|\boldsymbol{z}_j}$ depends on a shared manifold mapping $h(\boldsymbol{z})$, generally of lower dimension; $j \in \{s, t\}$. For instance, a source task of text prompt-to-image ($\boldsymbol{Z}_s$ to $\boldsymbol{X}_s$) and a target task of text prompt-to-music ($\boldsymbol{Z}_t$ to $\boldsymbol{X}_t$) may share common elements $\boldsymbol{Z}$ based on a latent semantic representation $h(\boldsymbol{Z})$ and task-specific elements $\boldsymbol{Z}_{j^c}$; $j \in \{s, t\}$. This framework broadens the scope of dimension reduction from linear subspaces (Li, 1991, 2018) to nonlinear manifolds, where $\boldsymbol{X}_j$ is conditionally independent of $\boldsymbol{Z}_j$ given $(h(\boldsymbol{Z}), \boldsymbol{Z}_{j^c})$.

Here, our primary focus is on understanding the interplay between the source distribution of $\boldsymbol{X}_s$ given $\boldsymbol{Z}_s$, $P_{\boldsymbol{x}_s|\boldsymbol{z}_s}(\boldsymbol{x}|\boldsymbol{z}_s) = P_s(\boldsymbol{x}, h_s(\boldsymbol{z}_s))$, and the target distribution of $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$, $P_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\boldsymbol{x}|\boldsymbol{z}_t) = P_t(\boldsymbol{x}, h_t(\boldsymbol{z}_t))$, through the shared SEC component $h$ between $h_s$ and $h_t$.

To learn $P_{\boldsymbol{x}_j|\boldsymbol{z}_j}$, we parameterize it as $P_{\boldsymbol{x}_j|\boldsymbol{z}_j}(\boldsymbol{x}, \boldsymbol{z}_j) = P_j(\boldsymbol{x}, h_j(\boldsymbol{z}_j); \theta_j)$ with $h$ from $\Theta_h$, the space of latent embeddings, and $\theta_j$ from $G_j$, either parametric or nonparametric; $j = s, t$. This approach defines the true distribution $P^0_{\boldsymbol{x}_j|\boldsymbol{z}_j}(\boldsymbol{x}, \boldsymbol{z}_j) = P_j(\boldsymbol{x}, h^0_j(\boldsymbol{z}_j); \theta^0_j)$ through true parameters $h^0_j(\boldsymbol{z}_j) = (h^0(\boldsymbol{z}), \boldsymbol{z}_{j^c})$ and $(\theta^0_j, h^0) = \arg\min_{\theta_j \in G_j, h \in \Theta_h} \mathrm{E}_{\boldsymbol{x}_j, \boldsymbol{z}_j} l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta_j, h)$, minimizing the expected loss $l_j$; $j = s, t$ [1]. Here, $\mathrm{E}_{\boldsymbol{x}_j, \boldsymbol{z}_j}$ is the expectation of $(\boldsymbol{X}_j, \boldsymbol{Z}_j)$ and we use $\Theta_j$ (a class of neural networks) as the action parameter space approximating the parameter space $G_j$ (a class of candidate functions); $j = s, t$. For the source task, we minimize its empirical loss $L_s(\theta_s, h) = \sum_{i=1}^{n_s} l_s(\boldsymbol{x}^i_s, \boldsymbol{z}^i_s; \theta_s, h)$ on a source training sample to yield

$$(\hat{\theta}_s, \hat{h}) = \underset{\theta_s \in \Theta_s, h \in \Theta_h}{\arg\min} \; L_s(\theta_s, h), \tag{2}$$

where $\Theta_h$ ensures latent structure identifiability. With $\hat{h}$, we minimize the target empirical loss $L_t(\theta_t, \hat{h}) = \sum_{i=1}^{n_t} l_t(\boldsymbol{x}^i_t, \boldsymbol{z}^i_t; \theta_t, \hat{h})$ to yield $\hat{\theta}_t = \arg\min_{\theta_t \in \Theta_t} L_t(\theta_t, \hat{h})$. The estimated distribution is $\hat{P}_{\boldsymbol{x}_j|\boldsymbol{z}_j}(\boldsymbol{x}|\boldsymbol{z}_j) = P_{\boldsymbol{x}_j|\boldsymbol{z}_j}(\boldsymbol{x}, \hat{h}_j(\boldsymbol{z}_j); \hat{\theta}_j)$, where $\hat{h}_t(\boldsymbol{z}_j) = (\hat{h}(\boldsymbol{z}), \boldsymbol{z}_{j^c})$. The distribution discrepancy is controlled by the excess risk $\mathrm{E}_{\boldsymbol{x}_j, \boldsymbol{z}_j}(l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta_j, h) - l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta^0_j, h^0))$. For example, the negative log-likelihood loss yields an error bound in the excess risk, implying that in the KL divergence.

**Assumption 2 (Transferability for conditional models)** *For some positive constant $c_1 > 0$ and $h \in \Theta_h$, $|\delta_t(h) - \delta_t(h^0)| \le c_1 |\delta_s(h) - \delta_s(h^0)|$, where $\delta_j(h) = \inf_{\theta_j \in \Theta_j} \mathrm{E}_{\boldsymbol{x}_j, \boldsymbol{z}_j}[l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta_j, h) - l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta^0_j, h^0)]$; $j \in \{s, t\}$.*

Assumption 2 characterizes the transitions of the excess risk for the latent structural representation $h$ from source to target tasks. A similar condition has been in a different context (Tripuraneni et al., 2020).

Denote the excess risk as $\rho^2_j(\gamma^0_j, \gamma_j) = \mathrm{E}_{\boldsymbol{x}_j, \boldsymbol{z}_j}[l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta_j, h) - l_j(\boldsymbol{X}_j, \boldsymbol{Z}_j; \theta^0_j, h^0)]$ with $\gamma_j = (\theta_j, h)$; $j \in \{s, t\}$. The following assumption specifies the generation error bound of the source for $\boldsymbol{X}_s$ given $\boldsymbol{Z}_s$, facilitating the target learning through transfer learning.

**Assumption 3 (Source error)** *There exists a sequence $\varepsilon_s$ indexed by $n_s$, such that the source generation error satisfies, for any $\varepsilon \ge \varepsilon_s$, $P(\rho_s(\gamma^0_s, \hat{\gamma}_s) \ge \varepsilon) \le \exp\left(-c_2 n_s^{1-\xi} \varepsilon^2\right)$, where $c_2 > 0$ and $\xi > 0$ are constants, $\hat{\gamma}_s = (\hat{\theta}_s, \hat{h})$ is defined in* (2)*, and $n_s^{1-\xi} \varepsilon_s^2 \to \infty$ as $n_s \to \infty$.*

Because the source error for $h$ is typically intertwined with that of $\theta_s$, any estimation error in $\theta_s$ carries over to $h$ through Assumption 2 and, in turn, affects the error in conditional generation.

### 2.2 Unconditional generation

**SEC for unconditional generation.** To sample from the marginal target distribution $P_{\boldsymbol{X}_t}$, we transfer a latent representation learned on the source task. The SEC postulates that the source and target variables, $\boldsymbol{X}_s$ and $\boldsymbol{X}_t$, arise from a *shared* latent vector $\boldsymbol{U}$ through task-specific decoders.

1. Although the minimizer of $\ell(\theta)$ may not be unique, we use the shorthand $\theta^* = \arg\min_{\theta \in \Theta} \ell(\theta)$ to denote some minimizer, i.e., $\ell(\theta^*) = \min_{\theta \in \Theta} \ell(\theta)$.

Let $g_t$ and $g_s$ map the latent space to the target and source observation spaces, respectively, so that $\boldsymbol{X}_t = g_t(\boldsymbol{U})$ and $\boldsymbol{X}_s = g_s(\boldsymbol{U})$. Consequently,

$$P_{\boldsymbol{x}_t}(\cdot) = P_{\boldsymbol{u}}(g_t^{-1}(\cdot)), \quad P_{\boldsymbol{x}_s}(\cdot) = P_{\boldsymbol{u}}(g_s^{-1}(\cdot)), \tag{3}$$

where $P_{\boldsymbol{u}}$ denotes the probability distribution of $\boldsymbol{u}$ and $g^{-1}$ denotes the inverse image of $g$, or $\{\boldsymbol{u} : g(\boldsymbol{u}) = x_t\}$. Figure 2 gives a concrete illustration of this shared-embedding architecture for source and target diffusion models. This configuration parallels the practical fine-tuning workflow for latent diffusion models, in which the diffusion backbone is frozen and only the decoder is adapted. The setting also mirrors style-transfer tasks, where diverse objectives rely on the same underlying visual content (e.g., the structural distribution of digit images captured by the diffusion module). At the same time, task-specific decoders transform this shared representation into distinct outputs (e.g., adapting to different handwriting styles through the decoder module).

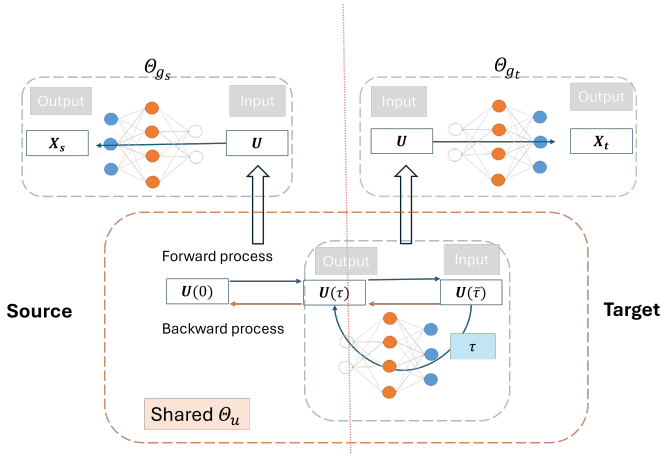



Figure 2: Shared architecture for unconditional diffusion generation. A common backbone $\Theta_u$ is first trained on the source data and subsequently fed into the target diffusion model $\Theta_u$ to transfer knowledge.

This SEC in (3) highlights that the source and target distributions share a common latent distribution within low-dimensional manifolds, defined by latent representations $g_s$ and $g_t$. For example, consider a source task of French text generation $\boldsymbol{X}_s$ from English and a target task of Chinese text generation $\boldsymbol{X}_t$ from English. Initially, the numerical embedding of a textual description $\boldsymbol{U}$ in English is transformed into French and Chinese using the transformations $g_s$ and $g_t$, respectively.

Given a latent representation $\{\boldsymbol{u}_s^i\}_{i=1}^{n_s}$ for $\{\boldsymbol{x}_s^i\}_{i=1}^{n_s}$ in the source training sample $\mathcal{D}_s$ encoded by an encoder, we first estimate the latent distribution $P_u$ using a generative model parameterized by $\Theta_u$ as $\hat{P}_u = P_u(\cdot; \hat{\theta}_u)$. Here, $\hat{\theta}_u = \arg\min_{\theta_u \in \Theta_u} L_u(\theta_u) = \arg\min_{\theta_u \in \Theta_u} \sum_{i=1}^{n_s} l_u(\boldsymbol{u}_s^i; \theta_u)$, where $\sum_{i=1}^{n_s} l_u(\boldsymbol{u}_s^i; \theta_u)$ represents an empirical loss to estimate $\theta_u$. With the estimated latent distribution $\hat{P}_u$, $g_t$ is estimated as $\hat{g}_t = \arg\min_{g_t \in \Theta_{g_t}} L_{g_t}(g_t) = \arg\min_{g_t \in \Theta_{g_t}} \sum_{i=1}^{n_t} l_{g_t}(\boldsymbol{u}_t^i, \boldsymbol{x}_t^i; g_t)$ based on a target training sample $\mathcal{D}_t = \{\boldsymbol{u}_t^i, \boldsymbol{x}_t^i\}_{i=1}^{n_t}$. Then $P_{\boldsymbol{x}_t}$ is estimated by $\hat{P}_{\boldsymbol{x}_t}(\cdot) = \hat{P}_u(\hat{g}_t^{-1}(\cdot))$.

Assumption 2 is not required for unconditional generation due to different shared structures in (3), with a separable loss function for $\theta_u$ and $g_t$. As in the conditional setting, $\theta_u^0$ is defined as the

minimizer of the population loss, given by $\theta_u^0 = \arg\min_{\theta_u \in G_u} \mathrm{E}_u l_u(\boldsymbol{u}; \theta_u)$, where $G_u$ represents the parameter space of $\theta_u$. Let $\rho_u^2(\theta_u^0, \theta_u) = \mathrm{E}_u[l_u(\boldsymbol{U}; \theta_u) - l_u(\boldsymbol{U}; \theta_u^0)]$.

Moreover, we assume an analogous condition to Assumption 3 for the source error of $\rho_u(\theta_u^0, \hat{\theta}_u)$.

**Assumption 4 (Source error for $\boldsymbol{U}$)** *There exists a sequence $\varepsilon_s$ indexed by $n_s$, such that the source generation error for $\boldsymbol{U}$ by estimating $\theta_u$ from $\{\boldsymbol{u}_s^i\}_{i=1}^{n_s}$ satisfies for any $\varepsilon \geq \varepsilon_s^u$, $P(\rho_u(\theta_u^0, \hat{\theta}_u) \geq \varepsilon) \leq \exp\left(-c_3 n_s^{1-\xi} \varepsilon^2\right)$, where $c_3 > 0$ and $\xi > 0$ are constants and $n_s^{1-\xi}(\varepsilon_s^u)^2 \to \infty$ as $n_s \to \infty$.*

The error bound $\varepsilon_s$, $\varepsilon_s^u$, $\xi$, $c_2$ and $c_3$ in Assumptions 3 and 4 can be determined in a specific source model; cf. Lemmas 21, 28, 36, and 39.

Before detailing the diffusion and flow frameworks, we present a concise overview of our theoretical guarantees in Table 1. This table lists each generative model, indicates whether the task is conditional or unconditional, specifies the transfer regime, enumerates the key assumptions, cites the corresponding theorem, and identifies the performance metric.

Table 1: Key assumptions, formal results, and distance metrics underlying our theoretical guarantees.

| Model | Task | Regime | Assumptions | Theorem | Metric |
|---|---|---|---|---|---|
| Diffusion | Conditional generation | Transfer | SEC; transferability; density smoothness; source error | 1 | TV, KL |
| | | Non-transfer | SEC; density smoothness | 2 | TV, KL |
| | | General | Density smoothness | 13 | TV, KL |
| | Unconditional generation | Transfer | SEC; source error; transformation smoothness | 3 | Wasserstein |
| | | Non-transfer | SEC; transformation smoothness | 4 | Wasserstein |
| Flow | Conditional generation | Transfer | SEC; transferability; transformation smoothness; source error | 5 | KL |
| | | Non-transfer | SEC; transformation smoothness | 6 | KL |
| | | General | Transformation smoothness | 29 | KL |
| | Unconditional generation | Transfer | SEC; source error; transformation smoothness | 7 | Wasserstein |
| | | Non-transfer | SEC; transformation smoothness | 8 | Wasserstein |

## 3. Diffusion models

This section considers diffusion models, following the setup in Section 2.

### 3.1 Forward and Backward Processes

A diffusion model incorporates both forward and backward diffusion processes.

**Forward process.** The forward process systematically transforms a random vector $\boldsymbol{X}(0)$ into pure white noise by progressively injecting white noise into a differential equation defined with the Ornstein-Uhlenbeck process, leading to diffused distributions from the initial state $\boldsymbol{X}(0)$:

$$\mathrm{d}\boldsymbol{X}(\tau) = -b_\tau \boldsymbol{X}(\tau)\mathrm{d}\tau + \sqrt{2b_\tau}\mathrm{d}W(\tau), \quad \tau \geq 0, \tag{4}$$

where $\boldsymbol{X}(\tau)$ has a probability density $p_{\boldsymbol{x}(\tau)}$, $\{W(\tau)\}_{\tau\geq 0}$ represents a standard Wiener process and $b_t$ is a non-decreasing weight function. Under (4), $\boldsymbol{X}(\tau)$ given $\boldsymbol{X}(0)$ follows $N(\mu_\tau \boldsymbol{X}(0), \sigma_\tau^2 \boldsymbol{I})$, where $\mu_\tau = \exp(-\int_0^\tau b_s \mathrm{d}s)$ and $\sigma_\tau^2 = 1 - \mu_\tau^2$. By setting $b_s = 1$, we simplify the model to $\mu_\tau = \exp(-\tau)$ and $\sigma_\tau^2 = 1 - \exp(-2\tau)$. In practice, the diffusion process terminates at a sufficiently large $\overline{\tau}$, ensuring the distribution of $\boldsymbol{X}(\tau)$, which is a mixture of the original state $\boldsymbol{X}(0)$ and white noise, approximates a standard Gaussian vector.

**Backward process.** Given $\boldsymbol{X}(\overline{\tau})$ in (4), a backward process is employed for sample generation for $\boldsymbol{X}(0)$. Assuming (4) satisfies certain conditions (Anderson, 1982), the backward process $\boldsymbol{V}(\tau) = \boldsymbol{X}(\overline{\tau} - \tau)$, starting with $\boldsymbol{X}(\overline{\tau})$, is derived as:

$$\mathrm{d}\boldsymbol{V}(\tau) = b_{\overline{\tau}-\tau}(\boldsymbol{V}(\tau) + 2\nabla \log p_{\boldsymbol{x}(\overline{\tau}-\tau)}(\boldsymbol{X}(\overline{\tau}-\tau))\mathrm{d}\tau + \sqrt{2b_{\overline{\tau}-\tau}}\mathrm{d}W(\tau); \quad \tau \geq 0, \tag{5}$$

where $\nabla \log p_{\boldsymbol{x}}$ is the score function which represents the gradient of $\log p_{\boldsymbol{x}}$.

**Score matching.** To estimate the unknown score function, we minimize a matching loss between the score and its approximator $\theta$: $\int_0^{\overline{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau)} \|\nabla \log p_{\boldsymbol{x}(\tau)}(\boldsymbol{X}(\tau)) - \theta(\boldsymbol{X}(\tau), \tau)\|^2 \mathrm{d}\tau$, where $\|\boldsymbol{x}\| = \sqrt{\sum_{j=1}^{d_x} \boldsymbol{x}_j^2}$ is the Euclidean norm, which is equivalent to minimizing the following loss Oko et al. (2023),

$$\int_{\underline{\tau}}^{\overline{\tau}} \mathrm{E}_{\boldsymbol{x}(0)} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)} \|\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}(\boldsymbol{X}(\tau)|\boldsymbol{X}(0)) - \theta(\boldsymbol{X}(\tau), \tau)\|^2 \mathrm{d}\tau, \tag{6}$$

with $\underline{\tau} = 0$. In practice, to avoid score explosion due to $\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)} \to \infty$ as $\tau \to 0$ and to ensure training stability, we restrict the integral interval to $\underline{\tau} > 0$ (Oko et al., 2023; Chen et al., 2023a) in the loss function. Then, both the integral and $\mathrm{E}_{\boldsymbol{x}(0)}$ can be precisely approximated by sampling $\tau$ from a uniform distribution on $[\underline{\tau}, \overline{\tau}]$ and a sample of $\boldsymbol{X}(0)$ from the conditional distribution of $\boldsymbol{X}(0)$ given $\boldsymbol{Z}$.

**Generation.** To generate a random sample of $\boldsymbol{V}(\tau)$, we replace the score $\nabla \log p_{\boldsymbol{x}(\overline{\tau}-\tau)}$ by its estimate $\hat{\theta}$ in (5) to yield $\boldsymbol{V}(\tau)$ in the backward equation. For implementation, we may utilize a discrete-time approximation of the sampling process, facilitated by numerical methods for solving stochastic differential equations, such as Euler-Maruyama and stochastic Runge-Kutta methods (Song et al., 2020).

### 3.2 Conditional diffusion via transfer learning

To generate a target sample from $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$, we use a conditional diffusion model to learn the conditional probability density $p_{\boldsymbol{x}_t|\boldsymbol{z}_t}$, as described in (4)-(5).

In this approach, we assign $\boldsymbol{X}(0) = \boldsymbol{X}_t$ to our target task in (4). After deriving an estimated latent structure $\hat{h}$ from the pre-trained diffusion model (source), we employ a conditional diffusion model (target), transferring $\hat{h}$ to improve the synthesis task of generating $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$.

Given a target training sample $(\boldsymbol{x}_t^i, \boldsymbol{z}_t^i)_{i=1}^{n_t}$, we follow (4)-(5) to construct an empirical score matching loss $L_t(\theta_t, \hat{h}) = \sum_{i=1}^{n_t} l_t(\boldsymbol{x}_t^i, \boldsymbol{z}_t^i; \theta_t, \hat{h})$ in (6) with

$$l_t(\boldsymbol{x}_t^i, \boldsymbol{z}_t^i; \theta_t, \hat{h}) = \int_{\underline{\tau}_t}^{\overline{\tau}_t} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)} \|\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}(\boldsymbol{X}(\tau)|\boldsymbol{x}_t^i) - \theta_t(\boldsymbol{X}(\tau), \hat{h}_t(\boldsymbol{z}_t^i), \tau)\|^2 \mathrm{d}\tau, \quad (7)$$

where $(\underline{\tau}_t, \overline{\tau}_t)$ denotes early stopping for $(0, +\infty)$ and $\hat{h}_t(\boldsymbol{z}_t) = (\hat{h}(\boldsymbol{z}), \boldsymbol{z}_{t^c})$. The estimated score $\hat{\theta}_t(\boldsymbol{x}(\tau), \hat{h}(\boldsymbol{z}), \tau) = \arg\min_{\theta_t \in \Theta_t} L_t(\theta_t, \hat{h})$. We will use the neural network for $\Theta_t$.

**Neural network.** An $\mathbb{L}$-layer network $\Phi$ is defined by a composite function $\Phi(\boldsymbol{x}) = (\mathbf{A}_{\mathbb{L}}\sigma(\cdot) + \boldsymbol{b}_{\mathbb{L}}) \circ \cdots (\mathbf{A}_2\sigma(\cdot) + \boldsymbol{b}_2) \circ (\mathbf{A}_1\boldsymbol{x} + \boldsymbol{b}_1)$, where $\mathbf{A}_i \in \mathbb{R}^{d_{i+1} \times d_i}$ is a weight matrix and $\boldsymbol{b}_i \in \mathbb{R}^{d_{i+1}}$ is the bias of a linear transformation of the $i$-th layer, and $\sigma$ is the ReLU activation function, defined as $\sigma(\boldsymbol{x}) = \max(\boldsymbol{x}, 0)$. Then, the parameter space $\Theta_t$ is set as $\mathrm{NN}(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t)$ with $\mathbb{L}_t$ layers, a maximum width of $\mathbb{W}_t$, effective parameter number $\mathbb{S}_t$, the sup-norm $\mathbb{B}_t$, and parameter bound $\mathbb{E}_t$:

$$\mathrm{NN}(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t) = \{\Phi : \mathbb{R}^{d_{x_t}+d_{h_t}+1} \to \mathbb{R}^{d_{x_t}}, \max_{1 \le i \le \mathbb{L}_t} d_i \le \mathbb{W}_t, \sum_{i=1}^{\mathbb{L}_t} (\|\mathbf{A}_i\|_0 + \|\boldsymbol{b}_i\|_0) \le \mathbb{S}_t,$$
$$\sup_{\boldsymbol{x} \in \mathbb{R}^{d_{x_t}+d_{h_t}}} \|\Phi(\boldsymbol{x}, \tau)\|_\infty \le \mathbb{B}_t(\tau), \sup_\tau \mathbb{B}_t(\tau) \le \mathbb{B}_t, \max_{1 \le i \le \mathbb{L}_t} (\|\mathbf{A}_i\|_\infty, \|\boldsymbol{b}_i\|_\infty) \le \mathbb{E}_t\}, \quad (8)$$

where $d_{x_t}$ and $d_{h_t}$ denote the dimensions of $\boldsymbol{X}_t$ and the output of $h_t$, respectively, $\|\cdot\|_\infty$ denotes the maximum absolute value of the entries, and $\|\cdot\|_0$ denotes the number of nonzero entries.

**Conditional generation.** We approximate (5) by substituting the score $\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{z}_t}$ with its estimate $\hat{\theta}_t$, yielding:

$$\mathrm{d}\hat{\boldsymbol{V}}(\tau) = b_{\overline{\tau}_t - \tau}(\hat{\boldsymbol{V}}(\tau) + 2\hat{\theta}_t(\hat{\boldsymbol{V}}(\tau), \boldsymbol{z}_t, \overline{\tau}_t - \tau))\mathrm{d}\tau + \sqrt{2b_{\overline{\tau}_t - \tau}}\mathrm{d}W(\tau), \tau \in [0, \overline{\tau}_t - \underline{\tau}_t^*], \quad (9)$$

where we start the backward process from an initial state $\hat{\boldsymbol{V}}(0) \sim N(\boldsymbol{0}, \boldsymbol{I})$ and terminate the process at $\tau = \overline{\tau}_t - \underline{\tau}_t^*$ with $0 \le \underline{\tau}_t^* \le \underline{\tau}_t$, which will be determined later based on the density smoothness. We then utilize $\hat{\boldsymbol{V}}(\overline{\tau}_t - \underline{\tau}_t^*)$ as a generated sample to approximate $\boldsymbol{X}(0)$. The resulting conditional density estimate $\hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t}$ corresponds to the distribution $p_{\hat{\boldsymbol{v}}(\overline{\tau}_t - \underline{\tau}_t^*)|\boldsymbol{z}_t}$. Note that, because we apply early stopping at $\underline{\tau}_t$ during training of (7), we need to extend the reverse–time interval from $\tau \in [0, \overline{\tau}_t - \underline{\tau}_t]$ to $\tau \in [0, \overline{\tau}_t - \underline{\tau}_t^*]$. For the extended segment $\tau \in [\overline{\tau}_t - \underline{\tau}_t, \overline{\tau}_t - \underline{\tau}_t^*]$, we freeze the estimator, replacing $\hat{\theta}_t\big(\hat{\boldsymbol{v}}(\tau), \boldsymbol{z}_t, \overline{\tau}_t - \tau\big)$ by $\hat{\theta}_t\big(\hat{\boldsymbol{v}}(\overline{\tau}_t - \underline{\tau}_t), \boldsymbol{z}_t, \underline{\tau}_t\big)$.

Next, we introduce assumptions specific to diffusion models.

**Smooth class.** Let $\boldsymbol{\alpha}$ be multi-index with $|\boldsymbol{\alpha}| \le \lfloor r \rfloor$, where $\lfloor r \rfloor$ is the integer part of $r > 0$. A Hölder ball $\mathcal{C}^r(\mathcal{D}, \mathbb{R}^m, B)$ of radius $B$ with the degree of smoothness $r$ from domain $\mathcal{D}$ to $\mathbb{R}^m$ is defined by:

$$\left\{(g_1, \cdots, g_m) : \max_{1 \le l \le m} \left( \max_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor} \sup_{\boldsymbol{x}} |\partial^{\boldsymbol{\alpha}} g_l(\boldsymbol{x})| + \max_{|\boldsymbol{\alpha}| = \lfloor r \rfloor} \sup_{\boldsymbol{x} \ne \boldsymbol{y}} \frac{|\partial^{\boldsymbol{\alpha}} g_l(\boldsymbol{x}) - \partial^{\boldsymbol{\alpha}} g_l(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|^{r - \lfloor r \rfloor}} \right) < B \right\}.$$

**Assumption 5 (Target density)** *Assume that the true conditional density of $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$ is expressed as $p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\boldsymbol{x}|\boldsymbol{z}_t) = \exp(-c_4\|\boldsymbol{x}\|^2/2) \cdot f_t(\boldsymbol{x}, h_t^0(\boldsymbol{z}_t))$, where $f_t$ is a non-negative function and $c_4 > 0$ is a constant. Additionally, $f_t$ is lower bounded by a positive constant and belongs to*

*a Hölder ball* $\mathcal{C}^{r_t}(\mathbb{R}^{d_{x_t}} \times [0,1]^{d_{h_t}}, \mathbb{R}, B_t)$ *with* $r_t > 0$ *and* $B_t > 0$*, where* $d_{h_t}$ *is the dimension of* $(h(\boldsymbol{z}), z_{t^c})$.

Assumption 5 posits that the density ratio between the target and a Gaussian density falls within a Hölder class, bounded by upper and lower limits. This condition for $p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}$ has been cited in the literature for managing approximation errors in diffusion models, as shown in Fu et al. (2024); Oko et al. (2023). This condition, introduced in Fu et al. (2024), relaxes the restricted support condition in Oko et al. (2023), leading to some smooth characteristics of the score function used in Chen et al. (2023c,a,b).

Next, we present the error bounds for conditional diffusion generation via transfer learning. The network $\Theta_t$ and the stopping criteria are set with specified parameters:

$$\mathbb{L}_t = c_L \log^4 K, \mathbb{W}_t = c_W K \log^7 K, \mathbb{S}_t = c_S K \log^9 K, \log \mathbb{B}_t = c_B \log K, \log \mathbb{E}_t = c_E \log^4 K,$$
$$\log \underline{\tau}_t = -c_{\underline{\tau}_t} \log K, \overline{\tau}_t = c_{\overline{\tau}_t} \log K, \underline{\tau}^*_t = \mathbb{I}_{\{r_t \le 1\}} \underline{\tau}_t, \tag{10}$$

for sufficiently large constants $c_L - c_{\underline{\tau}_t}$, with $K$ a tuning parameter for its complexity, depending on the training size $n_t$, the smoothness degree $r_t$, and the dimensions of $\boldsymbol{X}_t$ and $h_t$, $d_{x_t}$ and $d_{h_t}$. Note that the configuration $\mathbb{W}_t \gg \mathbb{L}_t$ corresponds to a wide network.

The choice of $\underline{\tau}^*_t$ enables us to extend the backward SDE all the way to $\tau = \overline{\tau}_t$, thereby approximating $\boldsymbol{X}_t(0)$ by $\widehat{V}(0)$. This extension is valid under the smoothness condition $r_t > 1$, which ensures the desired continuity of the score function and allows its behavior over the tail interval $[0, \underline{\tau}]$ to be well represented by the score at $\underline{\tau}$.

To establish Theorems 1–4, we address several technical challenges involved in integrating source and target tasks. Specifically, we ensure model transferability for conditional generation and develop suitable latent representations for unconditional generation, while effectively controlling the source error as detailed in Assumption 3. Central to our approach is the utilization of the lower-dimensional manifold structure defined by the shared embedding condition (SEC), which significantly enhances distribution estimation accuracy. Additionally, we capitalize on structural properties specific to diffusion models to achieve our results.

**Theorem 1 (Conditional diffusion via transfer learning)** *Under Assumptions 1-3 and 5, with setting* (10) *by* $K \asymp n_t^{\frac{d_{x_t}+d_{h_t}}{d_{x_t}+d_{h_t}+2r_t}}$*, the generation error of conditional transfer diffusion models is*

$$\mathrm{E}_{\mathcal{D}_t,\mathcal{D}_s}\mathrm{E}_{\boldsymbol{z}_t}[\mathrm{TV}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{m_t} n_t + \varepsilon_s), \text{ if } r_t > 0;$$
$$\mathrm{E}_{\mathcal{D}_t,\mathcal{D}_s}\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{\frac{1}{2}}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{m_t} n_t + \varepsilon_s), \text{ if } r_t > 1.$$

*Here,* $m_t = \max(\frac{19}{2}, \frac{r_t}{2} + 1)$*,* $\asymp$ *means mutual boundedness, and* TV *and* $\mathcal{K}$ *denote the TV-norm and the KL divergence. The quantity* $d_{x_t}$ *is the dimension of the target data* $\boldsymbol{X}_t$*, while* $d_{h_t}$ *denotes the dimension of the condition representation induced by* $h_t$*.*

A formal non-asymptotic bound is provided in Theorem 20 in Appendix A.

To compare transfer conditional and non-transfer diffusion generations, we adopt the framework of the transfer model while omitting source learning. Specifically, we define the conditional

distribution without leveraging the source knowledge of $h$ from the estimated score function

$$\tilde{\theta}_t(\boldsymbol{x}_t, \tilde{h}_t(\boldsymbol{z}_t), \tau) = \underset{\theta_t \in \tilde{\Theta}_t, h \in \Theta_h}{\arg\min} \; L_t(\theta_t, h). \tag{11}$$

We impose a smoothness constraint on the neural network, explicitly defining:

$$\tilde{\Theta}_t = \theta_t \in \mathrm{NN}(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t, \lambda_t) : \mathbb{R}^{dx_t + dh_t + 1} \to \mathbb{R}^{dx_t}, \tag{12}$$

where this set represents a ReLU neural network analogous to equation (8), supplemented by an additional Hölder-norm bound :

$$\lambda_t = \sup_{\boldsymbol{x} \in \mathbb{R}^{d_x}, \boldsymbol{z} \neq \boldsymbol{z}', \tau \in [\underline{\tau}, \overline{\tau}]} \frac{\|\theta_t(\boldsymbol{x}, \boldsymbol{z}', \tau) - \theta_t(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty}{\|\boldsymbol{z} - \boldsymbol{z}'\|_2^{\alpha_t}}, \tag{13}$$

where $\alpha_t = r_t$ if $r_t \leq 1$ and $\alpha_t = \min(1, r_t - 1)$ if $r_t > 1$.

