Journal of Machine Learning Research 26 (2025) 1-54 Submitted 6/23; Revised 2/25; Published 4/25

Adaptive Distributed Kernel Ridge Regression: A Feasible Distributed Learning Scheme for Data Silos

Shao-Bo Lin sblin1983@gmail.com
Xiaotong Liu ariesoomoon@gmail.com
Di Wang* wang.di@xjtu.edu.cn
Center for Intelligent Decision-Making and Machine Learning,
School of Management,
Xi'an Jiaotong University, Xi'an, China

Hai Zhang† zhanghai@nwu.edu.cn
School of Mathematics,
Northwest University, Xi'an, China

Ding-Xuan Zhou dingxuan.zhou@sydney.edu.au
School of Mathematics and Statistics,
University of Sydney, Sydney, Australia

Editor: John Shawe-Taylor

Abstract

Data silos, mainly caused by privacy and interoperability, significantly constrain collaborations among different organizations with similar data for the same purpose. Distributed learning based on divide-and-conquer provides a promising way to settle the data silos, but it suffers from several challenges, including autonomy, privacy guarantees, and the necessity of collaborations. This paper focuses on developing an adaptive distributed kernel ridge regression (AdaDKRR) by taking autonomy in parameter selection, privacy in communicating non-sensitive information, and the necessity of collaborations for performance improvement into account. We provide both solid theoretical verifications and comprehensive experiments for AdaDKRR to demonstrate its feasibility and effectiveness. Theoretically, we prove that under some mild conditions, AdaDKRR performs similarly to running the optimal learning algorithms on the whole data, verifying the necessity of collaborations and showing that no other distributed learning scheme can essentially beat AdaDKRR under the same conditions. Numerically, we test AdaDKRR on both toy simulations and two real-world applications to show that AdaDKRR is superior to other existing distributed learning schemes. All these results show that AdaDKRR is a feasible scheme to overcome data silos, which are highly desired in numerous application regions such as intelligent decision-making, pricing forecasting, and performance prediction for products.

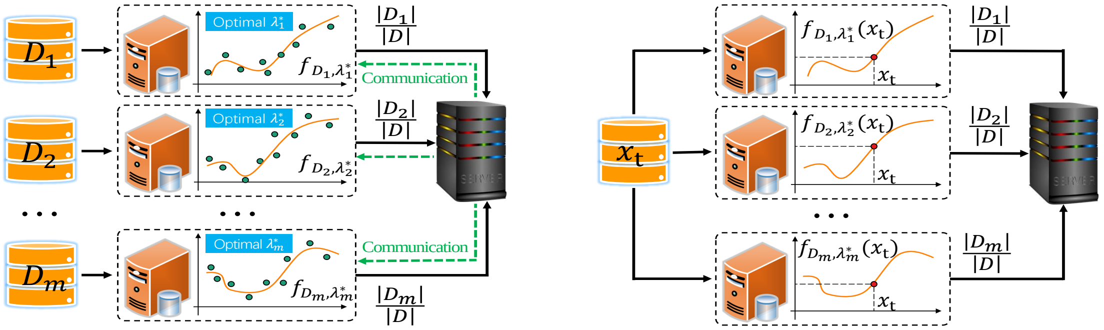
Keywords: distributed learning, data silos, learning theory, kernel ridge regression

1. Introduction

Big data has made profound impacts on people's decision-making, consumption patterns, and ways of life (Davenport et al., 2012; Tambe, 2014), with many individuals now making

*. Corresponding author

†. Corresponding author



(a) Flow of training (b) Flow of testing

Figure 1: Training and testing flows of DKRR

decisions based on analyzing data rather than consulting experts; shopping online based on historical sales data rather than going to physical stores; gaining insights into consumer behaviors and preferences based on the consumption data rather than language communication. With the help of big data, organizations can identify patterns, trends, and correlations that may not be apparent in data of small size, which leads to more accurate predictions, better understanding of behaviors, and improved