**Theorem 2 (Non-transfer conditional diffusion)** *Suppose $\tilde{\Theta}_t$ has the same configuration as $\Theta_t$ from Theorem 1 and an additional constraint on $\tilde{\Theta}_t$: $\lambda_t = c_\lambda$ for $r > 1$ and $\lambda_t = c_\lambda / \sigma_{\underline{\tau}}$ for $r \leq 1$, provided that $c_\lambda$ is sufficiently large. Under Assumption 5, the generation error of the non-transfer conditional diffusion model, adhering to the same stopping criteria from Theorem 1, is given by:*

$$\mathrm{E}_{\mathcal{D}_t} \mathrm{E}_{\boldsymbol{z}_t} [\mathrm{TV}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t} + d_{h_t} + 2r_t}} \log^{m_t} n_t + \varepsilon_t^h), \text{ if } r_t > 0;$$

$$\mathrm{E}_{\mathcal{D}_t} \mathrm{E}_{\boldsymbol{z}_t} [\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t} + d_{h_t} + 2r_t}} \log^{m_t} n_t + \varepsilon_t^h), \text{ if } r_t > 1.$$

*Here, $\varepsilon_t^h$ is the error rate for estimating $h_t(\boldsymbol{z}_t)$ as defined by* (59) *in Lemma 23.*

The results in Theorems 1 and 2 provide valuable insights into conditional generation.

**Dimension reduction via $h(\boldsymbol{z})$ and large pre-trained models.** Theorem 1 indicates that generation accuracy is influenced by the source error $\varepsilon_s$, which depends on the size of the pre-training sample $n_s$. Larger pre-training datasets effectively reduce $\varepsilon_s$. Specifically, using a diffusion model configured similarly for the source task yields $\varepsilon_s = O(n_s^{-\frac{r_s}{d_{x_s} + d_{h_s} + 2r_s}} \log^{m_s} n_s + \varepsilon_s^h)$, as derived in Lemma 21, where $r_s$ denotes the smoothness degree of $p_{\boldsymbol{x}_s|\boldsymbol{z}_s}$ and $m_s = \max(\frac{19}{2}, \frac{r_s}{2} + 1)$. When pre-training models are significantly large such that $n_s$ substantially exceeds $n_t$, the source error $\varepsilon_s$ becomes minimal, making the term $n_t^{-\frac{r_t}{d_{x_t} + d_{h_t} + 2r_t}} \log^{m_t} n_t$ the dominant factor influencing the accuracy of the transfer model. This rate generally surpasses the conventional estimation rate $n_t^{-\frac{r_t}{d_{x_t} + d_{z_t} + 2r_t}}$ for conditional densities over $(\boldsymbol{x}_t, \boldsymbol{z}_t)$ within $[0,1]^{d_{x_t} + d_{z_t}}$, as indicated in (Shen and Wong, 1994; Wong and Shen, 1995), assuming a smoothness degree of $r_t$. This improvement occurs because the effective dimension $d_{x_t} + d_{h_t}$ is smaller than $d_{x_t} + d_{z_t}$, attributed to the compact latent representation $h_t(\boldsymbol{z}_t)$ ($d_{h_t} \leq d_{z_t}$). Thus, $\varepsilon_t$ achieves an accelerated convergence rate through (nonlinear) dimensionality reduction.

**Transfer versus nontransfer conditional diffusion.** Building on Theorem 2, we contrast the excess–risk behavior of transfer and non–transfer conditional diffusion models to quantify the benefit of pre–training. When extensive source data yield a high-quality latent representation $\hat{h}$, the

target learner can regard $\hat{h}$ as fixed, thereby reducing the effective complexity of the target problem. Specifically, the excess risk of the transfer model satisfies whereas the nontransfer counterpart incurs with $\varepsilon_s \ll \varepsilon_t^h$ whenever the source pre-training set is large. Thus the leading term of the transfer bound is strictly smaller, formalizing the systematic performance gains achievable with transfer learning in data rich pre-training regimes. When the source and target tasks are weakly related, pre-training may instead degrade performance, a phenomenon known as negative transfer. Section 6 demonstrates both the gains and the hazards via a controlled numerical study. In practice, simple model-selection heuristics such as cross-validation can reliably detect and alleviate negative transfer (Hu and Zhang, 2023).

**Connection to Fu et al. (2024) for non-transfer conditional generation.** The study by Fu et al. (2024) investigates conditional diffusion generation without transfer learning with the density $p_{\boldsymbol{x}_t|z_t}$ over dimension $d_{x_t}$ and sample size $n_t$. They establish a TV-norm rate $O(n_t^{-\frac{r_t}{d_{x_t}+d_{z_t}+2r_t}} \log^{m_t} n_t)$ under their Assumption 3.3, paralleling Assumption 5 but on $\mathbb{R}^{d_{x_t}} \times [0,1]^{d_{z_t}}$ rather than a manifold $\mathbb{R}^{d_{x_t}} \times [0,1]^{d_{h_t}}$. This rate aligns with the minimax rate presented in Oko et al. (2023) for unconditional generation, excluding a logarithmic term. Conversely, Theorem 2 describes a KL divergence rate of $O(n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}})$, incorporating a logarithmic factor for non-transfer conditional generation when $r_t > 1$. Given that KL divergence provides a stronger metric compared to the TV-norm by Pinsker's inequality, the reduced dimension $d_{x_t} + d_{h_t}$ (generally smaller due to $d_{h_t} < d_{z_t}$ and the nonlinear manifold structure $h(\boldsymbol{z}_t)$) enhances convergence. Thus, leveraging latent representations guided by SEC typically leads to accelerated convergence rates.

### 3.3 Unconditional diffusion via transfer learning

For an unconditional generation of $\boldsymbol{U}$, we use the diffusion model to learn the latent distribution of $\boldsymbol{U}$ from the target data $\{\boldsymbol{u}_s^i\}_{i=1}^{n_s}$, with $\boldsymbol{X}(0) = \boldsymbol{U}$ in (4). Then, the empirical score matching loss in (6) $L_u$ is $L_u(\theta_u) = \sum_{i=1}^{n_s} \int_{\underline{\tau}}^{\bar{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)} \|\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}(\boldsymbol{X}(\tau)|\boldsymbol{u}_s^i) - \theta_u(\boldsymbol{X}(\tau),\tau)\|^2 \mathrm{d}\tau$, where $\theta_u \in \Theta_u = \mathrm{NN}(\mathbb{L}_u, \mathbb{W}_u, \mathbb{S}_u, \mathbb{B}_u, \mathbb{E}_u)$. The estimated score function for the target task is $\hat{\theta}_u = \arg\min_{\theta_u \in \Theta_U} L_U(\theta_u)$. Then, we set $l_{g_t}(\boldsymbol{u}, \boldsymbol{x}; g_t)$ as the reconstruction squared $L_2$ loss and minimize the loss to yield $\hat{g}_t = \arg\min_{g_t \in \Theta_{g_t}} \sum_{i=1}^{n_t} l_{g_t}(\boldsymbol{u}_t^i, \boldsymbol{x}_t^i; g_t) = \arg\min_{g_t \in \Theta_{g_t}} \sum_{i=1}^{n_t} \|g_t(\boldsymbol{u}_t^i) - \boldsymbol{x}_t^i\|_2^2$, where $\Theta_{g_t} = \mathrm{NN}(\mathbb{L}_g, \mathbb{W}_g, \mathbb{S}_g, \mathbb{B}_g, \mathbb{E}_g)$. Finally, $P_{\boldsymbol{x}_t}$ is estimated by $\hat{P}_{\boldsymbol{x}_t} = \hat{P}_u(\hat{g}_t^{-1})$.

Next, we introduce specific assumptions for unconditional generation.

**Assumption 6 (Smoothness of $g_t^0$)** *Assume that $g_t^0 \in \mathcal{C}^{r_g}(\mathbb{R}^{d_u}, [0,1]^{d_{x_t}}, B_g)$ with radius $B_g > 0$ and degree of smoothness $r_g > 0$.*

The network class $\Theta_g$ is set with specified parameters:

$$\begin{aligned} &\mathbb{L}_g = c_L L \log L, \mathbb{W}_g = c_W W \log W, \mathbb{S}_g = c_S W^2 L \log^2 W \log L, \\ &\mathbb{B}_g = c_B, \log \mathbb{E}_g = c_E \log(WL). \end{aligned} \tag{14}$$

Here, $c_L, c_W, c_S, c_B, c_E$ are sufficiently large constants, and $(W, L)$ are the parameters to control the complexity of the estimator class and dependent on $d_u$, $n_t$ and $r_t$. This configuration allows for flexibility in the network's architecture, enabling the use of either wide or deep structures.

**Theorem 3 (Unconditional diffusion via transfer learning)** *Under Assumptions 1, 4 and 6, with setting* (14) *by* $WL \asymp n_t^{\frac{d_u}{2(d_u+2r_g)}}$*, the generation error for unconditional transfer diffusion models in the Wasserstein distance is*

$$\mathrm{E}_{\mathcal{D}_t,\mathcal{D}_s}\mathcal{W}(P^0_{\boldsymbol{x}_t},\hat{P}_{\boldsymbol{x}_t}) = O(n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t + \varepsilon_s^u),$$

*where* $m_g = \max(\frac{5}{2},\frac{r_g}{2})$.

To compare transfer and non-transfer diffusion generation, we define the non-transfer estimation as

$$\tilde{P}_{\boldsymbol{x}_t} = \tilde{P}_u(\hat{g}_t^{-1}),$$

where $\tilde{P}_u$ is estimated using diffusion models characterized by the score function $\tilde{\theta}_u$. Specifically, $\tilde{\theta}_u$ is obtained by substituting the sample set in $L_u$ with $\{\boldsymbol{u}_t^i\}_{i=1}^{n_t}$.

**Theorem 4 (Non-transfer via unconditional diffusion)** *Suppose there exists a sequence* $\varepsilon_t^u$ *indexed by* $n_t$ *such that* $n_t^{1-\xi}(\varepsilon_t^u)^2 \to \infty$ *as* $n_s \to \infty$ *and* $P(\rho_u(\theta_u^0,\tilde{\theta}_u) \geq \varepsilon) \leq \exp(-c_3 n_t^{1-\xi}\varepsilon^2)$ *for any* $\varepsilon \geq \varepsilon_t^u$ *and some constants* $c_3, \xi > 0$*. Under Assumption 6 and the same settings for* $\Theta_{g_t}$ *in Theorem 3, the generation error of the non-transfer unconditional diffusion model is*

$$\mathrm{E}_{\mathcal{D}_t}\mathcal{W}(P^0_{\boldsymbol{x}_t},\tilde{P}_{\boldsymbol{x}_t}) = O(n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t + \varepsilon_t^u).$$

For the unconditional case, we derive the generation error expressed in the Wasserstein distance $\mathcal{W}(\hat{P}_{\boldsymbol{x}},P^0_{\boldsymbol{x}}) = \sup_{\|f\|_{Lip}\leq 1}|\int f(\boldsymbol{x})(\mathrm{d}\hat{P}_{\boldsymbol{x}} - \mathrm{d}P^0_{\boldsymbol{x}})|$, where $\|f\|_{Lip} \leq 1$ indicates that $f$ is within the 1-Lipschitz class. The Wasserstein distance is appropriate for scenarios with dimension reduction, where the input dimension $d_u$ for $g_t$ is less than the output dimension $d_{x_t}$. In contrast, the KL divergence or the TV-norm may not be appropriate when $g_t$ is not invertible.

To understand the significance of this result in unconditional generation, we explore the following aspects:

**Dimension reduction via latent structures $\boldsymbol{U}$.** Theorem 3 shows the error rate for unconditional diffusion generation of density $p_{\boldsymbol{x}_t}$ is $O(n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t + \varepsilon_s^u)$ with $m_g = \max(\frac{5}{2},\frac{r_g}{2})$, where $\varepsilon_s^u = n_s^{-\frac{r_u}{d_u+2r_u}}\log^{m_u} n_s$ as established in Lemma 28, with $m_u = \max(\frac{19}{2},\frac{r_u}{2}+1)$ and $r_u$ denoting the smoothness degree of $p_{\boldsymbol{u}}$. Given sufficient pre-training data, particularly when $n_s >> n_t$, $\varepsilon_s^u$ becomes negligible, leaving $n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t$ as the primary determinant of accuracy. This rate differs from the $n_t^{-\frac{r_g}{d_{x_t}+2r_g}}$ rate for density estimation of $p_{\boldsymbol{x}_t}$, where $r_g$ represents the smoothness of $p_{\boldsymbol{x}_t}$ over $[0,1]^{d_{x_t}}$. The improved rate $\varepsilon_t$ results from dimension reduction since $d_u < d_{x_t}$, facilitated by the latent structures $\boldsymbol{U}$, leading to enhanced generation accuracy relative to methods lacking dimension reduction.

**Advantages of transfer learning.** Estimating the transferred latent distribution $P_{\boldsymbol{u}}$ from the source task significantly improves generative performance in the target task. A comparative analysis reveals a notable difference between transfer and non-transfer models in terms of generation error rates: $n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t + \varepsilon_s^u$ for the transfer model and $n_t^{-\frac{r_g}{d_u+2r_g}}\log^{m_g} n_t + \varepsilon_t^u$ for the non-transfer model, with $\varepsilon_j^u$; $j \in \{s,t\}$ representing the generation errors in source and target learning,

respectively. Here, $\varepsilon_s^u$ is significantly lower than $\varepsilon_t^u$ when the source model is extensively pre-trained, evidenced by $n_s \gg n_t$. This highlights the efficiency and effectiveness of transfer learning when leveraging well-prepared source models.

**Comparison with Oko et al. (2023) and Chen et al. (2023b) in non-transfer unconditional generation.** The work by Oko et al. (2023) on unconditional diffusion generation for a $d_{x_t}$-dimensional density $p_{\boldsymbol{x}_t}$ with a sample size $n_t$ sets a TV-norm upper bound at $O(n_t^{-\frac{r_t}{d_{x_t}+2r_t}} \log^{\frac{5d_{x_t}+8r_t}{2d_{x_t}}} n_t)$ with $r_t$ indicating the smoothness degree of $p_{\boldsymbol{x}_t}$. This rate, nearly minimax, requires the density $p_{\boldsymbol{x}_t}$ to be smoothly within a Besov ball $\mathbb{B}_{p,q}^r([0,1]^{d_{x_t}})$ and assumes infinite smoothness near boundaries as per Assumption 2.4. Meanwhile, Chen et al. (2023b) establishes a TV-norm upper bound at $O(n_t^{-\frac{1}{2(d_{x_t}+5)}})$ for low-dimensional linear subspaces with Lipschitz continuous score functions, a rate slower than the previous and suboptimal when $r_t = 2$. In contrast, Theorem 4 and Theorem 27 derive a Wasserstein distance bound of $O(n_t^{-\frac{r_g}{d_u+2r_g}} \log^{m_g} n_t + n_t^{-\frac{r_u}{d_u+2r_u}} \log^{m_u} n_t)$. With equal smoothness degrees ($r = r_g = r_u$), this bound becomes $O(n_t^{-\frac{r_t}{d_u+2r_t}} \log^{\max(\frac{19}{2}, \frac{r_t}{2}+1)} n_t)$. Given $d_u < d_{x_t}$ due to dimension reduction, this suggests a faster rate despite different metrics.

## 4. Normalizing flows

### 4.1 Coupling

Normalizing flows transform a random vector $\boldsymbol{X}$ into a base vector $\boldsymbol{V}$ with known density $p_{\boldsymbol{v}}$, through a diffeomorphic mapping $T(\boldsymbol{X})$, which is invertible and differentiable. The composition of these mappings, $T = \phi_K \circ \ldots \circ \phi_1$, with each $\phi_j$ modeled by a neural network, estimates $T$. The density of $\boldsymbol{X}$ is expressed as $p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{v}}(T(\boldsymbol{x})) \left|\det \frac{\partial T(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|$, with the determinant indicating the volume change under $T$. The maximum likelihood approach is used to estimate $T$, enabling the generation of new $\boldsymbol{X}$ samples by inverting $T$ on samples from $p_{\boldsymbol{v}}$.

**Coupling flows.** Coupling flows partition $\boldsymbol{x}$ into two parts $\boldsymbol{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$. Each flow employs a transformation $\phi_j(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1, q(\boldsymbol{x}_2, \omega_j(\boldsymbol{x}_1)))$; $j = 1, \ldots, K$. The function $q$ modifies $\boldsymbol{x}_2$ based on the output of $\omega$, where $q$ ensures that $\phi_j$ is invertible.

**Conditional coupling flows.** To add an additional condition input of $\boldsymbol{z}$ to the coupling layer, we adjust $\phi_j(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{z}) = (\boldsymbol{x}_1, q(\boldsymbol{x}_2, \omega(\boldsymbol{x}_1, \boldsymbol{z})))$.

### 4.2 Conditional flows via transfer learning

This section constructs coupling flows to model the conditional density $p_{\boldsymbol{x}_t|\boldsymbol{z}_t}$. We use the three-layer coupling flows defined in (65) with $q(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x} + \boldsymbol{y}$, denoted by $\mathrm{CF}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E}, \lambda)$, where $\phi_1$ is defined by a neural network $\omega_1 \in \mathrm{NN}_t(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E}, \lambda)$, with the maximum depth $\mathbb{L}$, the maximum width $\mathbb{W}$, the number of effective parameters $\mathbb{S}$, the supremum norm of the neural network $\mathbb{B}$, the parameter bound $\mathbb{E}$ and the Lipschitz norm of the neural network $\lambda$; $\phi_2$ is a permutation mapping; $\phi_3$ is defined by $\omega_3 = -\omega_1^{-1}$. For the target learning task, we define the parameter space as $\Theta_t = \mathrm{CF}_t(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t, \lambda_t)$.

$$\Theta_t = \{\theta_t(\boldsymbol{x}, \hat{h}_t(\boldsymbol{z})) : [\phi_3 \circ \phi_2 \circ \phi_1((\boldsymbol{x}, \boldsymbol{0}), \hat{h}_t(\boldsymbol{z}))]^{1:d_{x_t}}, \omega_1 \in \mathrm{NN}_t(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t, \lambda_t)\}. \quad (15)$$

Here, $(\boldsymbol{x}, \mathbf{0})$ is a zero padding vector of dimension $2d_{x_t}$ and $[\cdot]^{1:d_{x_t}}$ denotes the first $d_{x_t}$ elements. Given a target training sample $\{\boldsymbol{x}_t^i, \boldsymbol{z}_t^i\}_{i=1}^{n_t}$ and an estimated latent structures $\hat{h}$ derived from the pre-trained flow model based on a source training sample $(\boldsymbol{x}_s^i, \boldsymbol{z}_s^i)_{i=1}^{n_s}$, we define the loss function as $l_t = -\log p_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\boldsymbol{x}_t, \boldsymbol{z}_t; \theta_t, \hat{h})$, and then estimate the target flow by minimizing the negative log-likelihood as follows:

$$\hat{T}_t = \hat{\theta}_t(\boldsymbol{x}_t, \hat{h}_t(\boldsymbol{z}_t)) = \arg\min_{\theta_t \in \Theta_t} \sum_{i=1}^{n_t} -\log p_{\boldsymbol{v}}(\theta_t(\boldsymbol{x}_t^i, \hat{h}_t(\boldsymbol{z}_t^i))) - \log \left|\det(\nabla_{\boldsymbol{x}} \theta_t(\boldsymbol{x}_t^i, \hat{h}_t(\boldsymbol{z}_t^i)))\right|.$$

Here $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}) = \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}$. This allows for conditional generation via flows using $\boldsymbol{x}_t = \hat{T}_t^{-1}(\boldsymbol{v}, \boldsymbol{z}_t)$ and estimating the conditional target density as $\hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\boldsymbol{x}, \boldsymbol{z}) = p_{\boldsymbol{v}}(\hat{T}_t(\boldsymbol{x}, \boldsymbol{z})) \left|\det \nabla_{\boldsymbol{x}} \hat{T}_t(\boldsymbol{x}, \boldsymbol{z})\right|$.

We make several assumptions about the distribution of $\boldsymbol{X}_t$ given $\boldsymbol{Z}_t$.

**Assumption 7 (Transformation)** *Suppose that there exists $T_t^0(\boldsymbol{x}, \boldsymbol{z}) = \theta_t^0(\boldsymbol{x}, h_t^0(\boldsymbol{z}))$ such that $\boldsymbol{V} = T_t^0(\boldsymbol{X}_t, \boldsymbol{Z}_t)$. The true transform $\theta_t^0(\boldsymbol{x}_t, h_t^0(\boldsymbol{z}_t))$ and its inverse belong to a Hölder ball $\mathcal{C}^{r_t+1}([0,1]^{d_{x_t}+d_{h_t}}, [0,1]^{d_{x_t}}, B_t)$ of radius $B_t > 0$, while $|\det \nabla_{\boldsymbol{x}} \theta_t^0|$ is lower bounded by some positive constant. Moreover, the base vector $\boldsymbol{V}$ has a known smooth density in $\mathcal{C}^{\infty}([0,1]^{d_{x_t}}, \mathbb{R}, B_v)$ with a lower bound.*

This condition, a generalized version of the bi-Lipschitz condition used in Jin et al. (2024), enables the constructed invertible network to approximate $T^0$ while satisfying invertibility. The smoothness condition is critical for controlling the approximation error associated with the Jacobian matrix during the approximation process.

The network class $\Theta_t$ is set with specified parameters:

$$\begin{aligned} &\mathbb{L}_t = c_L L \log L, \mathbb{W}_t = c_W W \log W, \mathbb{S}_t = c_S W^2 L \log^2 W \log L, \\ &\mathbb{B}_t = c_B, \log \mathbb{E}_t = c_E \log(WL), \lambda_t = c_\lambda. \end{aligned} \tag{16}$$

To derive Theorems 5–8, we address essential technical challenges related to constructing invertible neural networks for flow approximation, as well as those previously outlined for diffusion models. Furthermore, we capitalize on the distinct structural properties inherent to flow models.

**Theorem 5 (Conditional flows via transfer learning)** *Under Assumptions 1-3 and 7, with $WL \asymp n_t^{\frac{d_{x_t}+d_{h_t}}{2(d_{x_t}+d_{h_t}+2r_t)}}$ in the setting* (16)*, the generation error of conditional transfer flow models is*

$$\mathrm{E}_{\mathcal{D}_t, \mathcal{D}_s} \mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{\frac{5}{2}} n_t + \varepsilon_s).$$

This theorem introduces the first results delineating the generation accuracy bounds for flow models, particularly in the context of transfer learning. It extends the existing approximation literature for invertible neural networks, as detailed in Jin et al. (2024) on approximation. Importantly, we establish an error bound that simultaneously addresses the mapping and the Jacobian matrix. Although these results are theoretically significant, practical implementation of such neural networks with constraints as per (71) may require considerable effort, as discussed in the appendix.

Regarding the non-transfer case, the transformation function $T$ is estimated by $\tilde{\theta}_t(\boldsymbol{x}_t, \tilde{h}_t(\boldsymbol{z}_t)) = \arg\min_{\theta_t \in \Theta_t, h \in \Theta_h} \sum_{i=1}^{n_t} -\log p_{\boldsymbol{x}_t|\boldsymbol{z}_t}(\boldsymbol{x}_t^i, \boldsymbol{z}_t^i; \theta_t; h)$.

**Theorem 6 (Non-transfer conditional flows)** *Under Assumption 7 and the same configurations in Theorem 5, the generation error of the non-transfer conditional flow model is*

$$\mathrm{E}_{\mathcal{D}_t}\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] = O(n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{\frac{5}{2}} n_t + \varepsilon_t^h).$$

*Here, $\varepsilon_t^h$ is the error rate for estimating $h_t(\boldsymbol{z}_t)$ as defined by* (59) *in Lemma 23.*

**Conditional flow generation: transfer versus non-transfer models.** The conditional flow generation rate via transfer learning is $n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{5/2} n_t + \varepsilon_s$. For the pre-trained (source) flow model, $\varepsilon_s$ is expressed as $n_s^{-\frac{r_s}{d_{x_s}+d_h+2r_s}} \log^{5/2} n_s + \varepsilon_s^h$, by Lemma 36, where $\varepsilon_s^h$ satisfies the integral entropy condition in Lemma 23. In comparison to the non-transfer model's error bound of $n_t^{-\frac{r}{d_{x_t}+d_h+2r}} \log^{5/2} n_t + \varepsilon_t^h$, the transfer flow model exhibits superior performance in contexts where the latent structure complexity is substantial and the source sample size $n_s$ is considerably larger than the target sample size $n_t$, such that $\varepsilon_s^h \ll \varepsilon_t^h$.

### 4.3 Unconditional flows via transfer learning

In unconditional generation, we estimate the latent distribution $p_u$ using coupling flows $\Theta_u = \mathrm{CF}_u(\mathbb{L}_u, \mathbb{W}_u, \mathbb{S}_u, \mathbb{B}_u, \mathbb{E}_u, \lambda_u)$, based on a source sample $\{\boldsymbol{u}_s^i\}_{i=1}^{n_s}$. The model $\theta_u$ is obtained by:

$$\hat{\theta}_u = \arg\min_{\theta_u \in \Theta_u} \sum_{i=1}^{n_s} l_u(\boldsymbol{u}_s^i; \theta_u) = \arg\min_{\theta_u \in \Theta_u} \sum_{i=1}^{n_s} -\log p_{\boldsymbol{v}}(\theta_u(\boldsymbol{u}_s^i)) - \log\left|\det \nabla\theta_u(\boldsymbol{u}_s^i)\right|.$$

Then, we minimize the reconstruction error loss on a target sample $\{\boldsymbol{u}_t^i, \boldsymbol{x}_t^i\}_{i=1}^{n_t}$ to obtain $\hat{g}_t = \arg\min_{g_t \in \Theta_{g_t}} \sum_{i=1}^{n_t} \|g_t(\boldsymbol{u}_t^i) - \boldsymbol{x}_t^i\|_2^2$, where $\Theta_{g_t} = \mathrm{NN}_g(\mathbb{L}_g, \mathbb{W}_g, \mathbb{S}_g, \mathbb{B}_g, \mathbb{E}_g)$. The estimation for $P_{\boldsymbol{x}_t}$ can be derived as $\hat{P}_{\boldsymbol{x}_t} = \hat{P}_u(\hat{g}_t^{-1})$. Next, we restate Assumption 6 for the true function $g_t^0$.

**Assumption 8 (Smoothness of $g_t^0$)** *Assume that $g_t^0 \in \mathcal{C}^{r_g}([0,1]^{d_u}, [0,1]^{d_{x_t}}, B_g)$ with radius $B_g > 0$ and degree of smoothness $r_g > 0$.*

The network class $\Theta_g$ is set with specified parameters:

$$\begin{aligned} &\mathbb{L}_g = c_L L \log L, \mathbb{W}_g = c_W W \log W, \mathbb{S}_g = c_S W^2 L \log^2 W \log L, \\ &\mathbb{B}_g = c_B, \log \mathbb{E}_g = c_E \log(WL). \end{aligned} \tag{17}$$

**Theorem 7 (Unconditional flows via transfer learning)** *Under Assumptions 1, 4 and 8, with setting* (17) *by $WL \asymp n_t^{\frac{d_u}{2(d_u+2r_g)}}$, the error for unconditional flow generation via transfer learning is,*

$$\mathrm{E}_{\mathcal{D}_t, \mathcal{D}_s}\mathcal{W}(P^0_{\boldsymbol{x}_t}, \hat{P}_{\boldsymbol{x}_t}) = O(n_t^{-\frac{r_g}{d_u+2r_g}} \log^{\frac{5}{2}} n_t + \varepsilon_s^u).$$

In the non-transfer case, we define the estimation $\tilde{p}_u$ with $\tilde{\theta}_u$ using the target data $\{\boldsymbol{u}_t^i\}_{i=1}^{n_t}$. The distribution estimation for $\boldsymbol{X}_t$ is given by $\tilde{P}_{\boldsymbol{x}_t} = \tilde{P}_u(\hat{g}_t^{-1})$.

**Theorem 8 (Non-transfer unconditional flows)** *Suppose there exists a sequence $\varepsilon_t^u$ indexed by $n_t$ such that $n_t^{1-\xi}(\varepsilon_t^u)^2 \to \infty$ as $n_s \to \infty$ and $P(\rho_u(\theta_u^0, \tilde{\theta}_u) \geq \varepsilon) \leq \exp(-c_3 n_t^{1-\xi}\varepsilon^2)$ for any $\varepsilon \geq \varepsilon_t^u$ and some constants $c_3, \xi > 0$. Under Assumption 8 and the same configurations of Theorem 7, the error in non-transfer flow generation is*

$$\mathrm{E}_{\mathcal{D}_t}\mathcal{W}(P^0_{\boldsymbol{x}_t}, \tilde{P}_{\boldsymbol{x}_t}) = O(n_t^{-\frac{rg}{d_u+2rg}} \log^{\frac{5}{2}} n_t + \varepsilon_t^u).$$

**Comparison of diffusion and flows**

**Generation accuracy.** Flow models learn probability densities by transforming the target distribution to a known base distribution. On the other hand, Gaussian diffusion models learn the target density by bridging the standard Gaussian and target distributions through score matching. Both approaches utilize round-trip processes and achieve similar error rates in both conditional and unconditional generation, as discussed in Theorems 2, 4, 6 and 8, except for the evaluation metrics and network architectures.

**Limit of assumptions.** These models require specific assumptions about the target density. Notably, coupling flows operate under slightly less stringent conditions than those for diffusion models. The assumptions for diffusion models primarily relate to the complexities of matching the score function and the necessity of approximating the target density by its smoothed version by Gaussian noise.

**Network architecture.** Theorem 1 suggests that diffusion generation requires a wide network for sufficient approximation. In contrast, Theorem 5 indicates that the architecture for flow models can vary in either network depth or width, offering greater flexibility. However, designing flows for specific tasks can be computationally challenging.

## 5. Core proof strategy

The proofs for our main theoretical results must overcome three principal technical hurdles:

**Propagating error from source to target under the SEC condition.** We first derive a model-agnostic risk-decomposition bound that requires no distributional assumptions. It splits the target risk into three components: (i) the source error, (ii) an approximation error, and (iii) an estimation error. This result hinges on extending the classical convergence theorem of Shen and Wong (1994) to the conditional setting needed for transfer learning.

**Controlling approximation and estimation errors in diffusion models.** Adopting the Gaussian-control framework of Fu et al. (2024), we inherit their guarantees on the score-matching loss, which recover the known total-variation bounds. To upgrade these guarantees to Kullback–Leibler divergence, we perform a refined analysis of the tail segment ($[0, \underline{t}]$) of the reverse diffusion, converting score-matching accuracy into KL guarantees.

**Simultaneously approximating mappings and their derivatives in flow models.** We introduce two key tools: (i) a new coupling-structure argument that links errors in function values and gradients, and (ii) the simultaneous-approximation theory for ReQU-activated neural networks of Belomestny et al. (2023), which provides joint control over both approximation and derivative errors.

### 5.1 Theory for transferred risk control

This section establishes a theoretical framework to quantify the excess risk associated with estimating high-dimensional distributions that lie on lower-dimensional manifolds, both in the context of

transfer learning and its non-transfer counterpart. This setting significantly differs from classical supervised transfer learning frameworks due to the potential degeneracy inherent in high-dimensional distributions. Leveraging the unique structural properties of diffusion and flow models, we derive generation error bounds directly linked to the excess risk in estimation accuracy. We specify precise conditions regarding the variance, sub-Gaussian properties of the loss functions, and the mechanisms governing knowledge transfer between source and target tasks.

As mentioned in Section 2, we parametrize the loss $l_j$ by $\gamma_j = (\theta_j, h) \in \Gamma_j = \Theta_j \times \Theta_h$ for conditional generation; $j \in \{s, t\}$. This setup includes task-specific parameters $\theta_j$'s and shared latent parameters $h$. In what follows, $\mathrm{E}$ and $\mathrm{Var}$ represent the expectation and variance concerning the associated randomness. We make the following assumptions.

**Assumption 9** *(Variance) There exist constants $c_{vj} > 0$ such that for all small $\varepsilon > 0$,*

$$\sup_{\{\rho_j(\gamma_j^0, \gamma_j) \le \varepsilon : \gamma_j \in \Gamma_j\}} \mathrm{Var}(l_j(\cdot; \gamma_j) - l_j(\cdot; \gamma_j^0)) \le c_{vj}\varepsilon^2; \quad j \in \{s, t\},$$

*where $\rho_j^2(\gamma_j^0, \gamma_j) = \mathrm{E}[l_j(\cdot; \gamma_j) - l_j(\cdot; \gamma_j^0)]$ is the excess risk; $j \in \{s, t\}$.*

This assumption specifies a connection between the variance and the mean of the loss function.

The assumption next ensures that the loss function exhibits an exponential tail behavior. A random variable from a function class $\mathcal{F}$ is said to satisfy Bernstein's condition with parameter $b > 0$ if $\sup_{f \in \mathcal{F}} \mathrm{E}(|f(\boldsymbol{X}) - \mathrm{E}[f(\boldsymbol{X})]|^k) \le \frac{1}{2} k! v b^{k-2}$ and $k \ge 2$, where $\sup_{f \in \mathcal{F}} \mathrm{Var}(f(X)) \le v^2$.

**Assumption 10** *(Bernstein) Assume that $l_j(\cdot; \gamma_j) - l_j(\cdot; \gamma_j^0)$ satisfies Bernstein's condition for $\gamma_j \in \Gamma_j$ with parameter $0 < c_{bj} < 4c_{vj}$; $j \in \{s, t\}$, where $c_{vj}$ is defined in Assumption 9.*

To measure the complexity of a function class, we define the bracketing $L_2$-metric entropy of $\mathcal{F} = \{f\}$. For any $u > 0$, a finite set of function pairs $(f_j^L, f_j^U)_{j=1}^N$ constitutes a $u$-bracketing of $\mathcal{F}$ if, for each $j = 1, \ldots, N$, the condition $[\mathrm{E}(f_j^L(\cdot) - f_j^U(\cdot))^2]^{1/2} \le u$ is satisfied. Furthermore, for any function $f \in \mathcal{F}$, there exists an index $j$ such that $f_j^L \le f \le f_j^U$. The bracketing $L$-metric entropy of $\mathcal{F}$, denoted as $H_B(\cdot, \mathcal{F})$, is then defined by the logarithm of the cardinality of the smallest-sized $u$-bracketing of $\mathcal{F}$.

Next, we derive some results for the transfer and non-transfer generation error under the conditional SEC setting. Let $\delta_j(h) = \inf_{\gamma_j = (\theta_j, h) : \theta_j \in \Theta_j} \rho_j^2(\gamma_j^0, \gamma_j)$ represent the approximation error given the true shared parameter $h$. Then $\delta_j(h^0)$ renders as an upper bound for the overall approximation error for the source and target tasks when $h^0 \in \Theta_h$ and $j \in \{s, t\}$. Define $\mathcal{F}_s = \{l_s(\cdot; \gamma_s) - l_s(\cdot; \pi\gamma_s^0) : \gamma_s \in \Gamma_s\}$ and $\mathcal{F}_t(h) = \{l_t(\cdot; \gamma_t) - l_t(\cdot; \pi\gamma_t^0) : \gamma_t = (\theta_t, h), \theta_t \in \Theta_t\}$ as candidate loss function classes induced by the parameters, where $\pi\gamma_j^0 \in \Gamma_j$ is an approximate point of $\gamma_j^0$ within $\Gamma_j$.

Theorem 9 establishes the excess risk associated with estimating high-dimensional distributions in the context of transfer learning.

**Theorem 9 (Transfer Learning)** *Under Assumptions 1, 2, 9, and 10, the error $\varepsilon_s$ and $\varepsilon_t$ satisfy: $\varepsilon_s \ge \sqrt{2\delta_s(h^0)}$, and $\varepsilon_t \ge \sqrt{2\delta_t(h^0)}$, as well as the following entropy bounds:*

$$\int_{k_s\varepsilon_s^2/16}^{4c_{vs}^{1/2}\varepsilon_s} H_B^{1/2}(u, \mathcal{F}_s)du \le c_{h_s} n_s^{1/2} \varepsilon_s^2, \text{ and } \int_{k_t\varepsilon_t^2/16}^{4c_{vt}^{1/2}\varepsilon_t} \sup_{h \in \Theta_h} H_B^{1/2}(u, \mathcal{F}_t(h))du \le c_{h_t} n_t^{1/2} \varepsilon_t^2, \quad (18)$$

*where $k_j$ and $c_{h_j}$ are constants depending on $c_{vj}$ and $c_{bj}$ for $j = s, t$. Then, the error bound for the target generation via $\hat{\gamma}_t$ is given by:*

$$P\left(\rho_t(\gamma_t^0, \hat{\gamma}_t) \geq x(\varepsilon_t + \sqrt{3c_1}\varepsilon_s)\right) \leq \exp(-c_{e_t} n_t (x\varepsilon_t)^2) + \exp(-c_{e_s} n_s (x\varepsilon_s)^2), \tag{19}$$

*for any $x \geq 1$ and some positive constants $c_{e_s}$ and $c_{e_t}$. This implies that $\rho_t(\gamma_t^0, \hat{\gamma}_t) = O_p(\varepsilon_t + \varepsilon_s)$ provided that $\min(n_t\varepsilon_t^2, n_s\varepsilon_s^2) \to \infty$ as $\min(n_s, n_t) \to \infty$.*

The target generation error errors $\varepsilon_t$ and $\varepsilon_s$, established in Theorem 9, conform to the entropy constraints detailed in (18) and are subject to the lower bounds $\varepsilon_j \geq \sqrt{2\delta_j(h^0)}$, reflecting the variance-bias trade-off inherent in a learning process. Notably, due to the utilization of a large pre-trained model, $\varepsilon_s$ is significantly less than $\varepsilon_t$, making $\varepsilon_t$ the predominant factor in transfer learning scenarios. Moreover, Theorem 10 suggests that the generation error in transfer learning is relatively smaller than its non-transfer counterpart, owning to shared parameters learned from the source task. The shared learning facilitates dimension reduction, thus enhancing the generation accuracy.

In the absence of knowledge transfer, let $\tilde{\gamma}_t = \arg\min_{\gamma_t \in \Gamma_t} L_t(\gamma_t)$ be the counterpart of $\hat{\gamma}_t$. Define the function class $\tilde{\mathcal{F}}_t = \{l(\cdot;\gamma_t) - l(\cdot;\pi\gamma_t^0)) : \gamma_t \in \Gamma_t\}$, where $\pi\gamma_t^0 \in \Gamma_t$ is an approximating point of $\gamma_t^0$ within $\Gamma_t$.

Theorem 10 establishes the excess risk associated with estimating high-dimensional distributions in the context of non-transfer learning.

**Theorem 10 (Non-transfer)** *Under Assumptions 9–10, let $\tilde{\varepsilon}_t$ satisfy the conditions*

$$\int_{k_t\tilde{\varepsilon}_t^2/16}^{4c_{vt}^{1/2}\tilde{\varepsilon}_t} H_B^{1/2}(u, \tilde{\mathcal{F}}_t)du \leq c_{h_t} n_t^{1/2} \tilde{\varepsilon}_t^2, \quad \tilde{\varepsilon}_t \geq \sqrt{2\delta_t(h^0)}. \tag{20}$$

*Then, the probability bound for target generation is as follows:*

$$P\Big(\rho(\gamma_t^0, \tilde{\gamma}_t) \geq \tilde{\varepsilon}_t\Big) \leq \exp(-c_{e_t} n_t \tilde{\varepsilon}_t^2), \tag{21}$$

*for some constant $c_{e_t} > 0$.*

Theorem 10 demonstrates a situation where the generation error of a transfer model remains lower compared to its non-transfer counterpart. This situation is evident when the source training size $n_s$ is significantly higher than that of the target $n_t$. Consequently, leveraging the shared representations for the source and target in transfer learning leads to a lower generation error than the non-transfer approach.

## 6. Numerical experiments

### 6.1 Simulations

We conduct simulation studies to demonstrate the effectiveness of transfer generative learning and validate our theoretical findings. Two simulation models are considered for the conditional and unconditional cases:

**Conditional generation.** The source variable is defined as $X_s = \sin Z_1 + \cos Z_2 + Z_3^2 + e_s$, where $\boldsymbol{Z} = (Z_1, \ldots, Z_3) \sim \text{Unif}(-2, 2)$ is a random vector, and $e_s \sim N(0, 1)$ is independent noise. The

target variable is $X_t = \sin Z_1 + \cos Z_2 + Z_3^2 + \exp e_t$, where $e_t \sim \text{Unif}(-1, 1)$ is independent noise.

**Unconditional generation.** The source variable is $X_s = (\sin U_1 + \cos U_2, U_1^2 + U_2^2, \tanh(U_1 U_2), \exp(U_1 - U_2), \log(|U_1| + 1) + \log(|U_2| + 1))$, where $\boldsymbol{U} = (\sin \varepsilon_1, \cos \varepsilon_2)$ and $\varepsilon_j \sim N(0, 1)$ for $j = 1, 2$. The target variable is $X_t = (\sin U_1 + \tanh U_2, U_1^2 + U_2, \exp(U_1 - U_2))$.

We employ diffusion models to learn the underlying distribution and generate synthetic samples closely following the original data's distribution. For the conditional task, the parameters $\Theta_t$, $\Theta_s$, and $\Theta_h$ are implemented as feedforward neural networks with three hidden layers, each comprising 128 units. For the unconditional task, $\Theta_u$ and $\Theta_{g_t}$ are set with the same NN architecture in the conditional models. The detailed frameworks for both cases are illustrated in Figures 1 and 2 of Section 2.

To illustrate the impact of the source data size $n_s$ on the accuracy of target generation, we select $n_s = \lfloor \exp(i) \rfloor$ for $i \in \{8.0, 8.5, 9.0, 9.5, 10.0, 10.5, 11.0\}$, while keeping the target sample size $n_t$ fixed at $5,000$. In this situation, the source and target simulation models differ in error structures. To compute the TV-norm between raw and synthetic samples, we approximate the empirical density via binning and then sum half the absolute differences between the approximated empirical densities. For the Wasserstein distance, we apply the Sinkhorn algorithm (Cuturi, 2013) which calculates the distributional distance by solving an optimal transport problem using suitable metrics.

As shown in Figure 3, generation error for both conditional and unconditional transfer diffusion models declines as the source sample size $n_s$ increases, ultimately outperforming their non-transfer counterparts once $n_s$ passes a modest threshold. Although small $n_s$ can occasionally trigger negative transfer, enlarging $n_s$, particularly when leveraging large pre-trained models, consistently yields positive transfer. In practice, negative transfer can be guarded against via cross-validation. These empirical trends closely mirror the theoretical guarantees of Theorems 1 and 3 and reinforce the interpretations of Theorems 2 and 4.

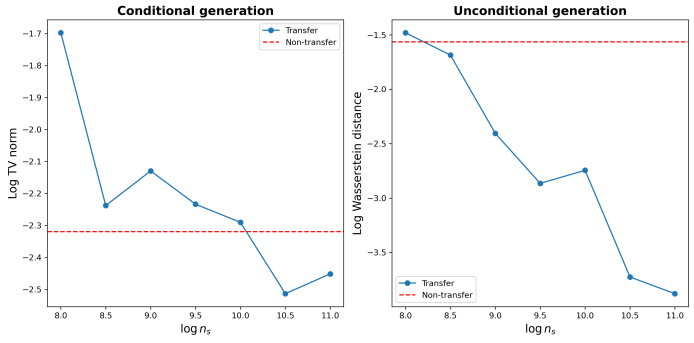



Figure 3: Log-scale generation errors as a function of source sample size $n_s$, with the target sample size fixed at $n_t = 5,000$. The left panel illustrates generation errors in the TV-norm for conditional generation, while the right panel shows generation errors in the Wasserstein distance for unconditional generation. The red dash line represents the errors associated with non-transfer methods.

### 6.2 Benchmark example: MNIST–USPS Digit Images

We investigate image generation on the MNIST–USPS benchmark (Carlucci et al., 2019), a challenging transfer-learning task because the two handwritten-digit corpora differ substantially in res-

olution, stroke style, and intra-class variability. We study both conditional and unconditional generation. The experiment details are given in Appendix B.

**Conditional generation.** We use the MNIST dataset with varying training sample sizes, $n_s \in \{1{,}000,\ 5{,}000,\ 10{,}000,\ 20{,}000,\ 35{,}000,\ 60{,}000\}$, to train a UNet model from the `Diffusers` library, augmented with a class-embedding layer for digit label conditioning. To synthesize USPS digits ($\{0, \cdots, 9\}$), we fine-tune this MNIST-pre-trained model on $n_t = 5{,}103$ USPS training images (approximately 70% of the dataset), while keeping the class-embedding layer frozen. Generation quality is evaluated on a held-out test set of 2,188 USPS images (about 30% of the total) using the 1-Wasserstein distance between real and generated distributions.

**Unconditional generation.** We restrict the task to digit "3" images—an appropriate subset given that the MNIST–USPS benchmark is intended for conditional generation. We start from a diffusion model pre-trained on MNIST and fine-tune it on $n_t = 460$ USPS digit-3 samples, capitalizing on the larger MNIST corpus despite its stylistic gap (see Figure 5b). During pre-training we vary the MNIST source size, $n_s \in \{100, 500, 1{,}000, 2{,}000, 3{,}500, 6{,}000\}$. The UNet denoiser and auto-encoder backbone are implemented with the `Diffusers` library. Generation quality is measured by the 1-Wasserstein distance on an independent test split of 198 USPS digit-3 images (approximately 30% of the dataset).

Figure 4 shows that, in both conditional and unconditional settings, the Wasserstein error of the transfer-diffusion model decreases monotonically as the MNIST pre-training set grows. Holding the USPS fine-tuning size fixed at $n_t$, the transfer model outperforms the USPS-only baseline once the source sample size $n_s$ surpasses a critical threshold. This phenomenon mirrors the trends reported in Section 6 and supports our theoretical claim that richer source data reduce target-domain error. The findings suggest that leveraging shared latent structure boosts generation fidelity, whereas an insufficient source corpus risks negative transfer. Latent-space interpolations (Figure 5) further reveal a smooth stylistic transition from MNIST to USPS.

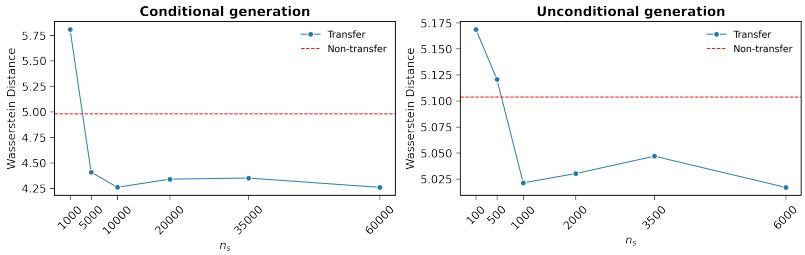
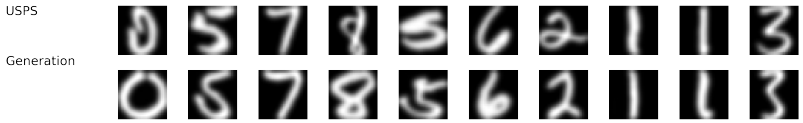
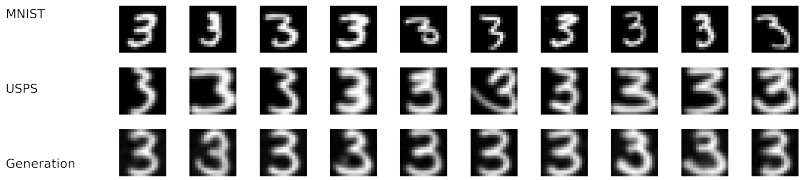



Figure 4: Conditional (all digits, $n_t = 5103$) and unconditional (digit "3", $n_t = 100$) generation accuracy on USPS, measured by the Wasserstein distance and plotted against the MNIST pre-training size $n_s$.The red dashed line indicates the USPS-only baseline without transfer.




(a) Conditional generation example. Real USPS examples (top) and generated samples (bottom).




(b) Unconditional generation example: Digit "3" examples—MNIST (top), USPS (middle), and USPS-style image generated by a diffusion model pretrained on MNIST and adapted to USPS (bottom).

Figure 5: Examples of image generation: conditional and unconditional generation.

## 7. Conclusion

This paper presents a comprehensive theoretical analysis of transfer learning for generative models, emphasizing the transformative potential of these models from a transfer learning perspective. We introduce a shared embedding framework to illustrate how the knowledge transfer between source and target domains facilitates synthetic data generation through shared embeddings. This research provides novel insights into the conditions under which transferred models can achieve enhanced generative performance by systematically quantifying generation errors. We apply our theory to two leading-edge generative models, diffusion models and coupling flows, yielding new results that address unresolved challenges in the field. Additionally, we develop a theory on non-transfer generation accuracy for these models, establishing a standalone benchmark with independent significance.

## Acknowledgments

This work was supported in part by NSF Grant DMS-2513668 and NIH Grants R01AG069895, R01AG065636, R01AG074858, and U01AG073079. It was also partially supported by the Minnesota Supercomputing Institute (MSI) at the University of Minnesota.

## Appendix A. Proofs

### A.1 Proofs for Section 5

**Proof** [Theorem 9] To bound $P(\rho_t(\gamma_t^0, \hat{\gamma}_t) \geq \varepsilon)$, note that

$$P(\rho_t(\gamma_t^0, \hat{\gamma}_t) \leq \varepsilon) \geq P(\rho_t(\gamma_t^0, \hat{\gamma}_t) \leq \varepsilon | (2\delta_t(\hat{h}))^{1/2} \leq \varepsilon) P((2\delta_t(\hat{h}))^{1/2} \leq \varepsilon), \tag{22}$$

where $P(\cdot|2\delta_t(\hat{h}))^{1/2} \leq \varepsilon)$ denotes the conditional probability given event $(2\delta_t(\hat{h}))^{1/2} \leq \varepsilon)$. Here $\varepsilon$, as defined in Theorem 9, is $\varepsilon_t + \sqrt{3c_1}\varepsilon_s$, with $\varepsilon_t \geq \sqrt{2\delta_t(h^0)}$, $\varepsilon_s \geq \sqrt{2\delta_s(h^0)}$,

$$\int_{k_t\varepsilon^2/16}^{4c_{vt}^{1/2}\varepsilon_t} \sup_{h\in\Theta_h} H_B^{1/2}(u, \mathcal{F}_t(h))du \leq c_{h_t} n_t^{1/2} \varepsilon_t^2, \tag{23}$$

and

$$\int_{k_s\varepsilon_s^2/16}^{4c_{vs}^{1/2}\varepsilon_s} H_B^{1/2}(u, \mathcal{F}_s)du \leq c_{h_s} n_s^{1/2} \varepsilon_s^2. \tag{24}$$

The first conditional probability is bounded by Proposition 1 (cf. Remark 1). Note that $\varepsilon \geq \varepsilon_t$ satisfies when $\varepsilon_t$ satisfies (23):

$$\int_{k_t\varepsilon^2/16}^{4c_{vt}^{1/2}\varepsilon} \sup_{h\in\Theta_h} H_B^{1/2}(u, \mathcal{F}_t(h))du \leq c_{h_t} n_t^{1/2} \varepsilon^2. \tag{25}$$

Hence, by Proposition 1, if Assumptions 1, 9 and 10 hold, the conditional probability given the condition $\sqrt{2\delta_t(\hat{h})} \leq \varepsilon$ is bounded:

$$P(\rho_t(\gamma_t^0, \hat{\gamma}_t) \leq \varepsilon | \sqrt{2\delta_t(\hat{h})} \leq \varepsilon) \geq 1 - \exp(-c_{e_t} n_t \varepsilon^2). \tag{26}$$

For the second probability, we will use Assumption 2 to show $\{\rho_s(\gamma_s^0, \hat{\gamma}_s) \leq \varepsilon_s\} \subset \{\sqrt{2\delta_t(\hat{h})} \leq \varepsilon\}$, where $\gamma_s = (\hat{\theta}, \hat{h})$. By the definition of $\delta_s(h) = \inf_{\{\gamma_s=(\theta_s,h):\phi_s\in\Theta_s, h\in\Theta_h\}} \rho_s^2(\gamma_s^0, \gamma_s)$, $\delta_s(\hat{h}) \leq \rho_s^2(\gamma_s^0, \hat{\gamma}_s)$. Under the event $\{\rho_s(\gamma_s^0, \hat{\gamma}_s) \leq \varepsilon_s\}$,

$$|\delta_s(h) - \delta_s(h^0)| \leq \delta_s(h) + \delta_s(h^0) \leq \rho_s^2(\gamma_s^0, \hat{\gamma}_s) + \delta_s(h^0) \leq \frac{3}{2}\varepsilon_s^2.$$

The last inequality uses the assumption that $\sqrt{2\delta_s(h^0)} \leq \varepsilon_s$.

$$\delta_t(\hat{h}) \leq \delta_t(h^0) + c_1|\delta_s(\hat{h}) - \delta_s(h^0)| \leq \delta_t(h^0) + c_1\rho_s^2(\gamma_s^0, \hat{\gamma}_s) + c_1\delta_s(h^0).$$

On the event that $\rho_s(\gamma_s^0, \hat{\gamma}_s) \leq \varepsilon_s$ with $\varepsilon_j \geq \sqrt{2\delta_j(h^0)}$, $j \in \{s, t\}$,

$$\delta_t(\hat{h}) \leq \delta_t(h^0) + c_1\varepsilon_s^2 + c_1\delta_s(h^0) \leq \frac{1}{2}\varepsilon_t^2 + c_1\frac{3}{2}\varepsilon_s^2 \leq \frac{1}{2}(\varepsilon_t + \sqrt{3c_1}\varepsilon_s)^2 \leq \frac{1}{2}\varepsilon^2.$$

This shows $\{\rho_s(\gamma_s^0, \hat{\gamma}_s) \leq \varepsilon_s\} \subset \{\sqrt{2\delta_t(\hat{h})} \leq \varepsilon\}$.

Let $\delta_s = \inf_{\gamma_s\in\Theta_s\times\Theta_h} \rho_s^2(\gamma_s^0, \gamma_s)$. Then, $\delta_s \leq \delta_s(h^0)$ when $h^0 \in \Theta_h$. Thus, $\varepsilon_s > \sqrt{2\delta_s(h^0)} \geq \sqrt{2\delta_s}$. Together with (24), we derive through Proposition 1 that there exists a constant $c_{e_s} > 0$ such that,

$$P((2\delta_t(\hat{h}))^{1/2} \leq \varepsilon) \geq P(\rho_s(\gamma_s^0, \hat{\gamma}_s) \leq \varepsilon_s) \geq 1 - \exp(-c_{e_s} n_s(\varepsilon_s)^2). \tag{27}$$

Plugging (26) and (27) into (22) leads to the final result. This completes the proof. ■

**Lemma 11** *Assume that $f(\boldsymbol{Y}) \in \mathcal{F}$ satisfies the Bernstein condition with some constant $c_b$ for an i.i.d. sample $\boldsymbol{Y}^1, \cdots, \boldsymbol{Y}^n$. Let $\varphi(M, v^2, \mathcal{F}) = \frac{M^2}{2[4v^2 + Mc_b/3n^{1/2}]}$, where $\mathrm{Var}(f(\boldsymbol{Y})) \leq v^2$. Assume that*

$$M \leq kn^{1/2}v^2/4c_b, \tag{28}$$

*with $0 < k < 1$ and*

$$\int_{kM/(8n^{1/2})}^{v} H_B^{1/2}(u, \mathcal{F})du \leq Mk^{3/2}/2^{10}, \tag{29}$$

*then*

$$P^*(\sup_{\{f\in\mathcal{F}\}} n^{-1/2}\sum_{i=1}^{n}(f(\boldsymbol{Y}^i) - \mathrm{E}\, f(\boldsymbol{Y}^i)) \geq M) \leq 3\exp(-(1-k)\varphi(M, v^2, n)),$$

*where $P^*$ denotes the outer probability measure for $\boldsymbol{Y}^1, \cdots, \boldsymbol{Y}^n$.*

**Proof** [Lemma 11] The result follows from the same arguments as in the proof of Theorem 3 in (Shen and Wong, 1994) with $\mathrm{Var}(f(X)) \leq v^2$. Note that Bernstein's condition replaces the upper boundedness condition there, and (4.5) there is not needed here. ■

Suppose that the class of loss functions is indexed by $\gamma \in \Gamma$ and $\mathcal{F} = \{l(\cdot, \pi\gamma^0) - l(\cdot, \gamma)\}$, where $\pi\gamma^0 \in \Gamma$ is an approximate point of $\gamma_t^0$ within $\Gamma$. Suppose $\hat{\gamma} = \arg\min_{\gamma\in\Gamma}\sum_{i=1}^{n} l(\boldsymbol{Y}^i, \gamma)$. Then the following general results are established for $\rho^2(\gamma^0, \hat{\gamma}) = \mathrm{E}(l(\boldsymbol{Y}, \gamma) - l(\boldsymbol{Y}, \gamma^0))$.

**Proposition 1** *If $l(\boldsymbol{Y}, \gamma) - l(\boldsymbol{Y}, \gamma^0)$ satisfies the variance condition (Assumption 9) with a constant $c_v$ and the Bernstein condition (Assumption 10) with a constant $c_v$, there exists a small constant $k$ ($0 < \frac{c_b}{4c_v} \leq k < 1$) such that for $\varepsilon > 0$ satisfying*

$$\int_{k\varepsilon^2/16}^{4c_v^{1/2}\varepsilon} H_B^{1/2}(u, \mathcal{F})du \leq c_h n^{1/2}\varepsilon^2, \quad \varepsilon^2 \geq 2\inf_{\gamma\in\Gamma}\rho^2(\gamma^0, \gamma), \tag{30}$$

*for $c_h = \frac{k^{3/2}}{2^{11}}$ and $c_e = \frac{(1-k)}{(8c_v^2 + \frac{1}{24})}$, we have*

$$P(\rho(\gamma^0, \hat{\gamma}) \geq \varepsilon) \leq 4\exp(-c_e n\varepsilon^2).$$

**Remark 12** *This proposition holds for the source estimation. In the transfer case, the entropy and the approximation error for target learning may depend on the $\hat{h}$ derived from the source. The probability bound here can be extended to the condition probability given the condition of $\hat{h}$ by the independence between source and target training samples assumed in Assumption 1.*

Theorem 10 is a direct result obtained from Proposition 1.

**Proof** [Proposition1] Let $\nu_n(l(\gamma) - l(\pi\gamma^0)) = n^{-1/2}\sum_{i=1}^{n}(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \pi\gamma^0) - \mathrm{E}(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \pi\gamma^0)))$ be an empirical process indexed by $\gamma \in \Gamma$, and $L(\gamma) = \sum_{i=1}^{n}(l(\boldsymbol{Y}^i, \gamma) - \mathrm{E}\, l(\boldsymbol{Y}^i, \gamma))$.

For $l = 0, \cdots$, let $A_l = \{\gamma \in \Gamma : 2^l\varepsilon^2 \le \rho^2(\gamma^0, \gamma) < 2^{l+1}\varepsilon^2\}$. Note that $\sup_{A_l} \mathrm{Var}(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \gamma^0)) \le c_v 2^{l+1}\varepsilon^2$ (Assumption 9) and $\inf_{A_l} E(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \pi\gamma^0)) \ge (2^l - \frac{1}{2})\varepsilon^2$ when $\inf_{\gamma\in\Gamma} \rho^2(\gamma^0, \gamma) \le \rho^2(\gamma^0, \pi\gamma^0) \le \frac{1}{2}\varepsilon^2$; for $i = 1, \cdots, n$. By Assumption 9, $\sup_{A_l} Var(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \pi\gamma^0)) \le 4\sup_{A_l} Var(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \gamma^0)) + 4\sup_{A_l} Var(l(\boldsymbol{Y}^i, \gamma) - l(\boldsymbol{Y}^i, \pi\gamma^0)) \le 8c_v 2^{l+1}\varepsilon^2$. Then $P(\rho(\gamma^0, \hat{\gamma}) \ge \varepsilon)$ is bounded by

$$\begin{aligned} & P^*(\sup_{\{\rho(\gamma^0,\gamma)\ge\varepsilon, \gamma\in\Gamma\}} (L(\gamma) - L(\pi\gamma^0))) \ge 0) \le \sum_{l=0}^{\infty} P^*(\sup_{A_l}(L(\gamma) - L(\pi\gamma^0))) \ge 0) \\ = & \sum_{l=0}^{\infty} P^*(\sup_{A_l} \nu_n((l(\gamma) - l(\pi\gamma^0))) \ge n^{1/2}(2^l - 1/2)\varepsilon^2)). \end{aligned}$$

To apply Lemma 11 to the empirical process over each $A_l$ ($l = 0, \cdot$), we set $M = M_l = n^{1/2}(2^l - 1/2)\varepsilon^2$ and $v^2 = v_l^2 = 8c_v 2^{l+1}\varepsilon^2$ there. Then, $\frac{M}{n^{1/2}v^2} \le \frac{1}{16c_v} \le \frac{k}{4c_{bj}}$, given that $k \ge \frac{c_{bj}}{4c_v}$ according to Assumption 9, leading to (28). Consequently, $\varphi(M_l, v_l^2, \mathcal{F}) = \frac{M_l^2}{8c_v v_l^2 + 2M_l c_b/3n^{1/2}} \ge \frac{M_l^2}{(8c_v + 1/24c_v)v_l^2}$. Furthermore, for any $\varepsilon$ meeting (30), it also fulfills (29) with $(M_l, v_l^2)$ for $l \ge 1$ by examining the least favorable scenario of $l = 0$. By Assumption 10,

$$\begin{aligned} & \sum_{l=0}^{\infty} P^*(\sup_{A_l} \nu_n(l(\gamma) - l(\pi\gamma^0)) \ge n^{1/2}(2^l - 1/2)\varepsilon^2)) \\ \le & \ 3\sum_{l=0}^{\infty} \exp(-(1-k)\frac{(2^l - 1/2)^2 n\varepsilon_j^2}{c_v 2^{l+1}} \frac{1}{(8c_v + 1/24c_v)}) \le 4\exp(-c_e n_j \varepsilon^2), \end{aligned}$$

where $c_e = \frac{(1-k)}{(8c_v^2 + \frac{1}{24})}$. This completes the proof. ■

## A.2 Proofs for Section 3

### A.2.1 ERROR FOR DIFFUSION GENERATION

This subsection presents a general theory for the generation accuracy of conditional and unconditional diffusion models in a generic situation without non-transfer and dimension reduction. Then, we will modify the general results tailored to situations of transfer learning and dimension reduction subsequently.

Consider the conditional generation task for a $d_x$-dimensional vector $\boldsymbol{X}$ given a $d_z$-dimensional vector $\boldsymbol{Z}$. Following the generation process described in Section 3, we use (4)-(5) to construct an empirical score matching loss based on a training set $(\boldsymbol{x}^i, \boldsymbol{z}^i)_{i=1}^n$ of size $n$, $L(\theta) = \sum_{i=1}^n l(\boldsymbol{x}^i, \boldsymbol{z}^i; \theta)$ in (6) with

$$l(\boldsymbol{x}, \boldsymbol{z}; \theta) = \int_{\underline{\tau}}^{\overline{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x},\boldsymbol{z}} \|\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x},\boldsymbol{z}}(\boldsymbol{X}(\tau)|\boldsymbol{x}, \boldsymbol{z}) - \theta(\boldsymbol{X}(\tau), \boldsymbol{z}, \tau)\|^2 \mathrm{d}\tau, \tag{31}$$

where the estimated score $\hat{\theta}(\boldsymbol{x}(\tau), \boldsymbol{z}, \tau) = \arg\min_{\theta\in\Theta} L(\theta)$. Here, the parameter space $\Theta$ is defined as: $\Theta = \{\theta \in \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E}) : \mathbb{R}^{d_x + d_z + 1} \to \mathbb{R}^{d_x}\}$, representing a ReLU network with $\mathbb{L}$

layers, a maximum width of $\mathbb{W}$, the number of effective parameters $\mathbb{S}$, the output sup-norm $\mathbb{B}$ and the parameter bound $\mathbb{E}$.

Following the sampling scheme of the reverse process (9) and its alternative described in Section 3.2, we derive the density of the generation sample $\hat{p}_{\boldsymbol{x}|\boldsymbol{z}}$.

Next, we make some assumptions about the true conditional density $p^0_{\boldsymbol{x}|\boldsymbol{z}}$.

**Assumption 11** *Assume that $p^0_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}|\boldsymbol{z}) = \exp(-c_f\|\boldsymbol{x}\|^2/2)\cdot f(\boldsymbol{x},\boldsymbol{z})$, where $f$ belongs to $\mathcal{C}^r(\mathbb{R}^{d_x}\times[0,1]^{d_z},\mathbb{R},B)$ for a constant radius $B>0$ and $c_f>0$ is a constant. Assume that $f$ is lower bounded away from zero with $f\geq \underline{c}$.*

Next, we give the results of the generation accuracy for conditional diffusion models.

**Theorem 13 (Generation error of diffusion models)** *Under Assumption 11, setting the neural network's structural hyperparameters of $\Theta=\mathrm{NN}(\mathbb{L},\mathbb{W},\mathbb{S},\mathbb{B},\mathbb{E})$ as follows: $\mathbb{L}=c_L\log^4 K$, $\mathbb{W}=c_W K\log^7 K$, $\mathbb{S}=c_S K\log^9 K$, $\log\mathbb{B}=c_B\log K$, $\log\mathbb{E}=c_E\log^4 K$, with stopping criteria from (4)-(5) as $\log\underline{\tau}=-c_{\underline{\tau}}\log K$, $\overline{\tau}=c_{\overline{\tau}}\log K$, and $\underline{\tau}^*=\mathbb{I}_{\{r\leq 1\}}\underline{\tau}$, where $\{c_L,c_W,c_S,c_B,c_E,c_{\underline{\tau}},c_{\overline{\tau}}\}$ are sufficiently large constants, yields the error in diffusion generation via transfer learning:*

$$P(\mathrm{E}_{\boldsymbol{z}}[\mathrm{TV}(p^0_{\boldsymbol{x}|\boldsymbol{z}},\hat{p}_{\boldsymbol{x}|\boldsymbol{z}})]\geq x(\beta_n+\delta_n))\leq\exp(-c_e n^{1-\xi}(x(\beta_n+\delta_n))^2),\ \textit{if } r>0;$$
$$P(\mathrm{E}_{\boldsymbol{z}}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}|\boldsymbol{z}},\hat{p}_{\boldsymbol{x}|\boldsymbol{z}})]\geq x(\beta_n+\delta_n))\leq\exp(-c_e n^{1-\xi}(x(\beta_n+\delta_n))^2),\ \textit{if } r>1,$$

*for any $x\geq 1$, some constant $c_e>0$ and a small $\xi>0$. Here, $\beta_n$ and $\delta_n$ represent the estimation and approximation errors, given by: $\beta_n\asymp\sqrt{\frac{K\log^{19}K}{n}}$, $\quad\delta_n\asymp K^{-\frac{r}{d_x+d_z}}\log^{\frac{r}{2}+1}K$. Setting $\beta_n=\delta_n$ to solve for $K$, and neglecting the logarithmic term, leads to $\beta_n=\delta_n\asymp n^{-\frac{r}{d_x+d_z+2r}}\log^m n$ with the optimal $K\asymp n^{\frac{d_x+d_z}{d_x+d_z+2r}}$, with $m=\max(\frac{19}{2},\frac{r}{2}+1)$. Consequently, this provides the best bound $n^{-\frac{r}{d_x+d_z+2r}}\log^m n$.*

*Moreover, we extend this error bound to unconditional diffusion generation with $\boldsymbol{Z}=\emptyset$ and $d_z=0$, $\mathrm{TV}(p^0_{\boldsymbol{x}},\hat{p}_{\boldsymbol{x}})=O_p(n_t^{-\frac{r}{d_x+2r}}\log^m n_t)$ for $r>0$ and $\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}},\hat{p}_{\boldsymbol{x}})=O_p(n_t^{-\frac{r}{d_x+2r}}\log^m n_t)$ for $r>1$.*

To simplify the notation, we write $p_{\boldsymbol{x}(\tau)|\boldsymbol{z}}(\boldsymbol{x}|\boldsymbol{z})$ as $p_\tau(\boldsymbol{x}|\boldsymbol{z})$ and $p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}(\boldsymbol{x}|\boldsymbol{x}(0))$ as $p_\tau(\boldsymbol{x}|\boldsymbol{x}(0))$ in subsequent proofs.

**Lemma 14 (Approximation error of $\Theta$)** *Under Assumption 11, there exists a ReLU network $\Theta=\mathrm{NN}(\mathbb{L},\mathbb{W},\mathbb{S},\mathbb{B},\mathbb{E})$ with depth $\mathbb{L}=c_L\log^4 K$, width $\mathbb{W}=c_W K\log^7 K$, the number of effective parameters $\mathbb{S}=c_S K\log^9 K$, the parameter bound $\log\mathbb{E}=c_E\log^4 K$, and $\mathbb{B}=\sup_\tau\mathbb{B}(\tau)=\sup_\tau c_B\sqrt{\log K}/\sigma_\tau$, such that for $\theta^0$ there exists $\pi\theta^0\in\Theta$ with*

$$\rho(\theta^0,\pi\theta^0)=O(K^{-\frac{r}{d_x+d_z}}\log^{\frac{r}{2}+1}K),\tag{32}$$

*provided that $\log\underline{\tau}=-c_{\underline{\tau}}\log K$ and $\overline{\tau}=c_{\overline{\tau}}\log K$ with sufficiently large constants $c_{\underline{\tau}}>0$ and $c_{\overline{\tau}}>0$. Furthermore, there exists a subnetwork $\mathrm{NN}(\mathbb{L},\mathbb{W},\mathbb{S},\mathbb{B},\mathbb{E},\lambda)$, which has $\alpha$ Hölder continuity with $\lambda=c_\lambda$, $\alpha=\min(1,r-1)$ when $r>1$ and $\lambda=c_\lambda/\sigma_{\underline{\tau}}$, $\alpha=r$ when $r\leq 1$, where $c_\lambda$ is a sufficiently large positive constant. For $\theta^0$, there exists $\pi\theta^0\in\tilde{\Theta}=\mathrm{NN}(\mathbb{L},\mathbb{W},\mathbb{S},\mathbb{B},\mathbb{E},\lambda)$ such that* (32) *holds.*

The approximation error for $\Theta = \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})$ in Lemma 14 directly follows from Fu et al. (2024), which outlines this network's architecture and the input domain of the neural network can be limited in $[-R, R]^{d_x} \times [0,1]^{d_z} \times [\underline{\tau}, \overline{\tau}]$ where $R = c_x\sqrt{\log K}$ with $c_x$ is a positive constant dependent on the parameters in the true density. The approximation error for $\tilde{\Theta} = \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E}, \lambda)$ is obtained by applying an $\alpha$ Hölder constraint to the network, where we choose $\lambda$ by the following lemma on the smooth property of the true score function.

The subsequent lemma elucidates the degree of smoothness of the gradient.

**Lemma 15 (Gradient Hölder continunity)** *Under Assumption 11, for any $\boldsymbol{x} \in \mathbb{R}^{d_x}$, $\tau > 0$, and $\boldsymbol{z}, \boldsymbol{z}' \in [0,1]^{d_z}$,*

$$\sup_{\boldsymbol{x}} \frac{\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}')\|_\infty}{\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha} \le \begin{cases} c_1^h, & \text{with } \alpha = \min(r-1, 1) \text{ if } r > 1, \\ c_2^h/\sigma_\tau, & \text{with } \alpha = r \text{ if } r \le 1, \end{cases}$$

*where $c_1^h = (B/\underline{c} + B^2\sqrt{d_z}^{1-\alpha}/\underline{c}^2)\max(1, 1/c_f)$, $c_2^h = \sqrt{\frac{\pi}{2}}\left(B/\underline{c} + B^2/\underline{c}^2\right)\max(1, \sqrt{1/c_f})$, and $B, \underline{c}, c_f$ are specified in Assumption 11.*

**Proof** Consider any $\boldsymbol{x} \in \mathbb{R}^{d_x}$, $\tau > 0$, and $\boldsymbol{z}, \boldsymbol{z}' \in [0,1]^{d_z}$ in what follows.

By (4), the density of $\boldsymbol{X}(\tau)$ given $\boldsymbol{Z}$ can be expressed through a mixture of the Gaussian and initial distribution:

$$p_\tau(\boldsymbol{x}|\boldsymbol{z}) = \int p_N(\boldsymbol{x}; \mu_\tau \boldsymbol{y}, \sigma_\tau) p_{\boldsymbol{x}(0)|\boldsymbol{z}}(\boldsymbol{y}|\boldsymbol{z}) \mathrm{d}\boldsymbol{y}, \tag{33}$$

where $p_N(\cdot; \mu_\tau \boldsymbol{y}, \sigma_\tau)$ is the Gaussian density of $N(\mu_\tau \boldsymbol{y}, \sigma_\tau^2 \boldsymbol{I})$ with $\mu_\tau = \exp(-\tau)$, $\sigma_\tau^2 = 1 - \exp(-2\tau)$, and $p_{\boldsymbol{x}(0)|\boldsymbol{z}}(\boldsymbol{x}|\boldsymbol{z}) = \exp(-c_f\|\boldsymbol{x}\|^2/2) \cdot f(\boldsymbol{x}, \boldsymbol{z})$ is given in Assumption 11.

Direct calculations from (33) yield that

$$\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) = -\frac{c_f \boldsymbol{x}}{(\bar{\mu}_\tau^2 + c_f \bar{\sigma}_\tau^2)} + \frac{\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)}{g(\boldsymbol{x}, \boldsymbol{z}, \tau)}, \tag{34}$$

where $\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) = \frac{\partial g(\boldsymbol{x}, \boldsymbol{z}, \tau)}{\partial \boldsymbol{x}}$ and

$$g(\boldsymbol{x}, \boldsymbol{z}, \tau) = \int f(\boldsymbol{y}, \boldsymbol{z}) p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} \ge \underline{c}. \tag{35}$$

The lower bound holds with $f \ge \underline{c}$. Here $p_N(\cdot; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau)$ is the Gaussian density of $N(\bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau^2 \boldsymbol{I})$ with $\bar{\mu}_\tau = \frac{\mu_\tau}{\mu_\tau^2 + c_f\sigma_\tau^2}$ and $\bar{\sigma}_\tau = \frac{\sigma_\tau}{\sqrt{\mu_\tau^2 + c_f\sigma_\tau^2}}$. Note that $\bar{\sigma}_\tau \to 0$ and $\bar{\mu}_\tau \to 1$ as $\tau \to 0$. Furthermore, $\bar{\mu}_\tau \le \max(1, 1/c_f)$ and $\bar{\sigma}_\tau \ge \sigma_\tau \min(1, 1/c_f^{1/2})$ since $\min(1, c_f) \le \mu_\tau^2 + c_f\sigma_\tau^2 \le \max(1, c_f)$. Hence,

$$\begin{aligned} \|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}')\|_\infty &\le \left\| \frac{\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)}{g(\boldsymbol{x}, \boldsymbol{z}, \tau)} - \frac{\nabla g(\boldsymbol{x}, \boldsymbol{z}', \tau)}{g(\boldsymbol{x}, \boldsymbol{z}', \tau)} \right\|_\infty \\ &\le \left\| \frac{\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla g(\boldsymbol{x}, \boldsymbol{z}', \tau)}{g(\boldsymbol{x}, \boldsymbol{z}', \tau)} \right\|_\infty + \left\|\nabla g(\boldsymbol{x}, \boldsymbol{z}', \tau)\right\|_\infty \left| \frac{g(\boldsymbol{x}, \boldsymbol{z}, \tau) - g(\boldsymbol{x}, \boldsymbol{z}', \tau)}{g(\boldsymbol{x}, \boldsymbol{z}, \tau) g(\boldsymbol{x}, \boldsymbol{z}', \tau)} \right|. \end{aligned} \tag{36}$$

To bound (36), we first consider the case of $r > 1$ ($\alpha = \min(r-1, 1)$). By integration by parts,

$$\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) = \int_{R^{d_x}} f(\boldsymbol{y}, \boldsymbol{z}) \nabla_{\boldsymbol{x}} p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} = \bar{\mu}_\tau \int_{R^{d_x}} \nabla f(\boldsymbol{y}, \boldsymbol{z}) p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y}. \quad (37)$$

By Assumption 11, $\|\nabla f(\boldsymbol{y}, \boldsymbol{z})\|_\infty \leq B$ since $f \in \mathcal{C}^r(\mathbb{R}^{d_x} \times [0,1]^{d_z}, \mathbb{R}, B)$. Then

$$\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \leq B\bar{\mu}_\tau. \quad (38)$$

By Assumption 11 with $r > 1$, $\|\nabla f(\boldsymbol{y}, \boldsymbol{z}) - \nabla f(\boldsymbol{y}, \boldsymbol{z}')\|_\infty \leq B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha$. By (37),

$$\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla g(\boldsymbol{x}, \boldsymbol{z}', \tau)\|_\infty \leq \bar{\mu}_\tau \int B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} \leq B\bar{\mu}_\tau \|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha. \quad (39)$$

Similarly as in (39), with $\alpha = \min(r-1, 1)$,

$$\|g(\boldsymbol{x}, \boldsymbol{z}, \tau) - g(\boldsymbol{x}, \boldsymbol{z}', \tau)\|_\infty \leq B\|\boldsymbol{z} - \boldsymbol{z}'\| \leq \sqrt{d_z}^{1-\alpha} B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha. \quad (40)$$

Plugging (35) and (38)–(40) into (36) yields that

$$\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}')\|_\infty \leq \left(B\bar{\mu}_\tau/\underline{c} + B^2\bar{\mu}_\tau \sqrt{d_z}^{1-\alpha}/\underline{c}^2\right) \|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha \leq c_1^h \|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha$$

for some constant $c_1^h = \left(B/\underline{c} + B^2\sqrt{d_z}^{1-\alpha}/\underline{c}^2\right) \max(1, 1/c_f)$ since $\bar{\mu}_\tau \leq \max(1, 1/c_f)$.

Next, we consider the case of $r \leq 1$, with $\alpha = r$. For any $j = 1, 2, \ldots, d_x$, the partial derivative of $g$ in the $j$-th element $\boldsymbol{x}_j$ of $\boldsymbol{x}$ $\nabla_{\boldsymbol{x}_j} g$ satisfies:

$$\begin{aligned} |\nabla_{\boldsymbol{x}_j} g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla_{\boldsymbol{x}_j} g(\boldsymbol{x}, \boldsymbol{z}', \tau)| &\leq \int |f(\boldsymbol{y}, \boldsymbol{z}) - f(\boldsymbol{y}, \boldsymbol{z}')| |\nabla_{\boldsymbol{x}_j} p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau)| \mathrm{d}\boldsymbol{y} \\ &\leq B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha \int |\nabla_{\boldsymbol{x}_j} p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau)| \mathrm{d}\boldsymbol{y} = B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha \sqrt{\frac{\pi}{2}} \frac{\bar{\mu}_\tau}{\bar{\sigma}_\tau}, \end{aligned} \quad (41)$$

since $\int |\nabla_{\boldsymbol{x}_j} p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau)| \mathrm{d}\boldsymbol{y} = \bar{\mu}_\tau \int \frac{|\boldsymbol{y}_j - \bar{\mu}_\tau \boldsymbol{x}_j|}{\sigma_\tau} p_N(\boldsymbol{y}_j; \bar{\mu}_\tau \boldsymbol{x}_j, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y}_j = \sqrt{\frac{\pi}{2}} \frac{\bar{\mu}_\tau}{\bar{\sigma}_\tau}$. As in (40),

$$\|g(\boldsymbol{x}, \boldsymbol{z}, \tau) - g(\boldsymbol{x}, \boldsymbol{z}', \tau)\|_\infty \leq B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha \int p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} = B\|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha. \quad (42)$$

By (41) and the fact that $0 < f(\boldsymbol{y}, \boldsymbol{z}) \leq B$,

$$\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \leq \sup_{1 \leq j \leq d_x} \int f(\boldsymbol{y}, \boldsymbol{z}) |\nabla_{\boldsymbol{x}_j} p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau)| \mathrm{d}\boldsymbol{y} \leq B \frac{\bar{\mu}_\tau}{\bar{\sigma}_\tau} \sqrt{\frac{\pi}{2}}. \quad (43)$$

Plugging the bounds from (35) and (41)–(43) into (36) yields that

$$\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}')\|_\infty \leq \sqrt{\frac{\pi}{2}} \left(B/\underline{c} + B^2/\underline{c}^2\right) \frac{\bar{\mu}_\tau}{\bar{\sigma}_\tau} \leq \frac{c_2^h}{\sigma_\tau} \|\boldsymbol{z} - \boldsymbol{z}'\|^\alpha,$$

for some constant $c_2^h = \sqrt{\frac{\pi}{2}}\left(B/\underline{c} + B^2/\underline{c}^2\right)\max(1, \sqrt{1/c_f})$ since $\bar{\mu}_\tau \le \max(1, 1/c_f)$. This completes the proof. ■

**Lemma 16 (Metric entropy)** *For the network $\Theta = \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})$ defined in Lemma 14, the metric entropy of $\mathcal{F} = \{l(\cdot;\theta) - l(\cdot;\pi\theta^0) : \theta \in \Theta\}$ is bounded, $H_B(u, \mathcal{F}) = O(K \log^{16} K \log(\frac{K}{u}))$.*

**Proof** [Lemma 16] For $\theta_j \in \Theta$; $j = 1, 2$, consider the case $\sup_{\boldsymbol{x},\boldsymbol{z},\tau} \|\theta_1(\boldsymbol{x}, \boldsymbol{z}, \tau) - \theta_2(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \le u$. By (31), for any $\boldsymbol{x}(0), \boldsymbol{z}$,

$$
\begin{aligned}
&|l(\boldsymbol{x}(0), \boldsymbol{z}; \theta_1) - l(\boldsymbol{x}(0), \boldsymbol{z}; \theta_2)| \\
\le &\int_{\underline{\tau}}^{\overline{\tau}} \int_{\mathbb{R}^{d_x}} (\|\theta_1(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z})\| + \|\theta_2(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z})\|) \\
&\|\theta_1(\boldsymbol{x}, \boldsymbol{z}, \tau) - \theta_2(\boldsymbol{x}, \boldsymbol{z}, \tau)\| p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z}) \mathrm{d}\boldsymbol{x} \mathrm{d}\tau \\
\le &2 d_x^{1/2} u \int_{\underline{\tau}}^{\overline{\tau}} \int_{\mathbb{R}^{d_x}} (\sup_{\theta\in\Theta,\boldsymbol{x},\boldsymbol{z}} \|\theta(\boldsymbol{x}, \boldsymbol{z}, \tau)\| + \|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z})\|) p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z}) \mathrm{d}\boldsymbol{x} \mathrm{d}\tau. \qquad (44)
\end{aligned}
$$

Moreover, note that $p_\tau(\boldsymbol{x}|\boldsymbol{x}(0), \boldsymbol{z}) = p_\tau(\boldsymbol{x}|\boldsymbol{x}(0)) = p_N(\boldsymbol{x}; \mu_\tau \boldsymbol{x}(0), \sigma_\tau^2)$, and

$$
\int_{\mathbb{R}^{d_x}} \|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0))\|^2 p_\tau(\boldsymbol{x}|\boldsymbol{x}(0)) \mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^{d_x}} \frac{\|\boldsymbol{x} - \mu_\tau \boldsymbol{x}(0)\|^2}{\sigma_\tau^4} p_\tau(\boldsymbol{x}|\boldsymbol{x}(0)) \mathrm{d}\boldsymbol{x} = \frac{d_x}{\sigma_\tau^2}. \qquad (45)
$$

By the assumption of Lemma 2, $\sup_{\theta\in\Theta,\boldsymbol{x},\boldsymbol{z}} \|\theta(\boldsymbol{x}, \boldsymbol{z}, \tau)\| \le \sqrt{d_x} c_B \frac{\sqrt{\log K}}{\sigma_\tau}$. By Cauchy-Schwartz inequality and (45), (44) is bounded by

$$
\begin{aligned}
(44) \le &2 d_x u \int_{\underline{\tau}}^{\overline{\tau}} (c_B \sqrt{\log K} + 1)/\sigma_\tau \mathrm{d}\tau \le 2 d_x^{1/2} u \int_{\underline{\tau}}^{\overline{\tau}} \frac{c_B \sqrt{\log K} + 1}{\sqrt{1 - e^{-1}} \min(1, \sqrt{2\tau})} \mathrm{d}\tau \\
\le &\frac{2 d_x (c_B \sqrt{\log K} + 1) c_{\overline{\tau}} \log K}{\sqrt{1 - e^{-1}}} u \le c^* (\log^{3/2} K) u,
\end{aligned}
$$

with $c^* = \frac{2 d_x (c_B + 1) c_{\overline{\tau}}}{\sqrt{1 - e^{-1}}}$ when $K > e$, where the forth inequality is because $\sigma_\tau = \sqrt{1 - \exp(-2\tau)} \ge \sqrt{1 - e^{-1}}\sqrt{2\tau}$ when $\tau \le \frac{1}{2}$ and $\sigma_\tau \ge \sqrt{1 - e^{-1}}$ when $\tau > \frac{1}{2}$.

Then, $(\mathbb{E}_{\boldsymbol{x},\boldsymbol{z}} |l(\boldsymbol{x}(0), \boldsymbol{z}; \theta_1) - l(\boldsymbol{x}(0), \boldsymbol{z}; \theta_2)|^2)^{1/2} \le c^* (\log^{3/2} K) u$. Consequently, by Lemma 2.1 in Ossiander (1987), we can bound the bracketing $L_2$ metric entropy by the $L_\infty$ metric entropy $H(u, \Theta)$, the logarithm of the number of $u$ balls in the sup norm needed to cover $\Theta$, $H_B(u, \mathcal{F}) \le H((2c^* \log^{3/2} K)^{-1} u, \Theta)$ for small $u > 0$, where $\Theta$ is defined in Lemma 14 same as in Fu et al. (2024).

By Lemma C.2 Oko et al. (2023) and Lemma D.8 Fu et al. (2024), $H(\cdot, \Theta)$ is bounded by the hyperparameters of depth $\mathbb{L}$, width $\mathbb{W}$, number of parameters $\mathbb{S}$, parameter bound $\mathbb{E}$ and the diameter of the input domain, $H(u, \Theta) \le O(\mathbb{S}\mathbb{L} \log(\mathbb{E}\mathbb{W}\mathbb{L} \max(R, \overline{\tau})/u)) = O(K(\log^{13} K)(\log^4 K - \log u))$ given the approximation error in Lemma 14 with $\mathbb{L} \asymp \log^4 K$, $\mathbb{W} \asymp K \log^7 K$, $\mathbb{S} \asymp K \log^9 K$, $\log \mathbb{E} \asymp \log^4 K$, $R \asymp \sqrt{\log K}$ and $\overline{\tau} \asymp \log K$. Thus, $H(\delta, \mathcal{F}) \le H((c^* \log^{3/2} K)^{-1} u, \Theta) = O\left(K \log^{16} K \log \frac{K}{u}\right)$. This completes the proof. ■

**Lemma 17** *Let $\alpha = \min(1, r-1)$ when $r > 1$. Under Assumption 11, for any $\boldsymbol{z} \in [0,1]^{d_z}$ and small $\underline{\tau} > 0$, there exists a constant $c > 0$, as given in* (51)*, such that*

$$\int_0^{\underline{\tau}} \mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\tau)|\boldsymbol{z}} \|\nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\underline{\tau})|\boldsymbol{z}) - \nabla \log p_{\tau}(\boldsymbol{X}(\tau)|\boldsymbol{z})\|^2 \mathrm{d}\tau \leq c\underline{\tau}^{1+\alpha}. \tag{46}$$

**Proof** Consider any $0 \leq \kappa < \underline{\tau}$. First,

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\underline{\tau})|\boldsymbol{z}) - \nabla \log p_{\kappa}(\boldsymbol{X}(\kappa)|\boldsymbol{z})\|^2 \leq I_3 + I_4,\\
&I_3 = \mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\underline{\tau})|\boldsymbol{z}) - \nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\kappa)|\boldsymbol{z})\|^2,\\
&I_4 = \mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\kappa)|\boldsymbol{z}) - \nabla \log p_{\kappa}(\boldsymbol{X}(\kappa)|\boldsymbol{z})\|^2.
\end{aligned}$$

By (52) in Lemma 18, for some constants $c_1^g > 0$ and $c_2^g > 0$,

$$I_3 \leq 2d_x (c_1^g)^2 (\mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\underline{\tau}) - \boldsymbol{X}(\kappa)\|^2 + 2d_x (c_2^g)^2 \mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\underline{\tau}) - \boldsymbol{X}(\kappa)\|^{2\alpha}).$$

For the first term of $I_3$, by (4) with $b_\tau = 1$,

$$\begin{aligned}
&\mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\underline{\tau}) - \boldsymbol{X}(\kappa)\|^2 = \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z},w(\tau)} \left\| \int_\kappa^{\underline{\tau}} -b_\tau \boldsymbol{X}(\tau) \mathrm{d}\tau + \sqrt{2b_\tau} \int_\kappa^{\underline{\tau}} dW(\tau) \right\|^2\\
&\leq 2\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z},w(\tau)} \left( \left\| \int_\kappa^{\underline{\tau}} \boldsymbol{X}(\tau) \mathrm{d}\tau \right\|^2 + 2 \left\| \int_\kappa^{\underline{\tau}} dW(\tau) \right\|^2 \right)\\
&\leq 2(\underline{\tau} - \kappa) \left( \int_\kappa^{\underline{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z}} \|\boldsymbol{X}(\tau)\|^2 \mathrm{d}\tau + 2d_x \right),
\end{aligned}$$

where the last inequality uses the fact that $\| \int_\kappa^{\underline{\tau}} \boldsymbol{X}(\tau) \mathrm{d}\tau \|^2 \leq (\underline{\tau} - \kappa) \int_\kappa^{\underline{\tau}} \|\boldsymbol{X}(\tau)\|^2 \mathrm{d}\tau$ by the Cauchy-Schwartz inequality, and $\int_\kappa^{\underline{\tau}} dW(\tau) \sim N(0, \underline{\tau} - \kappa)$.

Similarly, for the second term in $I_3$, by Jensen's inequality, $\mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\underline{\tau}) - \boldsymbol{X}(\kappa)\|^{2\alpha} \leq \left(\mathrm{E}_{\boldsymbol{x}(\underline{\tau}),\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\underline{\tau}) - \boldsymbol{X}(\kappa)\|^2\right)^\alpha \leq 2(\underline{\tau} - \kappa)^\alpha \left( \int_\kappa^{\underline{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z}} \|\boldsymbol{X}(\tau)\|^2 \mathrm{d}\tau + 2d_x \right)^\alpha$ for $0 < \alpha \leq 1$. Moreover, note that $\boldsymbol{X}(\tau) \sim N(\mu_\tau \boldsymbol{X}(0), \sigma_\tau^2 \boldsymbol{I})$ given $\boldsymbol{X}(0)$. Direct computation yields that

$$\begin{aligned}
\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z}} \|\boldsymbol{X}(\tau)\|^2 &= \mathrm{E}_{\boldsymbol{x}(0)|z} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)} \|\boldsymbol{X}(\tau)\|^2 \leq \mathbb{E}_{\boldsymbol{x}(0)|z} (\mu_\tau^2 \|\boldsymbol{X}(0)\|^2 + d_x)\\
&\leq \mathrm{E}_{\boldsymbol{x}(0)|z} \|\boldsymbol{X}(0)\|^2 + d_x \leq c_M,
\end{aligned} \tag{47}$$

where $c_M = \sqrt{2\pi} B d_x / c_f^{3/2} + d_x$ is derived from that fact that $p_{\boldsymbol{x}(0)|z}(\boldsymbol{x}|\boldsymbol{z}) \leq B \exp(-\frac{c_f \|\boldsymbol{x}\|^2}{2})$ for any $\boldsymbol{z}$ in Assumption 11. Hence, given sufficiently small $\underline{\tau}$, combining these two bounds yields that $I_3 \leq c_{I_3} (\underline{\tau} - \kappa)^\alpha$ for $c_{I_3} = 8d_x \max(c_1^g, c_2^g)^2 (c_M + 2d_x)$.

By (34), $I_4 \leq I_5 + I_6$ with

$$\begin{aligned}
I_5 &= \mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \left\| \frac{c_f \boldsymbol{X}(\kappa)}{(\mu_{\underline{\tau}}^2 + c_f \sigma_{\underline{\tau}}^2)} - \frac{c_f \boldsymbol{X}(\kappa)}{(\mu_\kappa^2 + c_f \sigma_\kappa^2)} \right\|^2,\\
I_6 &= \mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\nabla \log g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \tau) - \nabla \log g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \kappa)\|^2.
\end{aligned}$$

By (47), given $\mu_\tau = \exp(-\tau)$ and $\sigma_\tau^2 = 1 - \exp(-2\tau)$,

$$I_5 \leq \sup_{\tau \in [0, \underline{\tau}]} \frac{\partial}{\partial \tau} \left[ \frac{c_f}{\mu_\tau^2 + c_f \sigma_\tau^2} \right]^2 (\underline{\tau} - \kappa)^2 \mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\boldsymbol{X}(\kappa)\|^2 \leq c_{I_5} (\underline{\tau} - \kappa)^2,$$

where $\sup_{\tau \in [0, \underline{\tau}]} \left[ \frac{\partial}{\partial \tau} \left( \frac{c_f}{\mu_\tau^2 + c_f \sigma_\tau^2} \right) \right]^2 \leq 4 \frac{c_f^2 (1 - c_f)^2}{\min(1, c_f^2)}$ and $c_{I_5} = 4 \frac{c_f^2 (1 - c_f)^2}{\min(1, c_f^2)} c_M > 0$.

To bound $I_6$, we bound $\|\nabla g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \tau) - \nabla f(\boldsymbol{X}(\kappa), \boldsymbol{z})\|$ and $\|g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \tau) - f(\boldsymbol{X}(\kappa), \boldsymbol{z})\|$ separately. By (37), $\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) = \bar{\mu}_\tau \int_{R^{d_x}} \nabla f(\boldsymbol{y}, \boldsymbol{z}) p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y}$. By the triangle inequality, $\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla f(\boldsymbol{x}, \boldsymbol{z})\|$ is bounded by

$$\begin{aligned} &\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \bar{\mu}_\tau \nabla f(\bar{\mu}_\tau \boldsymbol{x}, \boldsymbol{z})\| + \bar{\mu}_\tau \|\nabla f(\bar{\mu}_\tau \boldsymbol{x}, \boldsymbol{z}) - \nabla f(\boldsymbol{x}, \boldsymbol{z})\| + \|\bar{\mu}_\tau \nabla f(\boldsymbol{x}, \boldsymbol{z}) - \nabla f(\boldsymbol{x}, \boldsymbol{z})\| \\ &\leq \bar{\mu}_\tau \left\| \int (\nabla f(\bar{\mu}_\tau \boldsymbol{x}) - \nabla f(\boldsymbol{y})) p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} \right\| + \bar{\mu}_\tau B \left\| (1 - \bar{\mu}_\tau) \boldsymbol{x} \right\|^\alpha + |1 - \bar{\mu}_\tau| B. \end{aligned} \tag{48}$$

Using the smooth property and the Cauchy-Schwartz inequality, the first term in (48) is bounded by $\bar{\mu}_\tau \int B \|\bar{\mu}_\tau \boldsymbol{x} - \boldsymbol{y}\|^\alpha p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} \leq \bar{\mu}_\tau B (d_x \bar{\sigma}_\tau^2)^{\alpha/2}$. Then, (48), with $\bar{\mu}_\tau = \frac{\mu_\tau}{\mu_\tau^2 + c_f \sigma_\tau^2}$ and $\bar{\sigma}_\tau = \frac{\sigma_\tau}{\sqrt{\mu_\tau^2 + c_f \sigma_\tau^2}}$, is bounded by

$$\begin{aligned} &(\bar{\mu}_\tau (d_x \bar{\sigma}_\tau^2)^{\alpha/2} + \bar{\mu}_\tau \left\| (1 - \bar{\mu}_\tau) \boldsymbol{x} \right\|^\alpha + |1 - \bar{\mu}_\tau|) B \\ &\leq \frac{B (2 d_x \tau)^{\alpha/2}}{\min(1, c_f^{1 + \alpha/2})} + \frac{(2 c_f + 1) B \tau^\alpha}{\min(1, c_f^{1+\alpha})} \|\boldsymbol{x}\|^\alpha + \frac{(2 c_f + 1) B \tau}{\min(1, c_f)}. \end{aligned}$$

The last inequality holds because $\sigma_\tau^2 = 1 - \exp(-2\tau) \asymp 2\tau$ when $\tau \to 0$. Combining these constants and using the bound in (47) leads to

$$\mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|\nabla g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \tau) - \nabla f(\boldsymbol{X}(\kappa), \boldsymbol{z})\|^2 \leq 3 \frac{\max((2 c_f + 1) B (c_M)^\alpha, B (2 d_x)^{\alpha/2})^2}{\min(1, c_f^{2 + 2\alpha})} \tau^\alpha. \tag{49}$$

Similarly,

$$\begin{aligned} \|g(\boldsymbol{x}, \boldsymbol{z}, \tau) - f(\boldsymbol{x}, \boldsymbol{z})\| &\leq \|g(\boldsymbol{x}, \boldsymbol{z}, \tau) - f(\bar{\mu}_\tau \boldsymbol{x}, \boldsymbol{z})\| + \|f(\bar{\mu}_\tau \boldsymbol{x}, \boldsymbol{z}) - f(\boldsymbol{x}, \boldsymbol{z})\| \\ &\leq \| \int (f(\bar{\mu}_\tau \boldsymbol{x}, \boldsymbol{z}) - f(\boldsymbol{y}, \boldsymbol{z}))) p_N(\boldsymbol{y}; \bar{\mu}_\tau \boldsymbol{x}, \bar{\sigma}_\tau) \mathrm{d}\boldsymbol{y} \| + B |1 - \bar{\mu}_\tau| \|\boldsymbol{x}\| \\ &\leq B (d_x \bar{\sigma}_\tau^2)^{1/2} + B |1 - \bar{\mu}_\tau| \|\boldsymbol{x}\| \leq \frac{B (2 d_x \tau)^{1/2}}{\min(1, c_f^{1/2})} + \frac{(2 c_f + 1) B}{\min(1, c_f)} \|\boldsymbol{x}\|. \end{aligned}$$

By (47),

$$\mathrm{E}_{\boldsymbol{x}(\kappa)|\boldsymbol{z}} \|g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \tau) - f(\boldsymbol{X}(\kappa), \boldsymbol{z})\|^2 \leq 2 \frac{\max((2 c_f + 1) B c_M, B (2 d_x)^{1/2})^2}{\min(1, c_f^2)} \tau^\alpha. \tag{50}$$

Plugging (49) and (50) into (36), we obtain

$$I_6 \leq \Big(\frac{2B}{\min(1, c_f^{\alpha+1})\underline{c}} + \frac{2B^2}{\min(1, c_f^2)\underline{c}^2}\Big)\mathrm{E}\|g(\boldsymbol{X}(\kappa), \boldsymbol{z}, \underline{\tau}) - f(\boldsymbol{X}(\kappa), \boldsymbol{z})\|^2 \leq c_{I_6}\underline{\tau}^{\alpha},$$

where $c_{I_6} = 10\frac{\max((2c_f+1)Bc_M, B(2d_x)^{1/2})^2}{\min(\min(1,c_f^{3+3\alpha})\underline{c}/B, \min(1,c_f^4)\underline{c}^2/B^2)}$. Combining the bounds for $I_3$ and $I_4$ through $I_5$ and $I_6$ yields the following result:

$$\begin{aligned}&\int_0^{\underline{\tau}} \mathrm{E}_{\boldsymbol{x}(\underline{\tau}), \boldsymbol{x}(\kappa)|\boldsymbol{z}}\|\nabla \log p_{\underline{\tau}}(\boldsymbol{X}(\underline{\tau})|\boldsymbol{z}) - \nabla \log p_{\kappa}(\boldsymbol{X}(\kappa)|\boldsymbol{z})\|^2 \mathrm{d}\kappa \\ &\leq \int_0^{\underline{\tau}} (c_{I_3}(\underline{\tau} - \kappa)^{\alpha} + c_{I_5}(\underline{\tau} - \kappa)^2 + c_{I_6}\underline{\tau}^{\alpha}\mathrm{d}\kappa \leq c\underline{\tau}^{1+\alpha}, \end{aligned} \tag{51}$$

with $c = \frac{1}{1+\alpha}(c_{I_3} + c_{I_5}) + c_{I_6}$. This completes the proof. ■

**Lemma 18** *Under Assumption 11 with $r > 1$, for any $\boldsymbol{z} \in [0,1]^{d_z}$, $\tau \geq 0$, and $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{d_x}$,*

$$\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}'|\boldsymbol{z})\|_\infty \leq c_1^g\|\boldsymbol{x} - \boldsymbol{x}'\|^\alpha + c_2^g\|\boldsymbol{x} - \boldsymbol{x}'\|, \tag{52}$$

*where $\alpha = \min(1, r-1)$, $c_1^g = \frac{B}{\min(1, C_f^{\alpha+1})\underline{c}}$ and $c_2^g = \left(\frac{B^2}{\min(1,c_f^2)\underline{c}^2} + \max(1, c_f)\right)$, with $B$ and $c_f$ defined in Assumption 11.*

**Proof** Note that for any $\boldsymbol{z} \in [0,1]^{d_z}$, $\tau \geq 0$, and $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^{d_x}$,

$$\begin{aligned}\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \nabla \log p_\tau(\boldsymbol{x}'|\boldsymbol{z})\|_\infty \leq &\left\|\frac{\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla g(\boldsymbol{x}', \boldsymbol{z}, \tau)}{g(\boldsymbol{x}, \boldsymbol{z}, \tau)}\right\|_\infty \\ &+ \|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \left|\frac{g(\boldsymbol{x}, \boldsymbol{z}, \tau) - g(\boldsymbol{x}', \boldsymbol{z}, \tau)}{g(\boldsymbol{x}, \boldsymbol{z}, \tau)g(\boldsymbol{x}', \boldsymbol{z}, \tau)}\right| + \frac{c_f}{(\mu_\tau^2 + c_f\sigma_\tau^2)}\|\boldsymbol{x} - \boldsymbol{x}'\|. \end{aligned} \tag{53}$$

By (37),

$$\begin{aligned}&\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau) - \nabla g(\boldsymbol{x}', \boldsymbol{z}, \tau)\|_\infty \\ &\leq \bar{\mu}_\tau \int \|\nabla f(\boldsymbol{y}, \boldsymbol{z}) - \nabla f(\boldsymbol{y} - \bar{\mu}_\tau(\boldsymbol{x} - \boldsymbol{x}'), \boldsymbol{z})\|_\infty p_N(\boldsymbol{y}; \bar{\mu}_\tau\boldsymbol{x}, \bar{\sigma}_\tau)\mathrm{d}\boldsymbol{y} \\ &\leq \bar{\mu}_\tau B\|\bar{\mu}_\tau(\boldsymbol{x} - \boldsymbol{x}')\|^\alpha \leq \bar{\mu}_\tau^{\alpha+1}B\|\boldsymbol{x} - \boldsymbol{x}'\|^\alpha. \end{aligned} \tag{54}$$

Similarly, $|g(\boldsymbol{x}, \boldsymbol{z}, \tau) - g(\boldsymbol{x}', \boldsymbol{z}, \tau)| \leq \bar{\mu}_\tau B\|\boldsymbol{x} - \boldsymbol{x}'\|$.

Finally, (54) together with the fact that $\|\nabla g(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \leq B\frac{\bar{\mu}_\tau}{\bar{\sigma}_\tau}\sqrt{\frac{\pi}{2}}$ in (43) and $g(\boldsymbol{x}, \boldsymbol{z}, \tau) \geq \underline{c} > 0$ in (35), leads to (52), as in the proof of Lemma 15. This completes the proof. ■

**Proof** [Theorem 13] It suffices to apply Proposition 1 with $l(\boldsymbol{Y};\gamma) = l(\boldsymbol{X},\boldsymbol{Z};\theta)$ for the diffusion settings to the excess risk

$$\rho^2(\theta^0,\theta) = \mathrm{E}_{\boldsymbol{x},\boldsymbol{z}}[l(\boldsymbol{x},\boldsymbol{z},\theta) - l(\boldsymbol{x},\boldsymbol{z},\theta^0)] = \int_{\underline{\tau}}^{\overline{\tau}} \mathrm{E}_{\boldsymbol{x}(\tau),\boldsymbol{z}} \|\nabla \log p_\tau(\boldsymbol{X}(\tau)|\boldsymbol{Z}) - \theta(\boldsymbol{X}(\tau),\boldsymbol{Z},\tau)\|^2 \mathrm{d}\tau.$$

Here, $\theta^0(\boldsymbol{x},\boldsymbol{z},\tau) = \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z})$ and $\theta \in \Theta$ is used to approximate $\theta^0$.

We first show that $l(\cdot;\theta)$ is bounded, satisfying the Bernstein condition with $c_b = O(\log^2 K)$. Note that $l(\boldsymbol{x}(0),\boldsymbol{z};\theta) = \int_{\underline{\tau}}^{\overline{\tau}} \int_{\mathbb{R}^{d_x}} \|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0),\boldsymbol{z}) - \theta(\boldsymbol{x},\boldsymbol{z},\tau)\|^2 p_\tau(\boldsymbol{x}|(\boldsymbol{x}(0),\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\tau \geq 0$. Further, $\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0),\boldsymbol{z}) - \theta(\boldsymbol{x},\boldsymbol{z},\tau)\|^2 \leq 2(\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0),\boldsymbol{z})\|^2 + \|\theta(\boldsymbol{x},\boldsymbol{z},\tau)\|^2)$. By the assumption of Lemma 2 that $\sup_{\boldsymbol{x},\boldsymbol{z}} \|\theta(\boldsymbol{x},\boldsymbol{z},\tau)\| \leq c_B \frac{\sqrt{\log K}}{\sigma_\tau}$ and (45),

$$\begin{aligned} l(\boldsymbol{x}(0),\boldsymbol{z};\theta) &\leq \int_{\underline{\tau}}^{\overline{\tau}} \frac{2d_x}{\sigma_\tau^2}\mathrm{d}\tau + \int_{\underline{\tau}}^{\overline{\tau}} \frac{2d_x c_B^2 \log K}{\sigma_\tau^2}\mathrm{d}\tau \\ &\leq 2d_x \int_{\underline{\tau}}^{\overline{\tau}} \frac{(c_B^2 \log K + 1)}{\sigma_\tau^2}\mathrm{d}\tau \leq 2d_x \int_{\underline{\tau}}^{\overline{\tau}} \frac{(c_B^2 \log K + 1)}{(1-e^{-1})\min(1,2\tau)}\mathrm{d}\tau \leq c_b^* \log^2 K, \end{aligned} \tag{55}$$

where $c_b^* = 2\frac{d_x(c_B^2 \log K + 1)(c_{\overline{\tau}} + c_{\underline{\tau}})}{1-e^{-1}}$.

To show that $l(\boldsymbol{x}(0),\boldsymbol{z};\theta^0)$ is bounded, where $\theta^0 = \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z})$ may not necessarily belong to $\Theta$, we consider $\boldsymbol{x}(0)$ within the range $[-c_R\sqrt{\log K}, c_R\sqrt{\log K}]^{d_x}$. This is achieved by employing the truncation at $c_R \log K$ for the unbounded density $p_\tau(\boldsymbol{x}|\boldsymbol{z})$. By (34), (35), and (43), $\|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z})\| \leq \frac{c_f}{(\bar{\mu}_\tau^2 + c_f\bar{\sigma}_\tau^2)}\|\boldsymbol{x}\| + \sqrt{\frac{d_x\pi}{2}}\frac{B\bar{\mu}_\tau}{\underline{c}\bar{\sigma}_\tau}$. By the Cauchy–Schwarz inequality,

$$\begin{aligned} &l(\boldsymbol{x}(0),\boldsymbol{z};\theta^0) = \int_{\underline{\tau}}^{\overline{\tau}} \int_{\mathbb{R}^{d_x}} \|\nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{x}(0),\boldsymbol{z}) - \theta^0(\boldsymbol{x},\boldsymbol{z},\tau)\|^2 p_\tau(\boldsymbol{x}|(\boldsymbol{x}(0),\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\tau \\ &\leq \int_{\underline{\tau}}^{\overline{\tau}} \frac{2d_x}{\sigma_\tau^2}\mathrm{d}\tau + 2\int_{\underline{\tau}}^{\overline{\tau}} \int_{\mathbb{R}^{d_x}} 2\left[\left(\sqrt{d_x\frac{\pi}{2}}\frac{B\bar{\mu}_\tau}{\underline{c}\bar{\sigma}_\tau}\right)^2 + \left(\frac{c_f\|\boldsymbol{x}\|}{\mu_\tau^2 + c_f\sigma_\tau^2}\right)^2\right] p_\tau(\boldsymbol{x}|(\boldsymbol{x}(0),\boldsymbol{z}))\mathrm{d}\boldsymbol{x}\mathrm{d}\tau \\ &\leq 2\int_{\underline{\tau}}^{\overline{\tau}} \frac{d_x(\frac{\pi B^2}{\underline{c}^2}\max(1,1/c_f)+1)}{\sigma_\tau^2}\mathrm{d}\tau + 4\int_{\underline{\tau}}^{\overline{\tau}} \max(1,c_f^2)(\mu_\tau\|\boldsymbol{x}(0)\|^2 + d_x\sigma_\tau^2)\mathrm{d}\tau \leq c_b^0 \log^2 K, \end{aligned} \tag{56}$$

where $c_b^0 = 2\frac{d_x(\frac{\pi B^2}{\underline{c}^2}\max(1,1/c_f)+1)(c_{\overline{\tau}}+c_{\underline{\tau}})}{1-e^{-1}} + 4\max(1,c_f^2)d_x c_R + 4d_x c_{\overline{\tau}}$.

Next, we verify the variance condition. By the Cauchy-Schwarz inequality,

$$\begin{aligned}
&l(\boldsymbol{x},\boldsymbol{z},\theta)-l(\boldsymbol{x},\boldsymbol{z},\theta^0)\\
&\le\Bigg[\int_{\underline{\tau}}^{\overline{\tau}}\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}\|2\nabla\log p_\tau(\boldsymbol{X}|\boldsymbol{x}(0),\boldsymbol{z})-\theta(\boldsymbol{X},\boldsymbol{z},\tau)-\theta^0(\boldsymbol{X},\boldsymbol{z},\tau)\|^2\mathrm{d}\tau\\
&\qquad\int_{\underline{\tau}}^{\overline{\tau}}\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}\|\theta(\boldsymbol{X},\boldsymbol{z},\tau)-\theta^0(\boldsymbol{X},\boldsymbol{z},\tau)\|^2\mathrm{d}\tau\Bigg]\\
&\le\sup_{\boldsymbol{x},\boldsymbol{z}}(l(\boldsymbol{x},\boldsymbol{z},\theta)+l(\boldsymbol{x},\boldsymbol{z},\theta^0))\int_{\underline{\tau}}^{\overline{\tau}}\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0)}\|\theta(\boldsymbol{X},\boldsymbol{z},\tau)-\theta^0(\boldsymbol{X},\boldsymbol{z},\tau)\|^2\mathrm{d}\tau.
\end{aligned}$$

By (55) and (56) and the sub-Gaussian property of $p_x^0$ with a sufficiently large $c_R$, we have

$$\begin{aligned}
&\mathrm{Var}_{\boldsymbol{x}(0),\boldsymbol{z}}(l(\boldsymbol{X}(0),\boldsymbol{Z};\theta)-l(\boldsymbol{X}(0),\boldsymbol{Z};\theta^0))\le\mathrm{E}_{\boldsymbol{x}(0),\boldsymbol{z}}(l(\boldsymbol{X}(0),\boldsymbol{Z};\theta)-l(\boldsymbol{X}(0),\boldsymbol{Z};\theta^0))^2\\
\le&\sup_{\|\boldsymbol{x}\|_\infty\le c_R\sqrt{\log K},\boldsymbol{z}}(l(\boldsymbol{x},\boldsymbol{z};\theta)+l(\boldsymbol{x},\boldsymbol{z};\theta^0))\int_{\underline{\tau}}^{\overline{\tau}}\mathrm{E}_{\boldsymbol{x},\boldsymbol{z}}\mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}}\|\theta(\boldsymbol{X},\boldsymbol{Z},\tau)-\theta^0(\boldsymbol{X},\boldsymbol{Z},\tau)\|^2\mathrm{d}\tau\\
\le&(c_b^0+c_b^*)(\log^2K)\rho^2(\theta^0,\theta),
\end{aligned}$$

This implies Assumption 9 with $c_v=O(\log^2K)$.

By Lemma 14, the approximation error is $\delta_n=O(K^{-\frac{r}{d_x+d_z}}\log^{\frac{r}{2}+1}K)$. By Lemma 16, there exists a positive $c_H$ such that $H(u,\mathcal{F})\le c_HK\log^{14}K\log\frac{Kn}{u}$. Then,

$$\int_{k\varepsilon^2/16}^{4c_v^{1/2}\varepsilon}H_B^{1/2}(u,\mathcal{F})\mathrm{d}u\le c_H\int_{k\varepsilon^2/16}^{4c_v^{1/2}\varepsilon}K\log^{16}K\log\frac{K}{u}\mathrm{d}u\le 4c_Hc_v^{1/2}\varepsilon\sqrt{K\log^{16}K\log\frac{K}{\varepsilon^2}}.$$

Solving the inequality for $\beta_n$: $4c_Hc_v^{1/2}\beta_n\sqrt{K\log^{16}K\log\frac{K}{\varepsilon^2}}\le c_h\sqrt{n}\beta_n^2$ with $c_v=2(c_b^*+c_b^0)\log^2K$ yields $\beta_n=2\frac{4c_H\sqrt{2(c_b^*+c_b^0)}}{c_h}\sqrt{\frac{K\log^{19}K}{n}}$.

Let $\varepsilon_n=\beta_n+\delta_n$ with $\beta_n\asymp\sqrt{\frac{K\log^{19}K}{n}}$ and $\delta_n\asymp K^{-\frac{r}{d_x+d_z}}\log^{\frac{r}{2}+1}K$ so that $\varepsilon_n$ satisfies the conditions of Proposition 1. Omitting the logarithmic term, let $\beta_n=\delta_n$ by $K\asymp n^{-\frac{d_x+d_z}{d_x+d_z+2r}}$ and this yields the optimal rate $\varepsilon_n\asymp n^{\frac{r}{d_x+d_z+2r}}\log^m n$ with $m=\max(\frac{19}{2},\frac{r}{2}+1)$. Note that $c_e\asymp\frac{1}{c_v^2}\asymp\log^4K$ in Proposition 1. With $\log K=O(\log n)$, for some constant $c_e>0$, $P(\rho(\theta^0,\hat\theta)\ge\varepsilon_n)\le 4\exp(-c_en^{1-\xi}\varepsilon_n^2)$, where $0<\xi<1$ is small such that $\frac{\log^2n}{n^\xi}=o(1)$ and $1-\xi-\frac{2r}{d_x+d_z+2r}>0$, implying that $\rho(\theta^0,\hat\theta)=O_p(\varepsilon_n)$ as $n^{1-\xi}\varepsilon_n^2\to\infty$ as $n\to\infty$. By Lemma 19, the desired result follows from (57). This completes the proof. ■

**Lemma 19 (KL divergence and score matching loss)** *Under the assumptions and settings in Theorem 13, suppose that the excess risk is bounded in that $\rho(\theta^0,\hat\theta)\le\varepsilon_n$ with high probability. Then, the diffusion generation errors under the total variation distance and the square root of the Kullback-Leibler divergence are also bounded by $\varepsilon_n$ with constants $c_{TV}$ and $c_{KL}$, respectively:*

$$\mathrm{E}_{\boldsymbol{z}}[\mathrm{TV}(p^0_{\boldsymbol{x}|\boldsymbol{z}},\hat p_{\boldsymbol{x}|\boldsymbol{z}})]\le c_{TV}\varepsilon_n,\ \textit{if}\ r>0;\ \mathrm{E}_{\boldsymbol{z}}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}|\boldsymbol{z}},\hat p_{\boldsymbol{x}|\boldsymbol{z}})]\le c_{KL}\varepsilon_n,\ \textit{if}\ r>1. \tag{57}$$

**Proof** The first inequality in (57) follows from Lemma D.5 in Fu et al. (2024). For the second inequality, by Girsanov's Theorem and Proposition C.3 in Chen et al. (2023a), the KL divergence can be bounded by the diffusion approximation to the standard Gaussian and score matching:

$$\begin{aligned}
&\mathcal{K}(p^0_{\boldsymbol{x}|\boldsymbol{z}}, \hat{p}_{\boldsymbol{x}|\boldsymbol{z}}) \leq \mathcal{K}(p_{\overline{\tau}}, p_N) + I_1(\boldsymbol{z}) + I_2(\boldsymbol{z}),\\
&I_1(\boldsymbol{z}) = \int_{\underline{\tau}}^{\overline{\tau}} \frac{1}{2} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{z}} \left\| \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \hat{\theta}(\boldsymbol{x}, \boldsymbol{z}, \tau) \right\|^2 \mathrm{d}\tau,\\
&I_2(\boldsymbol{z}) = \int_0^{\underline{\tau}} \frac{1}{2} \mathrm{E}_{\boldsymbol{x}(\tau), \boldsymbol{x}(\underline{\tau})|\boldsymbol{z}} \left\| \nabla \log p_\tau(\boldsymbol{x}|\boldsymbol{z}) - \hat{\theta}(\boldsymbol{x}(\underline{\tau}), \boldsymbol{z}, \underline{\tau}) \right\|^2 \mathrm{d}\tau,
\end{aligned} \tag{58}$$

where $p_N$ denotes the standard $d_x$-dimensional Gaussian density.

By Lemma C.4 of Chen et al. (2023a), $\mathcal{K}(p_{\overline{\tau}}, p_N) \leq (d_x + \mathrm{E}_{\boldsymbol{x}} \|\boldsymbol{X}\|^2) \exp(-\overline{\tau})$ , which is bounded through the exponential convergence of the forward diffusion process. The second term is bounded by the $L_2$-distance of the score function $\mathrm{E}_{\boldsymbol{z}} I_1(\boldsymbol{z}) = O(\rho^2(\theta^0, \hat{\theta}))$.

To bound $E_{\boldsymbol{z}} I_2(\boldsymbol{z})$, by the pointwise $L_2$-distance for $\tau$ in Theorem 3.4 of Fu et al. (2024), we obtain that

$$\int_{\mathbb{R}^{d_x}} p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) \|\nabla \log p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) - \pi\theta_0(\boldsymbol{x}, \boldsymbol{z}, \underline{\tau})\|^2 \mathrm{d}\boldsymbol{x} = O(\frac{1}{\sigma^2_{\underline{\tau}}} K^{-\frac{r}{d_x + d_z}} \log^{\frac{r}{2}+1} K) = O(\frac{1}{\sigma^2_{\underline{\tau}}} \varepsilon^2).$$

To get the estimation error for the point score matching loss at $\underline{\tau}$, we repeat the process to bound $\beta_n$ in the proof of Theorem 13 while modifying the integral upper bounds in (55) and (56) by replacing $c_b^*$ with $\frac{c_b^*}{\sigma^2_{\underline{\tau}}}$ and $c_b^0$ with $\frac{c_b^0}{\sigma^2_{\underline{\tau}}}$.

Moreover, together with the trade-off in the setting, we can bound the total error in the same order of the approximation error, for some constant $c > 0$,

$$\int_{\mathbb{R}^{d_x}} p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) \|\nabla \log p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) - \hat{\theta}(\boldsymbol{x}, \boldsymbol{z}, \underline{\tau})\|^2 \mathrm{d}\boldsymbol{x} = O(\frac{1}{\sigma^2_{\underline{\tau}}} \varepsilon^2).$$

Then

$$\begin{aligned}
\mathrm{E}_{\boldsymbol{z}} I_2(\boldsymbol{z}) \leq & \frac{\underline{\tau}}{2} \mathrm{E}_{\boldsymbol{z}} \int_{\mathbb{R}^{d_x}} p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) \|\nabla \log p_{\underline{\tau}}(\boldsymbol{x}|\boldsymbol{z}) - \hat{\theta}(\boldsymbol{x}, \boldsymbol{z}, \underline{\tau})\|^2 \mathrm{d}\boldsymbol{x}\\
& + \mathrm{E}_{\boldsymbol{z}} \int_0^{\underline{\tau}} \frac{1}{2} \mathrm{E}_{\boldsymbol{x}(\tau), \boldsymbol{x}(\underline{\tau})|\boldsymbol{z}} \|\nabla \log p_\tau(\boldsymbol{x}(\tau)|\boldsymbol{z}) - \log p_{\underline{\tau}}(\boldsymbol{x}(\underline{\tau})|\boldsymbol{z})\|^2 \mathrm{d}\tau\\
= & O(\frac{\underline{\tau}}{\sigma^2_{\underline{\tau}}} \varepsilon^2) + O(\underline{\tau}^{1+\alpha}).
\end{aligned}$$

Finally, in (58), by choosing $\underline{\tau} = K^{-c_{\underline{\tau}}}$ and $\overline{\tau} = c_{\overline{\tau}} \log K$ with sufficiently large $c_{\underline{\tau}}$ and $c_{\overline{\tau}}$ such that $\exp(-\overline{\tau}) + \underline{\tau}^{1+\alpha} \leq \varepsilon_n^2$, we obtain $\mathcal{K}(p^0_{\boldsymbol{x}|\boldsymbol{z}}, \hat{p}_{\boldsymbol{x}|\boldsymbol{z}}) = O(\varepsilon_n^2)$ with high probability. This completes the proof. ■

### A.2.2 PROOFS FOR SUBSECTION 3.2

Theorem 20 gives the non-asymptotic probability bound for the generation error in Theorem 1.

**Theorem 20 (Conditional diffusion via transfer learning)** *Under Assumptions 1-3 and 5, there exists a wide ReLU network $\Theta_t$ in* (8)*, with specified hyperparameters: $\mathbb{L}_t = c_L \log^4 K$, $\mathbb{W}_t = c_W K \log^7 K$, $\mathbb{S}_t = c_S K \log^9 K$, $\log \mathbb{B}_t = c_B \log K$, $\log \mathbb{E}_t = c_E \log^4 K$, such that the error in conditional diffusion generation via transfer learning, as described in Section 3.2, with stopping criteria: $\log \underline{\tau}_t = -c_{\underline{\tau}_t} \log K$ and $\overline{\tau}_t = c_{\overline{\tau}_t} \log K$ in* (4) *and* (5)*, is given by: for any $x \geq 1$,*

$$\mathrm{E}_{\boldsymbol{z}_t}[\mathrm{TV}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\varepsilon_t + \sqrt{3c_1}\varepsilon_s), r_t > 0;$$
$$\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{\frac{1}{2}}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\varepsilon_t + \sqrt{3c_1}\varepsilon_s), r_t > 1,$$

*with a probability exceeding $1 - \exp(-c_5 n_t^{1-\xi}(x\varepsilon_t)^2) - \exp\left(-c_2 n_s^{1-\xi}(x\varepsilon_s)^2\right)$ for some constant $c_5 > 0$. Here $c_L, c_W, c_S, c_B, c_E, c_{\overline{\tau}_t}, c_{\underline{\tau}_t}$ are sufficiently large constants, $K \asymp n_t^{\frac{dx_t + d_{h_t}}{dx_t + d_{h_t} + 2r_t}}$, $\varepsilon_t \asymp n_t^{-\frac{r_t}{dx_t + d_{h_t} + 2r_t}} \log^{m_t} n_t$, and $m_t = \max(\frac{19}{2}, \frac{r_t}{2} + 1)$, with $\asymp$ denoting mutual boundedness.*

**Proof** [Theorems 1 and 20] If the probability bound in Theorem 20 holds, we obtain the rate in expectation in Theorem 1. Specifically, let TV be $\mathrm{E}_{\boldsymbol{z}_t}[\mathrm{TV}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})]$. Then,

$$\begin{aligned}\mathbb{E}_{\mathcal{D}} \frac{\mathrm{TV}}{\varepsilon_t} &= \frac{1}{\varepsilon_t}\left(\mathbb{E}_{\mathcal{D}} \mathrm{TV}\mathbb{I}(\mathrm{TV} \leq \varepsilon_t) + \mathbb{E}_{\mathcal{D}} \mathrm{TV}\mathbb{I}(\mathrm{TV} > \varepsilon_t)\right) \\ &\leq 1 + \left(\int_1^\infty P(\mathrm{TV} > x\varepsilon_t)\mathrm{d}x + \frac{1}{\varepsilon_t} P(\mathrm{TV} > \varepsilon_t)\right) \leq 2.\end{aligned}$$

The last inequality holds by the fact that $\xi$ can be small enough. The rate of expected KL divergence can be proved in a similar way.

To prove Theorem 20, we first show that, under the conditions of Theorem 1, the error bound for the excess risk holds, for any $x > 1$,

$$\rho_t(\gamma_t^0, \hat{\gamma}_t) \leq x(\beta_t + \delta_t + \sqrt{3c_1}\varepsilon_s),$$

with a probability exceeding $1 - \exp(-c_5 n_t^{1-\xi}(x(\beta_t + \delta_t))^2) - \exp\left(-c_2 n_s^{1-\xi}(x\varepsilon_s)^2\right)$. Here, the estimation and approximation errors $\beta_t$ and $\delta_t$ are $\beta_t \asymp \sqrt{\frac{K \log^{19} K}{n_t}}$ and $\delta_t \asymp K^{-\frac{r_t}{d_{x_t} + d_{h_t}}} \log^{\frac{r_t}{2}+1} K$. This bound is proved using Theorem 9 with the approximation error and the entropy bounds obtained in Lemmas 14 and 16. The two assumptions on the loss function in Theorem 9 can be verified similarly as in the proof of Theorem 13. Hence, we only need to obtain the approximation error $\delta_t$ and the estimation error $\beta_t$.

Recall the definition of $\delta_t$, $\delta_t = \inf_{\theta_t \in \Theta_t} \mathrm{E}[l_j(\cdot; \theta_t, h^0) - l_j(\cdot; \theta_t^0, h^0)]$. By Assumptions 5, we bound $\delta_t$ by Lemma 14 by replacing $\boldsymbol{z}$ with the $d$-dimensional $h^0(\boldsymbol{z})$: there exists a ReLU network $\mathrm{NN}_t(\mathbb{L}_t, \mathbb{W}_t, \mathbb{S}_t, \mathbb{B}_t, \mathbb{E}_t)$ with depth $\mathbb{L}_t \asymp \log^4 K$, width $\mathbb{W}_t \asymp K \log^7 K$, effective parameter number $\mathbb{S}_t \asymp K \log^9 K$, parameter bound $\log \mathbb{E}_t \asymp \log^4 K$, and $\sup_{\boldsymbol{x},\boldsymbol{z}} \|\theta_t(\boldsymbol{x}, \boldsymbol{z}, \tau)\|_\infty \asymp \sqrt{\log K}/\sigma_\tau$, such that $\rho(\theta_t^0, \pi\theta_t^0) = O(K^{-\frac{r}{d_{x_t} + d_{h_t}}} \log^{r_t/2+1} K)$. For the estimation error $\beta_t$, we apply Lemma 16 by replacing the parameters of $\mathcal{F}$ by those of $\mathcal{F}_t$.

Setting $\beta_t = \delta_t$ for $K$, ignoring the logarithmic term, yields $\beta_t = \delta_t \asymp n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{m_t} n_t$ with optimal $K \asymp n_t^{\frac{d_{x_t}+d_{h_t}}{d_{x_t}+d_{h_t}+2r_t}}$ and $m_t = \max(\frac{19}{2}, \frac{r_t}{2}+1)$. Thus, the best bound is $\varepsilon_t \asymp n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{m_t} n_t$. ■

For the source task, we use the conditional diffusion model to learn $p_{\boldsymbol{x}_s|\boldsymbol{z}_s}$. This can be done similarly as estimating $\theta_t$ for $p_{\boldsymbol{x}_t|\boldsymbol{z}_t}$. Given $\Theta_s = \{\theta_s(\boldsymbol{x}, h(\boldsymbol{z}), \tau), \theta_s \in \mathrm{NN}_s(\mathbb{L}_s, \mathbb{W}_s, \mathbb{S}_s, \mathbb{B}_s, \mathbb{E}_s, \lambda_s)\}$, we define the loss as

$$\begin{aligned} L_s(\theta_s, h) &= \sum_{i=1}^{n_s} l_s(\boldsymbol{x}_s^i, \boldsymbol{z}_s^i; \theta_s, h) \\ &= \sum_{i=1}^{n_s} \int_{\underline{\tau}_s}^{\overline{\tau}_s} \mathrm{E}_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0),\boldsymbol{z}_s} \|\nabla \log p_{\boldsymbol{x}(\tau)|\boldsymbol{x}(0),\boldsymbol{z}_s}(\boldsymbol{X}(\tau)|\boldsymbol{x}_s^i, \boldsymbol{z}_s^i) - \theta_s(\boldsymbol{X}(\tau), h_s(\boldsymbol{z}_s^i), \tau)\|^2 \mathrm{d}\tau, \end{aligned}$$

which yields that $\hat{\theta}_s(\boldsymbol{x}, \hat{h}(\boldsymbol{z}), \tau) = \arg\min_{\theta_s \in \Theta_s, h \in \Theta_h} L_s(\theta_s, h)$.

To give the bound of $\varepsilon_s$ in Section 3, we make some assumptions similar to those for $p_{\boldsymbol{x}_t|\boldsymbol{z}_t}$.

**Assumption 12 (Source density)** *Assuming the true conditional density of $\boldsymbol{X}_s$ given $\boldsymbol{Z}$ is expressed as $p^0_{\boldsymbol{x}_s|\boldsymbol{z}_s}(\boldsymbol{x}|\boldsymbol{z}_s) = \exp(-c_6\|\boldsymbol{x}\|^2/2) \cdot f_s(\boldsymbol{x}, h_s^0(\boldsymbol{z}_s))$, where $f_s$ is a non-negative function and $c_6 > 0$ is a constant. Additionally, $f_s$ belongs to a Hölder ball $\mathcal{C}^{r_s}(\mathbb{R}^{d_{x_s}} \times [0,1]^{d_{h_s}}, \mathbb{R}^{d_{x_s}}, B_s)$ and is lower bounded away from zero.*

**Lemma 21 (Source error)** *Under Assumption 12, setting a network $\Theta_s$'s hyperparameters with sufficiently large constants $\{c_L, c_W, c_S, c_B, c_E, c_\lambda, c_{\underline{\tau}_s}, c_{\overline{\tau}_s}\}$: $\mathbb{L}_s = c_L \log^4 K$, $\mathbb{W}_s = c_W K \log^7 K$, $\mathbb{S}_s = c_S K \log^9 K$, $\log \mathbb{B}_s = c_B \log K$, $\log \mathbb{E}_s = c_E \log^4 K$, and $\lambda_s = c_\lambda$ for $r > 1$, $\lambda_s = c_\lambda/\sigma_{\underline{\tau}}$ for $r \le 1$, with diffusion stopping criteria from* (4)-(5) *as $\log \underline{\tau}_s = -c_{\underline{\tau}_s} \log K$ and $\overline{\tau}_s = c_{\overline{\tau}_s} \log K$, we obtain that, for any $x \ge 1$,*

$$P(\rho_s(\gamma_s^0, \hat{\gamma}_s) \ge x\varepsilon_s) \le \exp(-c_2 n_s^{1-\xi}(x\varepsilon_s)^2),$$

*with $\varepsilon_s = \beta_s + \delta_s + \varepsilon_s^h$, some constant $c_2 > 0$ and a small $\xi > 0$ same in Assumption 3. Here, $\beta_s$ and $\delta_s$ are given by: $\beta_s \asymp \sqrt{\frac{K \log^{19} K}{n_s}}$, $\quad \delta_s \asymp K^{-\frac{r_s}{d_{x_s}+d_{h_s}}} \log^{r_s/2+1} K$. $\varepsilon_s^h$ satisfies* (59) *in Lemma 23. Setting $K \asymp n_s^{\frac{d_{x_s}}{d_{x_s}+d_{h_s}+2r_s}}$ yields $\varepsilon_s \asymp n_s^{-\frac{r_s}{d_{x_s}+d_{h_s}+2r_s}} \log^{m_s} n_s + \varepsilon_s^h$, where $m_s = \max(\frac{19}{2}, \frac{r_s}{2}+1)$.*

**Proof** When $h_0 \in \Theta_h$, $\delta_s \le \delta_s(h^0)$. Hence, the upper bound can be given by $\delta_s(h^0)$ which is given by Lemma 14. Note that with the Hölder continuity in $\Theta$, the statistical error can be decomposed into $\beta_s + \varepsilon_s^h$ through Lemma 23. The bound of $\varepsilon_s^h$ is given by (59) in Lemma 23 and the bound of $\varepsilon_s$ is derived in the same way as $\varepsilon_t$ in the proof of Theorem 13. ■

Theorem 22 gives the non-asymptotic probability bound for the generation error in Theorem 2.

**Theorem 22 (Non-transfer conditional diffusion)** *Under Assumption 5, a wide ReLU network $\tilde{\Theta}_t$, as described in* (8) *and with the same configuration as $\Theta_t$ from Theorem 1, exists, with an*

*additional constraint on* $\tilde{\Theta}_t$*:* $\lambda_t = c_\lambda$ *for* $r > 1$ *and* $\lambda_t = c_\lambda/\sigma_{\mathcal{T}}$ *for* $r \leq 1$*, provided* $c_\lambda$ *is sufficiently large. Then, the generation error of the non-transfer conditional diffusion model, adhering to the same stopping criteria from Theorem 1, is given by: for any* $x \geq 1$

$$\mathrm{E}_{\boldsymbol{z}_t}[\mathrm{TV}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\tilde{\varepsilon}_t + \varepsilon_t^h), \text{ if } r > 0,$$
$$\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\tilde{\varepsilon}_t + \varepsilon_t^h), \text{ if } r > 1,$$

*with the target probability exceeding* $1 - \exp(-c_5 n_t^{1-\xi}(x(\tilde{\varepsilon}_t + \varepsilon_t^h))^2)$ *for some constant* $c_5 > 0$*. Here,* $\tilde{\varepsilon}_t \asymp n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{m_t} n_t$ *while* $\varepsilon_t^h$ *is defined by* (59) *in Lemma 23.*

**Proof** [Theorems 2 and 22] Theorem 22 can be proved similarly to Lemma 21 by replacing $r_s$ with $r_t$, $n_s$ with $n_t$ and $d_{x_s}$ with $d_{x_t}$. ■

Next, we give the metric entropy inequalities to the class of composite functions $\Theta_j \circ \Theta_h = \{\theta_j(\boldsymbol{x}_j, h_j(\boldsymbol{z}_j)), h_j(\boldsymbol{z}_j) = (h(\boldsymbol{z}), \boldsymbol{z}_{j^c}), h \in \theta_h, \theta_j \in \Theta_j\}$, $j \in \{s, t\}$.

**Lemma 23** *When* $\sup_{\theta_j \in \Theta_j} \sup_{\boldsymbol{x},\tau,\boldsymbol{y}\neq\boldsymbol{y}'} \frac{\|\theta_j(\boldsymbol{x},\boldsymbol{y},\tau) - \theta_j(\boldsymbol{x},\boldsymbol{y}',\tau)\|}{\|\boldsymbol{y}-\boldsymbol{y}'\|^{\alpha_j}} \leq \lambda_j$*, for* $j \in \{s, t\}$*,*

$$\int_{k_j(\varepsilon_j^h)^2/16}^{4c_{vj}^{1/2}(\varepsilon_j^h)} H^{1/2}\left(\frac{u^{\frac{1}{\alpha_j}}}{2\lambda_{l_j}\lambda_j (d_{\boldsymbol{z}})^{\frac{\alpha_j}{2}}}, \Theta_h\right) \mathrm{d}u \leq c_{h_j} n_j^{1/2} (\varepsilon_j^h)^2, \tag{59}$$

*where* $\lambda_{l_j}$ *is the Lipschitz norm of* $l_j$ *with respect to* $\theta_j$ *and* $\lambda_j$ *and* $\alpha_j$ *are the parameter settings in the neural network class.*

**Proof** For $\gamma_j, \gamma_j' \in \Theta_j \circ \Theta_h$,

$$\begin{aligned}
\sup_{\boldsymbol{x}_j,\boldsymbol{z}_j} |l_j(\boldsymbol{x}_j, \boldsymbol{z}_j; \gamma_j) - l_j(\boldsymbol{x}_j, \boldsymbol{z}_j; \gamma_j')| &\leq \sup_{\boldsymbol{x}_j,\boldsymbol{z}_j} \lambda_{l_j} \|\theta_j(\boldsymbol{x}_j, h_j(\boldsymbol{z}_j)) - \theta_j'(\boldsymbol{x}_j, h_j'(\boldsymbol{z}_j))\|_\infty \\
&\leq \sup_{\boldsymbol{x}_j,\boldsymbol{z}_j} \lambda_{l_j} \|\theta_j(\boldsymbol{x}_j, h_j(\boldsymbol{z}_j)) - \theta_j'(\boldsymbol{x}_j, h_j(\boldsymbol{z}_j))\|_\infty + \sup_{\boldsymbol{x}_j,\boldsymbol{z}_j} \lambda_{l_j} \|\theta_j'(\boldsymbol{x}_j, h_j(\boldsymbol{z}_j)) - \theta_j'(\boldsymbol{x}_j, h_j'(\boldsymbol{z}_j))\|_\infty \\
&\leq \sup_{\boldsymbol{x}_j,\boldsymbol{y}} \lambda_{l_j} \|\theta_j(\boldsymbol{x}_j, \boldsymbol{y}) - \theta_j'(\boldsymbol{x}_j, \boldsymbol{y})\|_\infty + \sup_{\boldsymbol{z}} \lambda_{l_j}\lambda_j d_{\boldsymbol{z}}^{\frac{\alpha_j}{2}} \|h(\boldsymbol{z}) - h'(\boldsymbol{z})\|_\infty^{\alpha_j}.
\end{aligned}$$

The last inequality follows from the Hölder continuity of $\theta$ and the definition of $h_j$, $h_j(\boldsymbol{z}_j) = (h(\boldsymbol{z}), \boldsymbol{z}_{j^c})$ for $j \in \{s, t\}$. Hence, $H_B^{1/2}(u, \mathcal{F}_s) \leq H^{1/2}(c_\theta' u, \Theta) + H^{1/2}(c_h' u^{\frac{1}{\alpha_j}}, \Theta_h)$, where $c_{\theta_s}' = \frac{1}{2\lambda_{l_s}}$ and $c_h' = \frac{1}{2\lambda_{l_s}\lambda_s (d_{\boldsymbol{z}})^{\frac{\alpha_j}{2}}}$. So, the integral inequality (18) is solved by $\beta_s$ defined in Lemma 21 and $\varepsilon_s^h$ in (59). The same holds for $\tilde{\mathcal{F}}_t$ in (20) with $\beta_t$ in the proof of Theorem 22 and $\varepsilon_t^h$.

By the proof of Lemma 16, we have $\lambda_{l_j} = c^* \log^{3/2}(n_j)$ in the conditional diffusion models. ■

### A.2.3 PROOFS OF SUBSECTION 3.3

We first give the formal version of Theorems 3 and 4 in non-asymptotic probability bounds.

**Theorem 24 (Unconditional diffusion via transfer learning)** *Under Assumptions 6 and 4, there exists a ReLU network* $\Theta_{g_t}$ *with hyperparameters:* $\mathbb{L}_g = c_L L \log L$*,* $\mathbb{W}_g = c_W W \log W$*,* $\mathbb{S}_g =$

$c_S W^2 L \log^2 W \log L$, $\mathbb{B}_g = c_B$, *and* $\log \mathbb{E}_g = c_E \log(WL)$, *such that the error for unconditional diffusion generation via transfer learning in the Wasserstein distance is* $\mathcal{W}(p^0_{\boldsymbol{x}}, \hat{p}_{\boldsymbol{x}}) \leq x(\varepsilon_t + \varepsilon^u_s)$ *with a probability exceeding* $1 - \exp(-c_3 n_s^{1-\xi}(x\varepsilon^u_s)^2) - \exp(-c_7 n_t (x\varepsilon_t)^2)$, *for any* $x \geq 1$ *and some constant* $c_7 > 0$. *Here,* $\{c_L, c_W, c_S, c_B, c_E\}$ *are sufficiently large positive constants,* $WL \asymp n_t^{\frac{d_u}{2(d_u+2r_g)}}$, $\varepsilon_t \asymp n_t^{-\frac{r_g}{d_u+2r_g}} \log^{m_g} n_t$, *and* $m_g = \max(\frac{5}{2}, \frac{r_g}{2})$.

**Theorem 25 (Non-transfer via unconditional diffusion)** *Suppose there exists a sequence* $\varepsilon^u_t$ *indexed by* $n_t$ *such that* $n_t^{1-\xi}(\varepsilon^u_t)^2 \to \infty$ *as* $n_s \to \infty$ *and* $P(\rho_u(\theta^0_u, \tilde{\theta}_u) \geq \varepsilon) \leq \exp(-c_3 n_t^{1-\xi}\varepsilon^2)$ *for any* $\varepsilon \geq \varepsilon^u_t$ *and some constants* $c_3, \xi > 0$. *Under Assumption 6 and the same settings for* $\Theta_{g_t}$ *of Theorem 3, the generation error of the non-transfer unconditional diffusion model satisfies: for any* $x \geq 1$ $\mathcal{W}(P^0_{\boldsymbol{x}_t}, \tilde{P}_{\boldsymbol{x}_t}) \leq x(\varepsilon_t + \varepsilon^u_t)$, *with a probability exceeding* $1 - \exp(-c_3 n_t^{1-\xi}(x\varepsilon^u_t)^2) - \exp(-c_7 n_t (x\varepsilon_t)^2)$. *Here* $\varepsilon_t \asymp n_t^{-\frac{r_g}{d_u+2r_g}} \log^{m_g} n_t$.

Referencing the definition of excess risk, let $\gamma_t = (g_t, \theta_u)$. Define the excess risks as $\rho^2_{g_t}(g^0_t, g_t) = \mathrm{E}_{\boldsymbol{u}}(l_{g_t}(\boldsymbol{U}, \boldsymbol{X}; g_t) - l_{g_t}(\boldsymbol{U}, \boldsymbol{X}; g^0_t))$ and $\rho^2_u(\theta_u, \theta^0_u) = \mathrm{E}_{\boldsymbol{u}}(l_u(\boldsymbol{U}; \theta_u) - l_u(\boldsymbol{U}; \theta^0_u))$, for $g_t$ and $\theta_u$, respectively. The total excess risk is denoted as $\rho^2_t(\gamma_t, \gamma^0_t) = \rho^2_{g_t}(g^0_t, g_t) + \rho^2_u(\theta_u, \theta^0_u)$. We adopt the same notation except for the loss functions used in this subsection.

Theorem 3 and Theorem 4 can be obtained directly using the bounds Lemma 26 and converting the excess risk bound to the Wasserstein distance by Lemma 27.

**Lemma 26 (Estimation error for the mapping** $g_t$**)** *Under Assumption 6, setting the hyperparameters of the neural network* $\Theta_g$*' with a set of sufficiently large positive constants* $\{c_L, c_W, c_S, c_B, c_E\}$ *such that* $\mathbb{L}_g = c_L L \log L$, $\mathbb{W}_g = c_W W \log W$, $\mathbb{S}_g = c_S W^2 L \log^2 W \log L$, $\mathbb{B}_g = c_B$, *and* $\log \mathbb{E}_g = c_E \log(WL)$, *with* $K = WL$, *yields the* $L_2$ *approximation error: for any* $x \geq 1$

$$\rho_{g_t}(g^0_t, \hat{g}_t) \leq x(\beta_g + \delta_g),$$

*with a probability exceeding* $1 - \exp(-c_7 n_t (x(\beta_g + \delta_g))^2)$. *Here, the estimation error* $\beta_g$ *and the approximation error* $\delta_g$ *are bounded by* $\beta_g \asymp \sqrt{\frac{K^2 \log^5 K}{n_t}}$, $\quad \delta_g \asymp K^{\frac{-2r_g}{d_u}} \log^{\frac{r_g}{2}} K$. *To obtain the optimal trade-off, we set* $\beta_g = \delta_g$ *to determine* $K$, *after ignoring the logarithmic term, the optimal bound is obtained by* $K^2 \asymp n_t^{\frac{d_u}{2(d_u+2r_g)}}$. *This yields* $\rho_{g_t}(g^0_t, \hat{g}_t) = O_p(n_t^{-\frac{r_g}{d_u+2r_g}} \log^{m_g} n_t)$, *where* $m_g = \max(\frac{5}{2}, \frac{r_g}{2})$.

**Proof** Using the sub-Gaussian property of $\boldsymbol{U}$ in Assumption 6, we focus our attention on $\boldsymbol{U} \in \mathcal{B} = [-c_u\sqrt{\log(WL)}, c_u\sqrt{\log(WL)})]^{d_u}$ by truncation for some sufficiently large $c_u > 0$. Note that $g^0_t$ is bounded by $B_g$ by Assumption 6 and $\sup_{g_t \in \Theta_g, \boldsymbol{u}} \|g_t(\boldsymbol{u})\|_\infty \leq \mathbb{B}_g$ in the setting of $\Theta_g$. Then, by choosing sufficiently large $c > 0$ so that

$$\int_{\boldsymbol{u} \in \mathbb{R}^{d_u}/\mathcal{B}} \|g_t(\boldsymbol{u}) - g^0_t(\boldsymbol{u})\|^2 p_u(\boldsymbol{u}) \mathrm{d}\boldsymbol{u} = O((WL)^{-\frac{4r_g}{d_u}}). \tag{60}$$

We transform $\boldsymbol{u}$ from $I$ into $[0,1]^{d_u}$ and apply Lemma 42. Specifically, let $\boldsymbol{y} = \frac{\boldsymbol{u} + c_u\sqrt{\log(WL)}}{2c_u\sqrt{\log(WL)}}$ and $\bar{g}^0_t(\boldsymbol{y}) = g_t(\boldsymbol{u})$, , which changes the Hölder-norm $B_g$ to $(2c_u)^{r_g} \log^{\frac{r_g}{2}}(WL) B_g$. Then there exists

an NN $\phi$ with depth $L\log L$ and width $W\log W$ such that

$$\sup_{\boldsymbol{y}\in[0,1]^{d_u}} \|\phi(\boldsymbol{y}) - \bar{g}_t^0(\boldsymbol{y})\|_\infty = O(\log^{\frac{r_g}{2}}(WL)(WL)^{-\frac{2r_g}{d_u}}).$$

Then, we let $\pi g_t^0(\boldsymbol{u}) = \phi(\frac{\boldsymbol{u}+c_u\sqrt{\log(WL)}}{2c_u\sqrt{\log(WL)}})$ and obtain the bound $\sup_{\boldsymbol{u}\in\mathcal{B}}\|\pi g_t^0(\boldsymbol{u}) - g_t^0(\boldsymbol{u})\|_\infty = O(\log^{\frac{r_g}{2}}(WL)(WL)^{-\frac{2r_g}{d_u}})$, which implies

$$\int_{\mathcal{B}} \|\pi g_t^0(\boldsymbol{u}) - g_t^0(\boldsymbol{u})\|^2 p_u(\boldsymbol{u})\mathrm{d}\boldsymbol{u} \le \sup_{\boldsymbol{u}\in\mathcal{B}} \|\pi g_t^0(\boldsymbol{u}) - g_t^0(\boldsymbol{u})\|_\infty^2 = O(\log^{r_g}(WL)(WL)^{-\frac{4r_g}{d_u}}).$$

By (60), $\delta_g = O(\log^{\frac{r_g}{2}}(WL)(WL)^{-\frac{2r_g}{d_u}})$. The estimation error $\beta_g$ is derived as in Lemma 16 with $H(u, \{l_{g_t}(\cdot; g_t), g_t \in \Theta_{g_t}\}) \asymp H(u, \Theta_{g_t}\}) = O(\mathbb{SL}\log(\mathbb{EWL}R/u))$ where $R = c_u\sqrt{\log(WL)}$ by the truncation. By the boundedness of $g_t^0$ and $g_t \in \Theta_{g_t}$, the square loss satisfies Assumptions 9 and 10 as in the proof of Theorem 13 for the score matching loss. The generation error is obtained by Proposition 1 as in the case of the score matching loss. This completes the proof. ■

**Lemma 27 (Error of unconditional diffusion generation)** *Under the conditions in Theorems 3 and 4, the errors for the transfer and non-transfer models under the Wasserstein distance $\mathcal{W}$ are bounded by $\mathcal{W}(P^0_{\boldsymbol{x}_t}, \hat{P}_{\boldsymbol{x}_t}) \le c_8\rho_t(\gamma_t^0, \hat{\gamma}_t)$, $\mathcal{W}(P^0_{\boldsymbol{x}_t}, \tilde{P}_{\boldsymbol{x}_t}) \le c_8\rho_t(\gamma_t^0, \tilde{\gamma}_t)$, for some constant $c_8 > 0$, leading to the error bounds under the Wasserstein distance $\mathcal{W}$ from Theorem 3 and Theorem 4.*

**Proof** [Lemma 27] By the triangle inequality,

$$\mathcal{W}(\hat{P}_{\boldsymbol{x}_t}, P^0_{\boldsymbol{x}_t}) \le \mathcal{W}(\hat{P}_u(\hat{g}_t^{-1}), P_u^0(\hat{g}_t^{-1})) + \mathcal{W}(P_u^0(\hat{g}_t^{-1}), P_u^0((g_t^0)^{-1})).$$

Note that when $g_t$ is bounded, $\hat{P}_{\boldsymbol{x}}$ and $P^0_{\boldsymbol{x}}$ are supported in a bounded domain with diameter $c_R$ which depends on $B_g$ and $\mathbb{B}_g$. Then $\mathcal{W}(\hat{P}_u(\hat{g}_t^{-1}), P_u^0(\hat{g}_t^{-1}))$ can be bounded by the TV distance with $c_{TV}$ given in Lemma 19,

$$\mathcal{W}(\hat{P}_u(\hat{g}_t^{-1}), P_u^0(\hat{g}_t^{-1})) \le c_R\mathrm{TV}(\hat{P}_u(\hat{g}_t^{-1}), P_u^0(\hat{g}_t^{-1})) \le c_R c_{TV}\rho_U(\theta_u^0, \hat{\theta}_u).$$

By the distance definition $\mathcal{W}(\hat{P}_{\boldsymbol{x}_t}, P^0_{\boldsymbol{x}_t}) = \sup_{\|f\|_{Lip}\le 1}\left|\int f(\boldsymbol{x})\mathrm{d}\hat{P}_X - \int f(\boldsymbol{x})\mathrm{d}P_X^0\right|$,

$$\mathcal{W}(P_u^0(\hat{g}_t^{-1}, P_u^0((g_t^0)^{-1}) = \sup_{\|f\|_{Lip}\le 1} |\mathrm{E}_u f\circ\hat{g}_t - \mathrm{E}_u f\circ g_t^0| \le \mathrm{E}_u\|\hat{g}_t - g_t^0\| \le \rho_g(g_t^0, \hat{g}_t).$$

Combining the above inequalities leads to the final result. This completes the proof. ■

Next, we will derive an explicit bound for $\rho(\theta_u^0, \hat{\theta}_u)$, which yields the generation error rate discussed in Section 3.

**Assumption 13 (Target density for $\boldsymbol{U}$)** *Suppose the latent density of $\boldsymbol{U}$, denoted $p^0_{\boldsymbol{u}}$, can be expressed as $p_{\boldsymbol{u}}(\boldsymbol{u}) = \exp(-c_9\|\boldsymbol{u}\|^2/2)\cdot f_u(\boldsymbol{u})$, where $c_9 > 0$ is a constant. Furthermore, $f_u$*

*is contained in a Hölder ball $\mathcal{C}^{r_u}(\mathbb{R}^{d_u}, \mathbb{R}, B_u)$ of radius $B_u$ and is bounded below by a positive constant.*

**Lemma 28 (Latent generation error)** *Under Assumption 13, if we choose the structure hyperparameters of any neural network in $\Theta_u$ to be $\mathbb{L}_u = c_L \log^4 K$, $\mathbb{W}_u = c_W K \log^7 K$, $\mathbb{S}_u = c_S K \log^9 K$, $\log \mathbb{B}_u = c_B \log K$, $\log \mathbb{E}_u = c_E \log^4 K$, with diffusion stopping criteria from (4)-(5) as $\log \underline{\tau}_u = -c_{\underline{\tau}} \log K$ and $\overline{\tau}_u = c_{\overline{\tau}} \log K$, where $\{c_L, c_W, c_S, c_B, c_E, c_{\underline{\tau}}, c_{\overline{\tau}}\}$ are sufficiently large constants, then the excess risk is bounded by: for any $x \geq 1$,*

$$P(\rho(\theta_u^0, \hat{\theta}_u) \geq x\varepsilon_s^u) \leq \exp(-c_3 n_s^{1-\xi}(x\varepsilon_s^u)^2),$$

*with some constant $c_3 > 0$ and a small $\xi > 0$ same in Assumption 4. Here, $\varepsilon_s^u = \beta_u + \delta_u$ with the estimation error $\beta_u$ and the approximation error $\delta_u$ defined as $\beta_u \asymp \sqrt{\frac{K \log^{19} K}{n_s}}$, $\delta_u \asymp K^{-\frac{r_u}{d_u}} \log^{\frac{r_u}{2}+1} K$. To obtain the optimal trade-off, we set $\beta_u = \delta_u$ to determine $K$, after ignoring the logarithmic term, the optimal bound is obtained by $K \asymp n_s^{\frac{d_u}{d_u+2r_u}}$. This yields $\varepsilon_s^u \asymp n_s^{-\frac{r_u}{d_u+r_u}} \log^{m_u} n_s$, where $m_u = \max(\frac{19}{2}, \frac{r_u}{2}+1)$. Similarly, $\varepsilon_t^u \asymp n_t^{-\frac{r_u}{d_u+2r_u}} \log^{m_u} n_t$.*

**Proof** This lemma is a direct consequence of Theorem 13. ■

## A.3 Proofs of Section 4

### A.3.1 GENERATION ACCURACY OF COUPLING NORMALIZING FLOWS

Given a training sample set $(\boldsymbol{x}^i, \boldsymbol{z}^i)_{i=1}^n$, we define the loss function as $l = -\log p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{z}; \theta)$, and then estimate the flows by minimizing the negative log-likelihood as follows:

$$\begin{aligned}\hat{\theta}_t(\boldsymbol{x}, \boldsymbol{z}) &= \arg\min_{\theta \in \Theta} -\sum_{i=1}^n \log p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}^i, \boldsymbol{z}^i; \theta) \\ &= \arg\min_{\theta \in \Theta} -\sum_{i=1}^n \left(\log p_{\boldsymbol{v}}(\theta(\boldsymbol{x}^i, \boldsymbol{z}^i)) + \log \left|\det\left(\nabla_{\boldsymbol{x}} \theta(\boldsymbol{x}^i, \boldsymbol{z}^i)\right)\right|\right), \end{aligned} \tag{61}$$

where $\Theta = \mathrm{CF}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E}, \lambda)$.

Next, we specify some conditions for the true mapping $T^0$.

**Assumption 14** *There exists a map $T^0 : [0,1]^{d_x} \times [0,1]^{d_z} \to [0,1]^{d_x}$ such that $\boldsymbol{v} = T^0(\boldsymbol{x}, \boldsymbol{z})$, where $\boldsymbol{v}$ a random vector with a known lower bounded smooth density, $p_{\boldsymbol{v}} \in \mathcal{C}^\infty([0,1]^{d_x}, \mathbb{R}, B_v)$. For any $\boldsymbol{z} \in [0,1]^{d_z}$, $T^0(\cdot, \boldsymbol{z})$ is invertible and $|\det(\nabla_{\boldsymbol{x}} T^0)|$ is lower bounded. Moreover, $T^0(\boldsymbol{v}, \boldsymbol{z})$ and its inverse given $\boldsymbol{z}$ belong to Hölder ball $\mathcal{C}^{r+1}([0,1]^{d_x} \times [0,1]^{d_z}, [0,1]^{d_x}, B)$.*

**Theorem 29 (Generation error of coupling flows)** *Under the conditions in Assumption 14, we set the neural network's structure hyperparameters within $\Theta$ with a set of sufficiently large positive constants $\{c_L, c_W, c_S, c_B, c_E, c_\lambda\}$ as follows: $\mathbb{L} = c_L L \log L$, $\mathbb{W} = c_W W \log W$, $\mathbb{S} = c_S W^2 L \log^2 W \log L$, $\mathbb{B} = c_B$, $\log \mathbb{E} = c_E \log(WL)$ and $\lambda = c_\lambda$. With these conditions, let*

*$K = WL$, and then the error in coupling flow models via transfer learning under the KL-divergence $\mathcal{K}$ is bounded: for any $x \geq 1$ and some constant $c_e > 0$,*

$$P(\mathrm{E}_{\boldsymbol{z}}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}|\boldsymbol{z}}, \hat{p}_{\boldsymbol{x}|\boldsymbol{z}})] \geq x(\beta_n + \delta_n)) \leq \exp(-c_e n(x(\beta_n + \delta_n))^2). \tag{62}$$

*Here, $\beta_n$ and $\delta_n$ represent the estimation and approximation errors: $\beta_n \asymp \sqrt{\frac{K^2 \log^5 K}{n}}$ and $\delta_n \asymp K^{\frac{-2r}{d_x+d_z}}$. In (62), setting $\beta_n = \delta_n$ to solve for $K$, and neglecting the logarithmic term, leads to $\beta_n = \delta_n \asymp n^{-\frac{r}{d_x+d_z+2r}}$ with the optimal $K \asymp \sqrt{n^{\frac{d_x+d_z}{d_x+d_z+2r}}}$. Consequently, this provides the best bound for (62) $n^{-\frac{r}{d_x+d_z+2r}} \log^{5/2} n$.*

*Moreover, when $\boldsymbol{Z} = \emptyset$ and $d_z = 0$, this error bound can be extended to the degenerate case, unconditional flow models, $\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}}, \hat{p}_{\boldsymbol{x}}) = O_p(n^{-\frac{r}{d_x+2r}} \log^{5/2} n)$.*

**Proof** [Theorem 29] Let $\theta^0 = T^0$. By Assumption 14, $\theta^0$ has a lower bounded determinant of the Jacobian matrix, that is, $|\det(\nabla_{\boldsymbol{x}}\theta^0(\boldsymbol{x}, \boldsymbol{z}))| > \underline{c}_{\theta^0} > 0$, and $p_v$ is lower bounded by some constant, $p_v \geq \underline{c}_v > 0$. Moreover, $|\det(\nabla_{\boldsymbol{x}}\theta(\boldsymbol{x}, \boldsymbol{z}))| \leq \Pi_{i=1}^{d_x} \|\nabla_{\boldsymbol{x}_i}\theta(\boldsymbol{x}, \boldsymbol{z})\| \leq (\sqrt{d_x}B)^{d_x}$ by Hadamard's inequality Różański et al. (2017).

Note that $\Theta$ is an invertible class. Then, $\inf_{\theta\in\Theta,\boldsymbol{x},\boldsymbol{z}} |\det(\nabla_{\boldsymbol{x}}\theta(\boldsymbol{x}, \boldsymbol{z}))| > \underline{c}_\theta$ for some constant $\underline{c}_\theta > 0$. Hence, the densities are lower bounded,

$$p^0_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{z}) \geq \underline{c}_v \underline{c}_{\theta^0}, \text{ and } p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x}, \boldsymbol{z}) \geq \underline{c}_v \underline{c}_\theta, \tag{63}$$

implying that

$$\frac{p_{\boldsymbol{x}|\boldsymbol{z}}}{p^0_{\boldsymbol{x}|\boldsymbol{z}}} = \frac{p_{\boldsymbol{v}}(\theta(\boldsymbol{x}, \boldsymbol{z}))|\det(\nabla_{\boldsymbol{x}}\theta(\boldsymbol{x}, \boldsymbol{z}))|}{p_{\boldsymbol{v}}(\theta^0(\boldsymbol{x}, \boldsymbol{z}))|\det(\nabla_{\boldsymbol{x}}\theta^0(\boldsymbol{x}, \boldsymbol{z}))|} \geq \underline{c}_r,$$

for some constant $\underline{c}_r = \frac{\underline{c}_v \underline{c}_\theta}{B_v(\sqrt{d_x}B)^{d_x}}$. Meanwhile, $\sup_{\theta\in\Theta,\boldsymbol{x},\boldsymbol{z}} |\det(\nabla_{\boldsymbol{x}}\theta(\boldsymbol{x}, \boldsymbol{z}))| \leq (\sqrt{d_x}\lambda)^{d_x}$ and $\frac{p_{\boldsymbol{x}|\boldsymbol{z}}}{p^0_{\boldsymbol{x}|\boldsymbol{z}}} \leq \frac{B_v(\sqrt{d_x}\lambda)^{d_x}}{\underline{c}_v, \underline{c}_{\theta^0}} := \overline{c}_r$. Then, $l(\boldsymbol{x}, \boldsymbol{z}; \theta) - l(\boldsymbol{x}, \boldsymbol{z}; \theta^0)$ is bounded, which satisfies Assumption 10 with $c_b = \max(|\log \overline{c}_r|, |\log \underline{c}_r|)$.

To verify the variance condition in Assumption 9, note that the likelihood ratio is bounded above and below. By Lemmas 4 and 5 of Wong and Shen (1995), the first and second moments of the difference of the log-likelihood functions is bounded:

$$\mathrm{E}_{\boldsymbol{x}|\boldsymbol{z}}(l(\boldsymbol{X}, \boldsymbol{Z}; \theta^0) - l(\boldsymbol{X}, \boldsymbol{Z}; \theta))^j \leq c_l \|p^{1/2}_{\boldsymbol{x}|\boldsymbol{z}} - (p^0_{\boldsymbol{x}|\boldsymbol{z}})^{1/2}\|^2_{L_2} \tag{64}$$

for $j = 1, 2$ and some constant $c_l > 0$ depend on $\underline{c}_r$. This implies that

$$\begin{aligned}\mathrm{Var}_{\boldsymbol{x},\boldsymbol{z}}(l(\boldsymbol{X}, \boldsymbol{Z}; \theta^0) - l(\boldsymbol{X}, \boldsymbol{Z}; \theta)) &\leq \mathrm{E}_{\boldsymbol{x},\boldsymbol{z}}[(l(\boldsymbol{X}, \boldsymbol{Z}; \theta^0) - l(\boldsymbol{X}, \boldsymbol{Z}; \theta))^2] \\ &\leq c_l \mathrm{E}_{\boldsymbol{z}} \|p^{1/2}_{\boldsymbol{x}|\boldsymbol{z}} - (p^0_{\boldsymbol{x}|\boldsymbol{z}})^{1/2}\|^2_{L_2} \leq c_l \rho^2(\theta^0, \theta).\end{aligned}$$

Hence, Assumption 9 holds with $c_v = c_l$.

Thus, we can apply Proposition 1 together with Lemmas 30 and 32 to give the desired result in the same manner as the proof of Theorem 1. By Lemma 32, there exists a positive $c_H$ such that

$H_B(u, \mathcal{F}) \leq c_H K^2 \log^4 K \log \frac{K}{u}$ . Then the integral entropy inequality can be solved by

$$\int_{k\varepsilon^2/16}^{4c_v^{1/2}\varepsilon} H_B^{1/2}(u, \mathcal{F}) \mathrm{d}u \leq c_H \int_{k\varepsilon^2/16}^{4c_v^{1/2}\varepsilon} K \log^4 K \log \frac{K}{u} \mathrm{d}u \leq 4 c_H c_v^{1/2} \varepsilon \sqrt{K^2 \log^4 K \log \frac{K}{\varepsilon^2}}.$$

Solving $\beta_n$: $4c_H c_v^{1/2} \beta_n \sqrt{K^2 \log^4 K \log \frac{K}{\varepsilon^2}} \leq c_h \sqrt{n} \beta_n^2$ yields $\beta_n = 2\frac{4c_H\sqrt{c_v}}{c_h} \sqrt{\frac{K^2 \log^5 K}{n}}$.

Let $\varepsilon_n \geq \beta_n + \delta_n$ with $\beta_n \asymp \sqrt{\frac{K^2 \log^3 K}{n}}$ so that $\varepsilon_n$ satisfies the conditions of Proposition 1. Note that $c_e \asymp \frac{1}{c_v^2} \asymp 1$ in Proposition 1. With $\log K = O(\log n)$, for some constant $c_e > 0$, $P(\rho(\theta^0, \theta) \geq \varepsilon_n) \leq 4 \exp(-c_e n \varepsilon_n^2)$. ■

The next lemma gives the approximation error for the coupling flows in the KL divergence. The coupling network has been shown to possess the universal approximation property (Ishikawa et al., 2023). But the approximation rate result is largely missing, except Jin et al. (2024) provided an approximation error bound for a bi-Lipschitz $T^0$, which cannot be used for a smooth $T^0$ to present the density-based metric result. Our approximation employs the zero-padding method and the ReQU network to give the explicit rate. The zero-padding technique is used in Lyu et al. (2022) to show that coupling flows are also universal approximators for the derivatives, and the ReQU network is used in Belomestny et al. (2021) to give the approximation rate for the derivatives of smooth functions.

**Lemma 30 (Approximation error)** *Under Assumption 14, there exists a coupling network* $\pi\theta^0 \in \mathrm{CF}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})$ *with* $\mathbb{L} = c_L L \log L$, $\mathbb{W} = c_W W \log W$, $\mathbb{S} = c_S W^2 L \log^2 W \log L$, $\mathbb{B} = c_B$, $\log \mathbb{E} = c_E \log(WL)$, *and* $\lambda = c_\lambda$, *such that*

$$\rho(\theta^0, \pi\theta^0) = O((WL)^{\frac{-2r}{d_x + d_z}}).$$

**Proof** [Lemma 30] Consider the zero padding method to derive the approximation result for coupling flows. Deep affine coupling networks are shown to be universal approximators in the Wasserstein distance if we allow training data to be padded with sufficiently many zeros (Koehler et al., 2021; Huang et al., 2020). We first present how the zero padding method works in Wasserstein distance. The proof process is to construct a coupling flow $(\boldsymbol{X}, \mathbf{0}) \mapsto (T(\boldsymbol{X}), \mathbf{0})$ as follows:

$$\boldsymbol{X} \xrightarrow{[\boldsymbol{I};\mathbf{0}]} \underbrace{(\boldsymbol{X}, \mathbf{0})}_{\boldsymbol{Y}^1} \xrightarrow{\phi_1} \underbrace{(\boldsymbol{X}, T(\boldsymbol{X}))}_{\boldsymbol{Y}^2} \xrightarrow{\phi_2} \underbrace{(T(\boldsymbol{X}), T(\boldsymbol{X}))}_{\boldsymbol{Y}^3} \xrightarrow{\phi_3} (T(\boldsymbol{X}), \mathbf{0}) \xrightarrow{[\boldsymbol{I},\mathbf{0}]} T(\boldsymbol{X}),$$

where $\mathbf{0}$ is the same dimension of $\boldsymbol{X}$. To achieve this, three coupling layers are used,

$$[\phi_1(\boldsymbol{Y}^1)]_j = \begin{cases} \boldsymbol{Y}_j^1 & j = 1, \ldots, d \\ \boldsymbol{Y}_j^1 + [T(\boldsymbol{Y}_{1:d}^1)]_j & j = d+1, \ldots, 2d \end{cases}$$
$$[\phi_2(\boldsymbol{Y}^2)]_j = \begin{cases} \boldsymbol{Y}_j^2 - ([T^{-1}(\boldsymbol{Y}_{(d+1):2d}^2)]_j + \boldsymbol{Y}_{j+d}^2) & j = 1, \ldots, d \\ \boldsymbol{Y}_j^2 & j = d+1, \ldots, 2d \end{cases}$$
$$[\phi_3(\boldsymbol{Y}^3)]_j = \begin{cases} \boldsymbol{Y}_j^3 & j = 1, \ldots, d \\ \boldsymbol{Y}_j^3 - \boldsymbol{Y}_{j-d}^3 & j = d+1, \ldots, 2d. \end{cases}$$

Then, if we have common networks that approximate $T$ and $T^{-1}$ well in $\phi_1$ and $\phi_2$, then we can control the approximation error for the coupling network. However, it is not sufficient to control the KL divergence by the approximation error in $T$, because this mapping is volume-preserving with the Jacobian determinant equal to one. The existing literature on affine coupling-based normalizing flows considers weak convergence Teshima et al. (2020) in the Wasserstein distance. The approximability to derivatives remains hardly untouched, except Lyu et al. (2022) which takes into account the approximation of derivatives but fails to give an explicit approximation error rate.

Next, we will adjust the zero padding method to approximate $T$ and $|J_T|$ simultaneously. The proof process is to construct a coupling flow $(\boldsymbol{X}) \mapsto (T(\boldsymbol{X}, \boldsymbol{Z}))$ as follows:

$$\boldsymbol{X} \xrightarrow{[\boldsymbol{I};\mathbf{0}]} \underbrace{(\boldsymbol{X}, \mathbf{0})}_{\boldsymbol{Y}^1} \xrightarrow{\phi_1} \underbrace{(\boldsymbol{X}, T(\boldsymbol{X}, \boldsymbol{Z}))}_{\boldsymbol{Y}^2} \xrightarrow{\phi_2} \underbrace{(T(\boldsymbol{X}, \boldsymbol{Z}), \boldsymbol{X})}_{\boldsymbol{Y}^3} \xrightarrow{\phi_3} (T(\boldsymbol{X}, \boldsymbol{Z}), \mathbf{0}) \xrightarrow{[\boldsymbol{I},\mathbf{0}]} T(\boldsymbol{X}, \boldsymbol{Z}).$$

where $\mathbf{0}$ is in the same dimension as $\boldsymbol{X}$. To achieve this, two coupling layers and a permutation layer are used,

$$[\phi_1(\boldsymbol{Y}_1)]_j = \begin{cases} \boldsymbol{Y}_{1,j} & \text{for } j = 1, \ldots, d_x, \\ \boldsymbol{Y}_{1,j} + [T(\boldsymbol{Y}_1^{1:d_x})]_j & \text{for } j = d_x+1, \ldots, 2d_x; \end{cases}$$
$$[\phi_2(\boldsymbol{Y}_2)]_j = \begin{cases} \boldsymbol{Y}_{2,j+d_x} & \text{for } j = 1, \ldots, d_x, \\ \boldsymbol{Y}_{2,j-d_x} & \text{for } j = d_x+1, \ldots, 2d_x; \end{cases}$$
$$[\phi_3(Y_3)]_j = \begin{cases} \boldsymbol{Y}_{3,j} & \text{for } j = 1, \ldots, d_x, \\ \boldsymbol{Y}_{3,j} - [T^{-1}(\boldsymbol{Y}_3^{1:d_x})]_j & \text{for } j = d_x+1, \ldots, 2d_x. \end{cases} \tag{65}$$

In this process, the Jacobian of the composite transformations remains $\nabla_{\boldsymbol{x}} T$,

$$[\boldsymbol{I}, \mathbf{0}] \nabla_{\boldsymbol{x}} \phi_3 \nabla_{\boldsymbol{x}} \phi_2 \nabla_{\boldsymbol{x}} \phi_1 \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} = [\boldsymbol{I}, \mathbf{0}] \begin{bmatrix} \boldsymbol{I} & 0 \\ -\nabla_{\boldsymbol{x}} T^{-1} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \mathbf{0} & \boldsymbol{I} \\ \boldsymbol{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} & \mathbf{0} \\ \nabla_{\boldsymbol{x}} T & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{I} \\ \mathbf{0} \end{bmatrix} = \nabla_{\boldsymbol{x}} T.$$

Here, the subscript $j$ refers to the $j$-th component of a vector, and $\boldsymbol{Y}_i^{1:d_x}$ refers to the first $d_x$ components of $Y_i$. The coupling layer; $j = 1, 3$, is defined by $\phi_j(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\boldsymbol{x}_1, \boldsymbol{x}_2 + \omega_j(\boldsymbol{x}_1))$ and $\omega_j(\boldsymbol{x}_2)$. To ensure invertibility, $\omega_1$ is required to be invertible and $\omega_3 = -\omega_1^{-1}$, where $\omega_1$ approximates $T$ using a mixed ReLU and ReQU neural network. Here, ReQU stands for rectified

quadratic unit, with $\sigma(x) = \max^2(0, x)$ as an activation function, which permits an approximate of a smooth function and its derivative simultaneously (Belomestny et al., 2023). Meanwhile, the second layer employs a permutation, with $\phi_2$ as a function to permute the first block with the second.

Using Lemma 31, we construct an invertible $\Phi$ to approximate $T^0$ while obtaining an analytical solution $\Phi^{-1}$ to inverse $\Phi$ under some constraints on $\Phi$. By Lemma 31, the network $\Phi$ has depth $O(L \log L)$ and width $O(W \log W)$ such that the approximation error is bounded by

$$\|\Phi - T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}) \text{ and } \|\nabla_x \Phi - \nabla_{\boldsymbol{x}} T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}).$$

Here, for a $d$ dimension output function $f$, $\|f\|_{\infty,2} = \|\|f\|_\infty\|_{L_2} = (\int_{\boldsymbol{x}} (\max_j |f_j(\boldsymbol{x})|)^2 \mathrm{d}\boldsymbol{x})^{1/2}$, where $f_j$ is the $j$-th element of $f$ and $\|g\|_{L_2} = \sqrt{\int g^2(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}}$ is the $L_2$ norm for the univariate output function. For the coupling flow $\pi\theta^0$, the approximation error is bounded by

$$\|\pi\theta^0 - \theta^0\|_{\infty,2} = \|\Phi - T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}).$$

Meanwhile, by the perturbation bound of the determinant in Lemma 33, we bound the determinant error with some positive constant $c_d$,

$$\|| \det(\nabla_{\boldsymbol{x}} \pi\theta^0)| - | \det(\nabla_{\boldsymbol{x}} \theta^0)|\|_{L_2} \le c_d \|\nabla_{\boldsymbol{x}} \Phi - \nabla_{\boldsymbol{x}} T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}). \tag{66}$$

On the other hand, let $p_{\boldsymbol{x}|\boldsymbol{z}}$ be the density given by $\pi\theta^0$. Note that the likelihood ratio is lower and upper bounded. Then, the KL divergence is upper bounded by the squared $L_2$ distance from (63) and (64),

$$\rho^2(\theta^0, \pi\theta^0) = \mathcal{K}(p^0_{\boldsymbol{x}|\boldsymbol{z}}, p_{\boldsymbol{x}|\boldsymbol{z}}) \le c_l \int (p^{1/2}_{\boldsymbol{x}|\boldsymbol{z}} - (p^0_{\boldsymbol{x}|\boldsymbol{z}})^{1/2})^2 \le 2c_l(\underline{c}_v \underline{c}_\theta + \underline{c}_v \underline{c}_{\theta^0}) \|p_{\boldsymbol{x}|\boldsymbol{z}} - p^0_{\boldsymbol{x}|\boldsymbol{z}}\|^2_{L_2}.$$

Moreover, by the transformation, we have the decomposition,

$$\begin{aligned}
&\|p_{\boldsymbol{x}|\boldsymbol{z}} - p^0_{\boldsymbol{x}|\boldsymbol{z}}\|_{L_2} = \|p_{\boldsymbol{v}}(\theta^0)| \det(\nabla_{\boldsymbol{x}} \theta^0)| - p_{\boldsymbol{v}}(\pi\theta^0)| \det(\nabla_{\boldsymbol{x}} \pi\theta^0)|\|_{L_2} \\
\le &\|p_{\boldsymbol{v}}(\theta^0)| \det(\nabla_{\boldsymbol{x}} \theta^0)| - p_{\boldsymbol{v}}(\pi\theta^0)| \det(\nabla_{\boldsymbol{x}} \theta^0)|\|_{L_2} \\
&+ \|p_{\boldsymbol{v}}(\pi\theta^0)| \det(\nabla_{\boldsymbol{x}} \theta^0)| - p_{\boldsymbol{v}}(\pi\theta^0)| \det(\nabla_{\boldsymbol{x}} \pi\theta^0)|\|_{L_2} \\
\le &(\sqrt{d_x} B)^{d_x} \sqrt{d_x} B_v \|\theta^0 - \pi\theta^0\|_{\infty,2} + B_v \|| \det(\nabla_{\boldsymbol{x}} \theta^0)| - | \det(\nabla_{\boldsymbol{x}} \pi\theta^0)|\|_{L_2}.
\end{aligned} \tag{67}$$

Hence,

$$K^{1/2}(p^0_{\boldsymbol{x}|\boldsymbol{z}}, p_{\boldsymbol{x}|\boldsymbol{z}}) \asymp \|T^0 - \Phi\|_{\infty,2} + \|| \det(\nabla_{\boldsymbol{x}} T^0)|) - | \det(\nabla_{\boldsymbol{x}} \Phi)|\|_{L_2} = O((WL)^{-\frac{2r}{d_x+d_z}}).$$

This completes the proof. ■

The next lemma will show that there exists an invertible neural network $\Phi$ to approximate $T^0$ and $\nabla_{\boldsymbol{x}} T^0$ simultaneously with a combination use of the ReLU network and ReQU network.

**Lemma 31** *There exists an invertible neural network* $\Phi$ *with* $\mathbb{W} = c_W W \log W$, $\mathbb{L} = c_L L \log L$, $\mathbb{S} = c_S \mathbb{W}^2 \mathbb{L}$, $\mathbb{B} = c_B$, $\log \mathbb{E} = c_E \log(WL)$ *and* $\lambda = c_\lambda$ *such that*

$$\|\Phi - T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}), \quad \|\nabla_x \Phi - \nabla_x T^0\|_{\infty,2} = O((WL)^{-\frac{2r}{d_x+d_z}}).$$

**Proof** Before proceeding, we describe the idea of building a neural network $\Phi$ in two steps: (1) approximating a smooth map $T^0$ with a local polynomial $\bar{T}$, as in Lu et al. (2021); (2) constructing an invertible neural network $\Phi$ to approximate $\bar{T}$, where $\Phi$ is subject to some constraints.

Let $\boldsymbol{s} = (\boldsymbol{x}, \boldsymbol{z})$ and $K = [(WL)^{2/d}]$. We uniformly partition a box area into non-overlapping hypercubes $\{\mathcal{B}^i\}_{i=1}^{K^d}$ with edge sizes $\frac{1}{K}$. Note that $T^0 \in \mathcal{C}^{1+r}$. For any $\boldsymbol{s} \in \mathcal{B}^i = \{\boldsymbol{s} : \boldsymbol{s}_j^i \le \boldsymbol{s}_j \le \boldsymbol{s}_j^i + \frac{1}{K}\}$ with $\boldsymbol{s}_j^i = \frac{\boldsymbol{K}_j}{K}$ and $\boldsymbol{k} \in [K-1]^d$, there exists $\xi_{\boldsymbol{s}} \in (0,1)$ such that

$$\begin{aligned} T^0(\boldsymbol{s}) &= \sum_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor} \frac{\partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i)}{\boldsymbol{\alpha}! \partial \boldsymbol{s}^{\boldsymbol{\alpha}}} (\boldsymbol{s} - \boldsymbol{s}^i)^{\boldsymbol{\alpha}} + \sum_{|\boldsymbol{\alpha}| = \lfloor r \rfloor + 1} \frac{\partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i + \xi_{\boldsymbol{s}}(\boldsymbol{s} - \boldsymbol{s}^i))}{\boldsymbol{\alpha}!} (\boldsymbol{s} - \boldsymbol{s}^i)^{\boldsymbol{\alpha}} \\ &= \sum_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor + 1} \frac{\partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i)}{\boldsymbol{\alpha}!} (\boldsymbol{s} - \boldsymbol{s}^i)^{\boldsymbol{\alpha}} + O(\|\boldsymbol{s} - \boldsymbol{s}^i\|^{r+1}) \equiv \bar{T}(\boldsymbol{s}) + O(\|\boldsymbol{s} - \boldsymbol{s}^i\|^{r+1}), \end{aligned}$$

where $\alpha$ is a multi-index with $|\cdot|$ indicating its size, and $\sup_{\boldsymbol{s}} \|\bar{T}(\boldsymbol{s}) - T^0(\boldsymbol{s})\|_\infty = O(K^{-(r+1)})$ and $\sup_{\boldsymbol{s}} \|\nabla_{\boldsymbol{x}} \bar{T}(\boldsymbol{s}) - \nabla_{\boldsymbol{x}} T^0(\boldsymbol{s})\|_\infty = O(K^{-r})$. Next, we construct an invertible network to approximate $\bar{T}$ with an error bounded by $O(K^{-r})$ in the $L_2$ distance.

To achieve this, we first adjust $\mathcal{B}^i$ to a new cube $\{\mathcal{B}^i(\epsilon)\}$with radius $\epsilon > 0$, where $[\mathcal{B}^i(\epsilon)]_j = [\boldsymbol{s}_j^i, \boldsymbol{s}_j^i + \frac{1}{K} - \epsilon]$ for a small $\epsilon > 0$. We approximate $\bar{T}$ in $\{\mathcal{B}^i(\epsilon)\}$ with its error controlled by choosing a small $\epsilon$.

The neural network $\Phi$ is constructed in three steps.

1. The first step constructs a ReLU network $\Phi_a$ to yield step functions over $\mathcal{B}^i(\epsilon)$ such that $\phi_a(\boldsymbol{s}) = \boldsymbol{s}^i$ if $\boldsymbol{s}^i \in \mathcal{B}^i(\epsilon)$ for $\phi_a \in \Phi_a$. This reduces the function approximation problem to a point-fitting problem at fixed grid points.

2. The second step constructs a group of ReLU networks $\Phi_b = \{\phi_b^{\boldsymbol{\alpha}}\}$ such that $\phi_b^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)$ is close to $\partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i)$ for $|\boldsymbol{\alpha}| \le \lfloor r \rfloor + 1$.

3. The last step constructs a ReQU network $\Phi_c$ such that $\phi_c(\boldsymbol{s}) \in \Phi_c$ to yield a polynomial in that $\phi_c(\boldsymbol{s} - \boldsymbol{s}^i, \phi_b^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)) = \sum_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor + 1} \frac{\phi_b^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)}{\boldsymbol{\alpha}!} (\boldsymbol{s} - \boldsymbol{s}^i)^{\boldsymbol{\alpha}}$ .

Combining $\Phi_a$–$\Phi_c$, we define an element $\phi(\boldsymbol{s})$ in the complete network $\Phi$ as

$$\phi(\boldsymbol{s}) = \phi_c(\boldsymbol{s} - \phi_a(\boldsymbol{s}), \phi_b^{\boldsymbol{\alpha}}(\phi_a(\boldsymbol{s}))) = \sum_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor + 1} \frac{\phi_b^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)}{\boldsymbol{\alpha}!} (\boldsymbol{s} - \boldsymbol{s}^i)^{\boldsymbol{\alpha}}. \tag{68}$$

Let $\boldsymbol{\alpha}(\beta, j) = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_j + 1, \ldots, \boldsymbol{\beta}_{d_s}]$ for $j \in [d_s]$ and $\boldsymbol{\beta} \in [r]^{d_s}$. By the chain rule of derivatives and the fact $\nabla \phi_a = \mathbf{0}$, we have, for $s \in \mathcal{B}^k(\epsilon)$,

$$[\nabla \phi(\boldsymbol{s})]_{ij} = \sum_{|\boldsymbol{\beta}| \le \lfloor r \rfloor} \frac{[\phi_b^{\boldsymbol{\alpha}(\boldsymbol{\beta}, j)}(\boldsymbol{s}^k)]_i}{\boldsymbol{\beta}!} (\boldsymbol{s} - \boldsymbol{s}^k)^{\boldsymbol{\beta}}. \tag{69}$$

Then, $\nabla\phi = [\nabla_{\boldsymbol{x}}\phi; \nabla_{\boldsymbol{z}}\phi]$ and $\nabla_{\boldsymbol{x}}\phi$ is the first $d_x$ rows of $\nabla\phi$, a square matrix of dimensions $d_x$.

Specifically,

**1. Approximating the step function.** For $\Phi_a$, we use Proposition 4.3 in Lu et al. (2021), presented in Lemma 40 to construct a ReLU NN with depth $O(L)$ and width $O(W)$ for each dimension to yield the step function. Then $\phi_a$ is constructed with a ReLU network with depth $O(L)$ and width $O(W)$ such that $\phi_a(\boldsymbol{s}) = \boldsymbol{s}^i$, if $\boldsymbol{s} \in \mathcal{B}^i(\epsilon)$.

**2. Point fitting.** As to $\Phi_b$, we use Lemma 41, the point-fitting result from Lu et al. (2021) to construct $\phi_b^{\boldsymbol{\alpha}} = \psi^2 \circ \psi^1$. The construction involves two steps. First, we construct $\psi^1$ bijectively mapping $\{0, 1, \ldots, K-1\}^d$ to $\{1, 2, \ldots, K^d\}$, where $\psi^1(\boldsymbol{k}/K) = \sum_{j=1}^d \boldsymbol{k}_j K^{j-1}$. Then we construct $[\psi^2(i)]_j$ to approximate $[\partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i)]_j$ with the pointwise error of $O((WL)^{-2\alpha})$, where $\psi^2(i)$ is a ReLU NN with depth $O(\alpha L \log L)$ and width $O(d_s W \log W)$. We choose $\alpha$ large enough so that $\max_i \|\psi_b^{\boldsymbol{\alpha}}(s^i) - \partial^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)\|_\infty = \max_i \|\psi^2(i) - \partial^{\boldsymbol{\alpha}}(\boldsymbol{s}^i)\|_\infty = o(\frac{1}{K^{r+1}})$.

**3. Approximating a polynomial.** For $\Phi_c$, we use the ReQU network to implement the product exactly according to Lemma 43. Let $\phi_c(\boldsymbol{s} - \boldsymbol{s}^i, \phi_b^{\boldsymbol{\alpha}}(s^i)) = \sum_{\boldsymbol{\alpha}} \phi_c^{\boldsymbol{\alpha}}(\boldsymbol{s} - \boldsymbol{s}^i, \phi_b^{\boldsymbol{\alpha}}(s^i))$. Then, each $\phi_c^{\boldsymbol{\alpha}}(\boldsymbol{s} - \boldsymbol{s}^i, \phi_b^{\boldsymbol{\alpha}}(s^i))$ is designed as a depth $[\log_2(|\boldsymbol{\alpha}| + 1)]$ and width $2^{[\log_2(|\boldsymbol{\alpha}|+1)]+1}$ ReQU network such that $\phi_c^{\boldsymbol{\alpha}}(\boldsymbol{s} - \boldsymbol{s}^i, \phi_b^{\boldsymbol{\alpha}}(s^i)) = \phi_b^{\boldsymbol{\alpha}}(s^i)(\boldsymbol{s} - \boldsymbol{s}^i)^\alpha$. Then, $\Phi_c$ is constructed as $\mathrm{NN}([\log_2(\|\alpha\|_1 + 1)], r^{d_s} 2^{[\log_2(\|\alpha\|_1+1)]+1})$.

Combining the networks in the three steps, $\Phi$ is a network with depth $\mathbb{L} \asymp L \log L$ and width $\mathbb{W} \asymp W \log W$. The effective neuron number is not greater than $\mathbb{W}^2\mathbb{L}$. Moreover, for any $\boldsymbol{s} \in \cup\mathcal{B}^i(\epsilon)$,

$$\|\phi(\boldsymbol{s}) - \bar{T}(\boldsymbol{s})\|_\infty \le \sum_{|\boldsymbol{\alpha}| \le \lfloor r \rfloor + 1} \|\phi_b^{\boldsymbol{\alpha}}(\boldsymbol{s}^i) - \partial^{\boldsymbol{\alpha}} T^0(\boldsymbol{s}^i)\| = o(\frac{1}{K^{r+1}}).$$

Similarly,

$$\|\nabla_{\boldsymbol{x}}\phi(\boldsymbol{s}) - \nabla_{\boldsymbol{x}}\bar{T}(\boldsymbol{s})\|_\infty \le \max_{j \in [d_s]} \sum_{|\boldsymbol{\beta}| \le \lfloor r \rfloor + 1} \|\phi_b^{\boldsymbol{\alpha}(\beta,j)}(\boldsymbol{s}^i) - \partial^{\boldsymbol{\alpha}(\beta,j)} T^0(\boldsymbol{s}^i)\|_\infty = o(\frac{1}{K^{r+1}}).$$

Combining the approximation error of $\bar{T}$, we can show, for any $\boldsymbol{s} \in \cup\mathcal{B}^i(\epsilon)$,

$$\|\phi(\boldsymbol{s}) - T^0(\boldsymbol{s})\|_\infty = O(\frac{1}{K^{r+1}}) \text{ and } \|\nabla\phi(\boldsymbol{s}) - \nabla T^0(\boldsymbol{s})\|_\infty = O(\frac{1}{K^r}).$$

**Invertibility constraints.** We next outline the constraints necessary to guarantee the invertibility of $\phi$. It is important to note that $\phi$ is defined as a piecewise polynomial function. To ensure its invertibility, two specific conditions must be met. We first require that $\phi$ is invertible in each cube and the image areas $\{\mathcal{Q}^i = \{\phi(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{B}^i(\epsilon)\}\}_{i=1}^{K^d}$ are disjoint. Specifically,

$$\begin{cases} \inf_{\boldsymbol{s} \in \mathcal{B}^i(\epsilon), \boldsymbol{s}' \in \mathcal{B}^j(\epsilon)} \|\phi(s) - \phi(s')\| > c_{ij}^{(1)} & i < j \in [K^d], \\ \inf_{\boldsymbol{s} \in \mathcal{B}^i(\epsilon)} |\det(\nabla_{\boldsymbol{x}}\phi(s))| > c_i^{(2)} & i \in [K^d]. \end{cases} \tag{70}$$

Here, $c_{ij}^{(1)} = O(\epsilon)$ is set to no more than $\frac{1}{\sqrt{d}B}\epsilon$ due to the fact that $\|T^0(\boldsymbol{s}) - T^0(\boldsymbol{s}')\| \ge \frac{1}{\sqrt{d}B}\|\boldsymbol{s} - \boldsymbol{s}'\| \ge \frac{1}{\sqrt{d}B}\epsilon$, and $c_i^{(2)} = O(1)$ is set to no more than $\underline{c}_{\theta^0} = \inf_{\boldsymbol{s}} |\det(\nabla_{\boldsymbol{x}} T^0(\boldsymbol{s}))|$.

When $r \leq 1$, the constraints can be simplified. Note that when $r \leq 1$, for $\boldsymbol{s} \in \mathcal{B}^i(\epsilon)$, $\phi(s) = \phi_b^{\mathbf{0}}(\boldsymbol{s}^i) + \phi_b(\boldsymbol{s}^i)(\boldsymbol{s} - \boldsymbol{s}^i)$ is a piece-wise linear map, where $\phi_b(\boldsymbol{s}^i) = [\phi_b^{\boldsymbol{\alpha}}(s^i)]_{|\boldsymbol{\alpha}|=1}$ is the Jacobian matrix with multi-index $\boldsymbol{\alpha}$ satisfying $|\boldsymbol{\alpha}| = 1$. Then $|\det(\nabla_{\boldsymbol{x}}\phi(s))| = |\det([\phi_b(s^i)]_{\boldsymbol{x}})|$, where $[\phi_b(s^i)]_{\boldsymbol{x}}$ represents the rows associated with $\boldsymbol{x}$. The constraints are simplified as

$$\begin{cases} \inf\limits_{\substack{\boldsymbol{s}\in\mathcal{B}^i(\epsilon)\\ \boldsymbol{s}'\in\mathcal{B}^j(\epsilon)}} \|\phi_b^{\mathbf{0}}(\boldsymbol{s}^i) + \phi_b(\boldsymbol{s}^i)(\boldsymbol{s} - \boldsymbol{s}^i) - \phi_b^{\mathbf{0}}(\boldsymbol{s}^j) - \phi_b(\boldsymbol{s}^j)(\boldsymbol{s}' - \boldsymbol{s}^j)\| > c_{ij}^{(1)}, i < j \in [K^d], \\ |\det([\phi_b(s^i)]_{\boldsymbol{x}})| > c_i^{(2)}, i \in [K^d]. \end{cases} \tag{71}$$

When these two conditions hold, given $\boldsymbol{z}$ and $\Phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{y}$, the inverse is constructed analytically when $r \leq 1$. Note that $\Phi(\boldsymbol{x}, \boldsymbol{z}) = [\phi_b(\boldsymbol{s}^i)]_{\boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{x}^i) + [\phi_b(\boldsymbol{s}^i)]_{\boldsymbol{z}}(\boldsymbol{z} - \boldsymbol{z}^i) + \phi_b^{\mathbf{0}}(\boldsymbol{s}^i)$. Given $\boldsymbol{z}$, if $\boldsymbol{y} \in \mathcal{Q}^i$, $\boldsymbol{x}$ can be solved by $\boldsymbol{x} = [\phi_b(\boldsymbol{s}^i)]_{\boldsymbol{x}}^{-1}(\boldsymbol{y} - \phi_b^{\mathbf{0}}(\boldsymbol{s}^i) - [\phi_b(\boldsymbol{s}^i)]_{\boldsymbol{z}}(\boldsymbol{z} - \boldsymbol{z}^i))$. When $r > 1$, we solve the $\lfloor r \rfloor + 1$-th order polynomial on $\mathcal{B}^i(\epsilon)$ numerically, as in Lyu et al. (2022).

Then, $\Phi$ defined in (68) satisfies the two conditions. Due to $\|\boldsymbol{s} - \boldsymbol{s}'\| \geq \sqrt{d_s}\epsilon$ for $\boldsymbol{s} \in \mathcal{B}^i(\epsilon)$ and $\boldsymbol{s}' \in \mathcal{B}^j(\epsilon)$ and the smooth property of $T^{-1}$, we have, for $y \in \mathcal{Q}_i$ and $y' \in \mathcal{Q}_j$,

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{y}'\|_\infty &\geq \|T^0(\boldsymbol{s}) - T^0(\boldsymbol{s}')\|_\infty - \|\boldsymbol{y} - T^0(\boldsymbol{s})\|_\infty - \|y' - T^0(s')\|_\infty \\ &\geq \frac{1}{\sqrt{d}B}\|\boldsymbol{s} - \boldsymbol{s}'\| - 2B\sqrt{d}\frac{1}{K^{r+1}} \geq \frac{1}{B}\epsilon - 2B\sqrt{d}\frac{1}{K^{r+1}} \geq \frac{1}{K^{r+1}}. \end{aligned} \tag{72}$$

The last inequality holds when choosing $\epsilon = \frac{c_\epsilon}{K^{r+1}}$ with $\frac{c_\epsilon}{\sqrt{d}B} - 2B\sqrt{d} = 1$. Hence, $\{\mathcal{Q}_i\}$ are disjoint with distances no less than $\frac{1}{K^2}$.

Let $\mathcal{B}(\epsilon) = \bigcup \mathcal{B}^i(\epsilon)$. The final approximation is constructed with an additional layer with the indicator function,

$$\phi^*(\boldsymbol{s}) = \mathbb{I}_{\mathcal{B}(\epsilon)}(\boldsymbol{s})\phi(\boldsymbol{s}) + (1 - \mathbb{I}_{\mathcal{B}(\epsilon)}(\boldsymbol{s}))\phi'(\boldsymbol{s}).$$

Here, $\phi'(\boldsymbol{s})$ can be set to any bounded and invertible function with bounded derivatives, and the image of $\phi'$ should be disjoint with any $\{\mathcal{Q}^i\}_{i=1}^{K^d}$. So, a simple choice for $\phi'$ is $\phi'(\boldsymbol{s}) = \boldsymbol{s} + B$, where $B$ is the upper bound for $\phi$ and $T^0$. Then $\sup_s \|\phi'(\boldsymbol{s})\|_\infty \leq B + 1$ and $\sup_s \|\nabla_{\boldsymbol{x}}\phi'(\boldsymbol{s})\|_\infty = \|I\|_\infty = 1$.

Furthermore, the indicator function can be implemented by $\mathbb{I}_{\mathcal{B}(\epsilon)}(\boldsymbol{s}) = 1 - \mathbb{I}_{[\frac{1}{K}, \infty)}(\max_j[\phi_a(\boldsymbol{s} + \epsilon) - \phi_a(\boldsymbol{s} - \epsilon)]_j)$. This is derived from the fact that $\|\phi_a(\boldsymbol{s} + \epsilon) - \phi_a(\boldsymbol{s} - \epsilon)\|_\infty \leq \max(\|\phi_a(\boldsymbol{s} + \epsilon) - (j-1)/K\|_\infty, \|j/K - \phi_a(\boldsymbol{s} - \epsilon)\|_\infty) < 1/K$ when $\boldsymbol{s}$ is an interior point in $\mathcal{B}(\epsilon)$, $j = 1, 2, \ldots, K$.

Setting $\epsilon = O(\frac{1}{K^{r+1}})$, we bound the $L_2$ error $\|\phi^* - T^0\|_{\infty,2}$ and $\|\nabla_{\boldsymbol{x}}\phi^* - \nabla_{\boldsymbol{x}}T^0\|_{\infty,2}$,

$$\begin{aligned} \|(\nabla_{\boldsymbol{x}}\phi^*) - (\nabla_{\boldsymbol{x}}T^0)\|_{\infty,2} &\leq \sqrt{\int_{[0,1]^d/(\mathcal{B}(\epsilon))} \|\nabla_{\boldsymbol{x}}\phi' - \nabla_{\boldsymbol{x}}T^0\|_\infty^2} + \sqrt{\int_{\mathcal{B}(\epsilon)} \|\nabla_{\boldsymbol{x}}\phi - \nabla_{\boldsymbol{x}}T^0\|_\infty^2} \\ &\leq Kd(1 + B)\epsilon + B\sqrt{d}\frac{1}{K^r} \leq (dBc_\epsilon + B\sqrt{d})\frac{1}{K^r}. \end{aligned} \tag{73}$$

and

$$\|\phi^* - T^0\|_{\infty,2} \leq \sqrt{\int_{[0,1]^d/(\mathcal{B}(\epsilon))} \|\phi' - T^0\|_\infty^2} + \sqrt{\int_{\mathcal{B}(\epsilon)} \|\phi - T^0\|_\infty}$$
$$\leq 2KdB\epsilon + B\sqrt{d}\frac{1}{K^r} \leq (2 + c_\epsilon dB)\frac{1}{K^r}, \tag{74}$$

leading to the desired result. This completes the proof.

**Implementation and optimization.** Achieving an invertible estimation of $T^0$ requires minimizing the negative log-likelihood function subject to the invertibility constraints (70) and (71). This optimization involves bi-level optimization, wherein lower-level optimization concerning $\boldsymbol{s}$ is performed within each constraint, while the upper-level optimization is conducted simultaneously. To simplify this bi-level optimization, one can reformulate it as single-level unconstrained optimization using regularization or the Karush-Kuhn-Tucker (KKT) condition, as discussed in Sinha et al. (2017). This reformulated problem can then be solved efficiently using stochastic gradient descent. ■

**Lemma 32 (Metric entropy)** *For the neural network class* $\Theta = \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})$ *defined in Lemma 30, the metric entropy of* $\mathcal{F} = \{l(\cdot;\theta) - l(\cdot;\pi\theta^0) : \theta \in \Theta\}$ *is bounded by* $H_B(u, \mathcal{F}) = O(K^2 \log^4 K \log K/u)$.

**Proof** [Lemma 32] By (63), for any $\theta_1, \theta_2 \in \Theta$, the likelihood ratio is bounded. So there exists a $c_r, c_r^a, c_r^b > 0$ such that for any $\boldsymbol{x}, \boldsymbol{z}$,

$$|l(\boldsymbol{x},\boldsymbol{z},\theta_1) - l(\boldsymbol{x},\boldsymbol{z},\theta_2)| = \left|\log(\frac{p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{z},\theta_2)}{p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{z},\theta_1)} - 1 + 1)\right| \leq c_r |p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{z},\theta_2) - p_{\boldsymbol{x}|\boldsymbol{z}}(\boldsymbol{x},\boldsymbol{z},\theta_1)|$$
$$\leq c_r^a \|\theta_1(\boldsymbol{x},\boldsymbol{z}) - \theta_2(\boldsymbol{x},\boldsymbol{z})\|_\infty + c_r^b |\,|\det(\theta_2(\boldsymbol{x},\boldsymbol{z}))| - |\det(\theta_1(\boldsymbol{x},\boldsymbol{z}))|\,|.$$

In the detailed setting of $\Theta$ in (68) in the proof of Lemma 30, we denote the neural network $\Phi$ in $\theta_1$ and $\theta_2$ as $\Phi(\boldsymbol{x},\boldsymbol{z};\theta_1)$ and $\Phi(\boldsymbol{x},\boldsymbol{z};\theta_2)$, respectively. Then $\|\theta_1(\boldsymbol{x},\boldsymbol{z}) - \theta_2(\boldsymbol{x},\boldsymbol{z})\|_\infty \leq \|\Phi(\boldsymbol{x},\boldsymbol{z};\theta_1) - \Phi(\boldsymbol{x},\boldsymbol{z};\theta_2)\|_\infty$ and $\big|\,|\det(\theta_1(\boldsymbol{x},\boldsymbol{z}))| - |\det(\theta_2(\boldsymbol{x},\boldsymbol{z}))|\big| \leq \sum_{|\boldsymbol{\alpha}|\leq\lfloor r\rfloor} c_d \|\phi_b^{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{z};\theta_1) - \phi_b^{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{z};\theta_2)\|_\infty$. Both $\Phi$ and $\phi_b$ belong to $\mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})$. Hence,

$$H_B(u,\mathcal{F}) \leq H(\frac{1}{2c_r^a}\frac{u}{2}, \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})) + r^{d_x+d_z} H(\frac{1}{2c_r^b c_d r^{d_x+d_z}}\frac{u}{2}, \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})).$$

By Lemma 4.2 in Oko et al. (2023), $H(u, \mathrm{NN}(\mathbb{L}, \mathbb{W}, \mathbb{S}, \mathbb{B}, \mathbb{E})) = O(\mathbb{S}\mathbb{L}\log(\mathbb{L}\mathbb{E}\mathbb{W}/u)) = O(W^2L^2\log^2 W \log^2 L \log(WL/u)) = O(K^2\log^4 K \log K/u)$ with $K = WL$. Although $\Theta$ is constructed with ReLU and ReQU layers, there are only ReQU layers in fixed depth and width, so the entropy bound for the ReLU network still holds here when $K$ is large enough. Therefore, $H(u,\mathcal{F}) = O(K^2\log^4 K \log K/u)$. This completes the proof. ■

Lemma 33 provides the perturbation bound for matrix determinants for the proof of Lemma 32.

**Lemma 33 (Theorem 2.12 of Ipsen and Rehman (2008))** *Let $A$ and $E$ be $d \times d$ matrices. Then*

$$|\det(A+E) - \det(A)| \leq d\|E\|_2 \max(\|A\|_2, \|A+E\|_2)^{d-1},$$

*where $\|A\|_2$ is the spectral norm of the matrix $A$.*

### A.3.2 PROOFS OF SUBSECTION 4.2

Theorem 34 gives the non-asymptotic probability bound for the generation error in Theorem 5.

**Theorem 34 (Conditional flows via transfer learning)** *Under Assumptions 1-3 and 7, there exists a coupling network $\Theta_t$ in* (15) *with specific hyperparameters: $\mathbb{L}_t = c_L L \log L$, $\mathbb{W}_t = c_W W \log W$, $\mathbb{S}_t = c_S W^2 L \log^2 W \log L$, $\mathbb{B}_t = c_B$, $\log \mathbb{E}_t = c_E \log(WL)$, and $\lambda = c_\lambda$, such that the error of conditional flow generation through transfer learning under the KL divergence is*

$$\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \hat{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\varepsilon_t + \sqrt{3c_1}\varepsilon_s),$$

*with a probability exceeding $1 - \exp(-c_{10} n_t (x\varepsilon_t)^2) - \exp(-c_2 n_s^{1-\xi}(x\varepsilon_s)^2)$ for any $x \geq 1$ and some constant $c_{10} > 0$. Here, $c_L, c_W, c_S, c_B, c_E, c_\lambda$ are sufficiently large constants, $WL \asymp n_t^{\frac{d_{x_t}+d_{h_t}}{2(d_{x_t}+d_{h_t}+2r_t)}}$ and $\varepsilon_t \asymp n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{5/2} n_t$.*

**Proof** [Theorems 5 and 34] As in the proof of Theorem 1, we prove Theorem 5 by applying the approximation error bound from Lemma 30 and the estimation error bound from Lemma 32 as outlined in Theorem 9. ■

Theorem 35 gives the non-asymptotic probability bound for the generation error in Theorem 6.

**Theorem 35 (Non-transfer conditional flows)** *Under Assumption 7, there exists a coupling network $\Theta_t$ of the same configurations as in Theorem 5, the generation error of the non-transfer conditional flow is*

$$\mathrm{E}_{\boldsymbol{z}_t}[\mathcal{K}^{1/2}(p^0_{\boldsymbol{x}_t|\boldsymbol{z}_t}, \tilde{p}_{\boldsymbol{x}_t|\boldsymbol{z}_t})] \leq x(\varepsilon_t + \varepsilon_t^h)$$

*with a probability exceeding $1 - \exp(-c_{10} n_t (x(\varepsilon_t + \varepsilon_t^h))^2)$ for any $x \geq 1$ and some constant $c_{10} > 0$. Here, $\varepsilon_t \asymp n_t^{-\frac{r_t}{d_{x_t}+d_{h_t}+2r_t}} \log^{5/2} n_t$.*

**Proof** [Theorems 6 and 35] Theorem 6 directly follows from the general result in Theorem 29, which provides the bound for $\varepsilon_t^h$ as established in Lemma 23. ■

We set $\Theta_s = \mathrm{CF}(\mathbb{L}_s, \mathbb{W}_s, \mathbb{S}_s, \mathbb{B}_s, \mathbb{E}_s, \lambda_s)$ and estimate the mapping $\hat{T}_s$ by minimizing the negative log-likelihood,

$$\begin{aligned}\hat{T}_s = \hat{\theta}_s(\boldsymbol{x}_s, \hat{h}(\boldsymbol{z})) &= \arg\min_{\theta_s \in \Theta_t, h \in \Theta_h} \sum_{i=1}^{n_s} -\log p_{\boldsymbol{x}_s|\boldsymbol{z}_s}(\boldsymbol{x}_s^i, \boldsymbol{z}_s^i; \theta_t; h) \\ &= \arg\min_{\theta_s \in \Theta_s, h \in \Theta_h} -\log p_{\boldsymbol{v}}(\theta_s(\boldsymbol{x}_s^i, h(\boldsymbol{z}_s^i))) - \log\left|\det(\nabla_{\boldsymbol{x}} \theta_s(\boldsymbol{x}_s^i, h(\boldsymbol{z}_s^i)))\right|,\end{aligned} \tag{75}$$

Next, we specify some assumptions on the true $T_s^0$.

**Assumption 15** *There exists a map $T_s^0(\boldsymbol{x}_s, \boldsymbol{z}_s) = \theta_s^0(\boldsymbol{x}_s, h_s(\boldsymbol{z}_s))$ such that $\boldsymbol{V} = T_s^0(\boldsymbol{X}_s, \boldsymbol{Z}_s)$, where $\boldsymbol{V}$ a random vector with a known lower bounded smooth density $p_v \in \mathcal{C}^\infty([0,1]^{d_{x_s}}, \mathbb{R}, B_v)$. Assume that $T_s^0$ and its inverse belong to a Hölder ball $\mathcal{C}^{r_s+1}([0,1]^{d_{x_s}+d_{h_s}}, [0,1]^{d_{x_s}}, B_s)$ , while $|\det \nabla_{\boldsymbol{x}} T_s^0|$ is lower bounded by some positive constant.*

Then, we obtain the source error rate $\varepsilon_s$.

**Lemma 36 (Source generation error)** *Under Assumption 15, setting the hyperparameters of $\Theta_s$ with sufficiently large positive constant set $\{c_L, c_W, c_S, c_B, c_E, c_\lambda\}$ as follows: $\mathbb{L}_s = c_L L \log L$, $\mathbb{W}_s = c_W W \log W$, $\mathbb{S}_s = c_S W^2 L \log^2 W \log L$, $\mathbb{B}_s = c_B$, $\log \mathbb{E}_s = c_E \log(WL)$ and $\lambda_s = c_\lambda$, we obtain that for any $x \geq 1$,*

$$P(\rho_s(\gamma_s^0, \hat{\gamma}_s) \geq x\varepsilon_s) \leq \exp(-c_2 n_s (x\varepsilon_s)^2),$$

*with $\varepsilon_s = \beta_s + \delta_s + \varepsilon_s^h$, some constant $c_2 > 0$ same in Assumption 3. Here, $\beta_s = \sqrt{\frac{K^2 \log^5 K}{n_s}}$, $\delta_s = K^{\frac{-2r_s}{d_s + d_{h_s}}}$ with $K = WL$, and $\varepsilon_s^h$ satisfies* (59) *in Lemma 23. Moreover, setting $K \asymp n_s^{\frac{d_s}{2(d_{x_s}+d_{h_s}+2r_s)}}$ yields*

$$\varepsilon_s \asymp \log^{5/2} n_s \left( n_s^{-\frac{r_s}{d_{x_s}+d_{h_s}+2r_s}} \right) + \varepsilon_s^h.$$

**Proof** This proof of this lemma is the same as that of Theorem 6, replacing the approximation error and metric entropy there by those in Lemma 30 and Lemma 32. ■

### A.3.3 PROOFS OF SUBSECTION 4.3

Theorems 37 and 38 present the formal version of Theorems 7 and 8 respectively.

**Theorem 37 (Unconditional flows via transfer learning)** *Under Assumptions 4 and 8, there exists a wide or deep ReLU network $\Theta_{g_t}$ with specific hyperparameters: $\mathbb{L}_g = c_L L \log L$, $\mathbb{W}_g = c_W W \log W$, $\mathbb{S}_g = c_S W^2 L \log^2 W \log L$, $\mathbb{B}_g = c_B$, and $\log \mathbb{E}_g = c_E \log(WL)$, such that the error for unconditional flow generation via transfer learning in Wasserstein distance is,*

$$P(\mathcal{W}(P_{\boldsymbol{x}_t}^0, \hat{P}_{\boldsymbol{x}_t}) \geq x(\varepsilon_t + \varepsilon_s^u)) \leq \exp(-c_3 n_s (x\varepsilon_s^u)^2) + \exp(-c_{11} n_t (x\varepsilon_t)^2)$$

*for any $x \geq 1$ and some constant $c_{11} > 0$. Here, $c_L, c_W, c_S, c_B, c_E$ are sufficiently large positive constants, $WL \asymp n_t^{\frac{d_u}{2(d_u+2r_g)}}$ and $\varepsilon_t \asymp n_t^{-\frac{r_g}{d_u+2r_g}} \log^{\frac{5}{2}} n_t$.*

**Theorem 38 (Non-transfer unconditional flows)** *Suppose there exists a sequence $\varepsilon_t^u$ indexed by $n_t$ such that $n_t^{1-\xi}(\varepsilon_t^u)^2 \to \infty$ as $n_s \to \infty$ and $P(\rho_u(\theta_u^0, \tilde{\theta}_u) \geq \varepsilon) \leq \exp(-c_3 n_t^{1-\xi} \varepsilon^2)$ for any $\varepsilon \geq \varepsilon_t^u$ and some constants $c_3, \xi > 0$. Under Assumption 8 the same conditions of Theorem 7, the error in non-transfer diffusion generation under the Wasserstein distance is, for any $x \geq 1$,*

$$P(\mathcal{W}(P_{\boldsymbol{x}_t}^0, \tilde{P}_{\boldsymbol{x}_t}) \geq x(\varepsilon_t + \varepsilon_t^u)) \leq \exp(-c_3 n_t (x\varepsilon_t^u)^2) + \exp(-c_{11} n_t (x\varepsilon_t)^2).$$

*Here* $\varepsilon_t \asymp n_t^{-\frac{r_g}{d_u+2r_g}} \log^{\frac{5}{2}} n_t$.

The results in Section 4.3 can be proved similarly as those in Section 3.2, except that we derive the generation error in the latent variable $\boldsymbol{U}$ from the flow model theory in Theorem 29.

**Assumption 16** *There exists a map* $T_u^0 : [0,1]^{d_u} \times \to [0,1]^{d_u}$ *such that* $\boldsymbol{V} = T_u^0(\boldsymbol{U})$*, where* $\boldsymbol{V}$ *a random vector with a known lower bounded smooth density in* $\mathcal{C}^\infty([0,1]^{d_u}, \mathbb{R}, B_v)$*. Assume that* $T_u^0(\boldsymbol{v})$ *and its inverse belong to a Hölder ball* $\mathcal{C}^{r_u+1}([0,1]^{d_u}, [0,1]^{d_u}, B_u)$ *of radius* $B_u > 0$*, while the* $|\det \nabla T_u^0|$ *is lower bounded by some positive constant.*

**Lemma 39 (Latent generation error)** *Under Assumption 16, setting network* $\Theta_u$*'s hyperparameters with a set of sufficiently large positive constants* $\{c_L, c_W, c_S, c_B, c_E, c_\lambda\}$ *as follows:* $\mathbb{L}_u = c_L L \log L$*,* $\mathbb{W}_u = c_W W \log W$*,* $\mathbb{S}_u = c_S W^2 L \log^2 W \log L$*,* $\mathbb{B}_u = c_B$*,* $\log \mathbb{E}_u = c_E \log(WL)$*, and* $\lambda_u = c_\lambda$*, we obtain that for any* $x \geq 1$*,*

$$P(\rho(\theta_u^0, \hat{\theta}_u) \geq x\varepsilon_s^u) \leq \exp\Big( - c_3 n_s (x\varepsilon_s^u)^2 \Big),$$

*with* $\varepsilon_s^u = \beta_u + \delta_u$ *and* $c_3 > 0$ *is the same with Assumption 4. Here,* $\beta_u \asymp \sqrt{\frac{K^2 \log^5 K}{n_s}}$*,* $\delta_u \asymp K^{\frac{-2r_u}{d_u}}$ *with* $K = WL$*. Moreover, setting* $K \asymp n_s^{\frac{d_s}{2(d_u+2r_u)}}$ *yields* $\varepsilon_s^u \asymp n_s^{-\frac{r_u}{d_u+2r_u}} \log^{5/2} n_s$*. Similarly,* $\varepsilon_t^u \asymp n_t^{-\frac{r_u}{d_u+2r_u}} \log^{5/2} n_t$*.*

### A.4 Auxiliary lemmas on neural network approximation theory

This section restates several neural network approximation results for various functions, which are used in our proofs.

The following lemma constructs a ReLU network to approximate a step function. Subsequently, denote by $\lceil x \rceil$ the ceiling of $x$ and denote by $\mathbb{N}$ and $\mathbb{N}^+$ all non-negative and positive integers.

**Lemma 40 (Step function, Proposition 4.3 of Lu et al. (2021))** *For any* $W, L, d \in \mathbb{N}^+$ *and* $\epsilon > 0$ *with* $K = \lceil W^{1/d} \rceil^2 \lceil L^{2/d} \rceil$ *and* $\epsilon \leq \frac{1}{3K}$*, there exists a one-dimensional ReLU network* $\phi$ *with width* $4W + 5$ *and depth* $4L + 4$ *such that*

$$\phi(x) = \frac{k}{K}, \quad \textit{if } x \in \left[ \frac{k}{K}, \frac{k+1}{K} - \epsilon \cdot \mathbb{I}_{\{k<K-1\}} \right]; k = 0, 1, \ldots, K-1.$$

*Moreover,* $\phi(\boldsymbol{x})$ *is linear in* $[\frac{k+1}{K} - \epsilon \cdot \mathbb{I}_{\{k<K-1\}}, \frac{k+1}{K}]$*.*

The following result allows us to construct a ReLU network with width $O(s\sqrt{W \log W})$ and depth $O(L \log L)$ to approximate function values at $O(W^2 L^2)$ points with an error $O(W^{-2s} L^{-2s})$.

**Lemma 41 (Point fitting, Proposition 4.4 of Lu et al. (2021))** *Given any* $W, L, s \in \mathbb{N}^+$ *and* $\zeta_i \in [0,1]$ *for* $i = 0, 1, \ldots, W^2 L^2 - 1$*, there exists a ReLU network* $\phi$ *with width* $8s(2W + 3) \log_2(4W)$ *and depth* $(5L + 8) \log_2(2L)$ *such that*

1. $|\phi(i) - \zeta_i| \leq W^{-2s} L^{-2s}$*, for* $i = 0, 1, \ldots, W^2 L^2 - 1$*;*

2. $0 \leq \phi(x) \leq 1$, *for any* $x \in \mathbb{R}$.

The following is a ReLU approximation result for a Hölder class of smooth functions, which is a simplified version of Theorem 1.1 in Lu et al. (2021) and Lemma 11 in Huang et al. (2022).

**Lemma 42 (Lemma 11 in Huang et al. (2022))** *For any $f \in \mathcal{C}^r([0,1]^d, \mathbb{R}, B)$, there exists a ReLU network $\Phi$ with $\mathbb{W} = c_W(W \log W)$, $\mathbb{L} = c_L(L \log L)$ and $\mathbb{E} = (WL)^{c_E}$ with some positive constants $c_W$, $c_L$ and $c_E$ dependent on $d$ and $r$, such that $\sup_{\boldsymbol{x} \in [0,1]^d} |\Phi(\boldsymbol{x}) - f(\boldsymbol{x})| = O(B(WL)^{-\frac{2r}{d}})$.*

The next lemma describes how to construct a ReQU network to approximate the multinomial function.

**Lemma 43 (Lemma 1 of Belomestny et al. (2023))** *For any $x = (x_1, \ldots, x_k) \in \mathbb{R}^k$ with $k \in \mathbb{N}, k \geq 2$, there exists a ReQU neural network $NN\left(\lceil \log_2 k \rceil, (k, 2^{\lceil \log_2 k \rceil + 1}, 2^{\lceil \log_2 k \rceil}, \ldots, 4, 1)\right)$, which implements the map $x \mapsto x_1 x_2, \ldots, x_k$. Moreover, this network contains at most $5 \cdot 2^{2\lceil \log_2 k \rceil}$ non-zero weights.*

## Appendix B. Experiment details

### B.1 Conditional image generation

**Datasets and Preprocessing.** We conduct our experiments on two standard handwritten-digit benchmarks. For source data, we use the MNIST training set (60,000 samples, $28 \times 28$ grayscale), and for target data we use the USPS training set (7,291 samples, $16 \times 16$ grayscale). All images are resized to $16 \times 16$, converted to tensors, and normalized to $[-1, 1]$. We randomly split USPS into 70% train (5,103 samples) and 30% test (2,188 samples) with a fixed random seed for reproducibility. DataLoaders use a batch size of 256 and shuffle the training splits.

**Model architecture.** We propose a conditional diffusion model given digit label based on a UNet. Our `ClassConditionedUnet` consists of: (1) A learnable embedding layer mapping each digit label $y \in \{0, \ldots, 9\}$ to a 4-dimensional vector. (2) A `UNet2DModel` (from HuggingFace Diffusers) with input channels $1 + 4 = 5$, output channels 1, three down-sampling blocks {`DownBlock2D`, `AttnDownBlock2D`×2}, three up-sampling blocks {`AttnUpBlock2D`×2, `UpBlock2D`}, and 2 ResNet layers per block. At each forward pass, we concatenate the expanded class embedding to the image tensor and predict the noise residual.

**Diffusion and optimization.** We use a DDPMScheduler with 1,000 timesteps and the "squaredcos_cap_v2" beta schedule. During training, we optimize all parameters with Adam (learning rate $1 \times 10^{-4}$) for 30 epochs on MNIST, recording the loss at each iteration.

**Fine-tuning on USPS.** After MNIST pretraining, we freeze the class-embedding weights and continue training only the UNet backbone on the USPS training split for an additional 30 epochs (Adam, learning rate $1 \times 10^{-4}$). This transfers digit-conditioned features learned on MNIST into the USPS domain.

**Evaluation.** We generate samples by starting from Gaussian noise and iteratively denoising with the learned model, conditioned on test-set labels. Evalaution metrics (Wasserstein distance) are reported on the USPS test split.

### B.2 Unconditional image generation

**Datasets and preprocessing.** We extract the digit "3" images from both MNIST and USPS. All images are resized to $16 \times 16$, converted to tensors, and normalized to $[-1, 1]$. From MNIST, we take all "3" examples ($N_{\text{MNIST},3}$), and from USPS we similarly filter to $N_{\text{USPS},3}$ examples, then split the latter into 70% train and 30% test using a fixed random seed. DataLoaders use batch size 256, shuffling the training splits.

**VAE architecture and joint training.** We train two independent VAEs (`AutoencoderKL`)—one for MNIST digit "3" and one for USPS digit "3." Each VAE comprises four down-sampling encoder blocks and four symmetric decoder blocks, with 10 convolutional layers per block and a latent dimension of 64.

Both networks are optimized *jointly* for 1 000 epochs using Adam (learning rate $1 \times 10^{-4}$). The overall loss function to minimize is

$$L = \alpha_{\text{MNIST}}\, L_{\text{VAE}}^{(\text{MNIST})} + \alpha_{\text{USPS}}\, L_{\text{VAE}}^{(\text{USPS})} + \lambda_{\text{align}}\, MMD\big(z^{(\text{MNIST})}, z^{(\text{USPS})}\big),$$

where $L_{\text{VAE}}^{(\cdot)}$ denotes the loss for VAE, $\alpha_{\text{MNIST}} = \alpha_{\text{USPS}} = 0.1$, $\lambda_{\text{align}} = 10$, and $z^{(\cdot)}$ denotes latent samples from the corresponding VAE. Here MMD encourages the two latent distributions to overlap, aligning the representations learned from MNIST and USPS.

**Maximum Mean Discrepancy (MMD).** Given samples $\boldsymbol{z}^{(1)} = \{z_i^{(1)}\}_{i=1}^m$ and $\boldsymbol{z}^{(2)} = \{z_j^{(2)}\}_{j=1}^n$ from two distributions, the MMD with kernel $k$ is

$$MMD^2(\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}) = \frac{\sum_{i \neq i'} k\big(\boldsymbol{z}_i^{(1)}, \boldsymbol{z}_{i'}^{(1)}\big)}{m(m-1)} + \frac{\sum_{j \neq j'} k\big(\boldsymbol{z}_j^{(2)}, \boldsymbol{z}_{j'}^{(2)}\big)}{n(n-1)} - \frac{2\sum_{i=1}^m \sum_{j=1}^n k\big(z_i^{(1)}, z_j^{(2)}\big)}{mn},$$

where we employ a Gaussian (RBF) kernel $k(x, y) = \exp\Big(-\frac{\|x-y\|^2}{2\sigma^2}\Big)$ with $\sigma = 1.0$.

**UNet architecture for latent diffusion.** To model the latent distribution, we use a `UNet2DModel` with the following configuration: **Input/Output channels:** $C_{\text{in}} = C_{\text{out}} = 64$ (latent dimensionality). **Sample size:** $16 \times 16$ spatial resolution. **Block structure:** *Down-sampling*: two blocks, both `AttnDownBlock2D`. *Up-sampling*: two blocks, `AttnUpBlock2D` then `UpBlock2D`. **Block channels:** `block_out_channels` = (128, 256). **Depth:** `layers_per_block` = 10 ResNet layer per block. **Normalization:** `GroupNorm` with `norm_num_groups=1`.

**Latent extraction and subsampling.** After alignment, we encode all MNIST-"3" images into their 64-dimensional latents and aggregate them into a pool. To study sample-efficiency, we randomly select subsets of size $k \in \{100, 500, 1000, 2000, 3500, 6000\}$ from this pool to serve as training data for diffusion.

**Unconditional diffusion training.** For each subset size $k$, we train the UNet on the $k$ MNIST latents for 30 epochs. We use Adam optimizer with learning rate as $1 \times 10^{-4}$.

**Evaluation.** We generate USPS-"3" test samples (198 images) by denoising from Gaussian noise, then use the Sinkhorn algorithm to approximate the Wasserstein-1 distance (blur = 0.05) between generated and real USPS latents using the Python GeomLoss library.

## References

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models. *arXiv preprint arXiv:2111.13606*, 2021.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12: 149–198, 2000.

Denis Belomestny, Eric Moulines, Alexey Naumov, Nikita Puchkin, and Sergey Samsonov. Rates of convergence for density estimation with gans. *arXiv preprint arXiv:2102.00199*, 2021.

Denis Belomestny, Alexey Naumov, Nikita Puchkin, and Sergey Samsonov. Simultaneous approximation of a smooth function and its derivatives by deep neural networks with piecewise-polynomial activations. *Neural Networks*, 161:242–253, 2023.

Fabio Maria Carlucci, Paolo Russo, Tatiana Tommasi, and Barbara Caputo. Hallucinating agnostic images to generalize across domains. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3227–3234. IEEE, 2019.

Hongrui Chen, Holden Lee, and Jianfeng Lu. Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions. In *International Conference on Machine Learning*, pages 4735–4763. PMLR, 2023a.

Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023b.

Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023c.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8188–8197, 2020.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Yaël Frégier and Jean-Baptiste Gouray. Mind2mind: transfer learning for gans. In *Geometric Science of Information: 5th International Conference, GSI 2021, Paris, France, July 21–23, 2021, Proceedings 5*, pages 851–859. Springer, 2021.

Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.

Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

Elizabeth Gibney. Is ai fuelling a reproducibility crisis in science. *Nature*, 608:250–251, 2022.

Xiaonan Hu and Xinyu Zhang. Optimal parameter-transfer learning by semiparametric model averaging. *Journal of Machine Learning Research*, 24:1–53, 2023.

Chin-Wei Huang, Laurent Dinh, and Aaron Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

Jian Huang, Yuling Jiao, Zhen Li, Shiao Liu, Yang Wang, and Yunfei Yang. An error analysis of generative adversarial networks for learning distributions. *Journal of machine learning research*, 23(116):1–43, 2022.

Ilse CF Ipsen and Rizwana Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776, 2008.

Isao Ishikawa, Takeshi Teshima, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Universal approximation property of invertible neural networks. *Journal of Machine Learning Research*, 24(287):1–68, 2023.

Bangti Jin, Zehui Zhou, and Jun Zou. On the approximation of bi-lipschitz maps by invertible neural networks. *Neural Networks*, page 106214, 2024.

Tero Karras, Miika Aittala, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Training generative adversarial networks with limited data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 12104–12114, 2020.

Frederic Koehler, Viraj Mehta, and Andrej Risteski. Representational aspects of depth and conditioning in normalizing flows. In *International Conference on Machine Learning*, pages 5628–5636. PMLR, 2021.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

Bing Li. *Sufficient dimension reduction: Methods and applications with R*. CRC Press, 2018.

Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

Yifei Liu, Rex Shen, and Xiaotong Shen. Novel uncertainty quantification through perturbation-assisted sample synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, doi.org/10.1109/TPAMI.2024.3393364, 2024.

Jianfeng Lu, Zuowei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation for smooth functions. *SIAM Journal on Mathematical Analysis*, 53(5):5465–5506, 2021.

Junlong Lyu, Zhitang Chen, Chang Feng, Wenjing Cun, Shengyu Zhu, Yanhui Geng, Zhijie Xu, and Chen Yongwei. Para-cflows: $c^k$-universal diffeomorphism approximators as superior neural surrogates. *Advances in Neural Information Processing Systems*, 35:28829–28841, 2022.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

Mina Ossiander. A central limit theorem under metric entropy with l2 bracketing. *The Annals of Probability*, pages 897–919, 1987.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Michał Różański, Roman Wituła, and Edyta Hetmaniok. More subtle versions of the hadamard inequality. *Linear Algebra and its Applications*, 532:500–511, 2017.

Xiaotong Shen and Wing Hung Wong. Convergence rate of sieve estimates. *The Annals of Statistics*, pages 580–615, 1994.

Xiaotong Shen, Yifei Liu, and Rex Shen. Boosting data analytics with synthetic volume expansion. *arXiv preprint arXiv:2310.17848*, 2023.

Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22 (2):276–295, 2017.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2020.

Takeshi Teshima, Isao Ishikawa, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Coupling-based invertible neural networks are universal diffeomorphism approximators. *Advances in Neural Information Processing Systems*, 33:3362–3373, 2020.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in neural information processing systems*, 33:7852–7862, 2020.

Xiyu Wang, Baijiong Lin, Daochang Liu, and Chang Xu. Efficient transfer learning in diffusion models via adversarial noise. *arXiv preprint arXiv:2308.11948*, 2023.

Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve mles. *The Annals of Statistics*, pages 339–362, 1995.

Wen Zhang, Lingfei Deng, Lei Zhang, and Dongrui Wu. A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, 10(2):305–329, 2022.

Yufan Zhou, Bingchen Liu, Yizhe Zhu, Xiao Yang, Changyou Chen, and Jinhui Xu. Shifted diffusion for text-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10157–10166, 2023.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2223–2232, 2017.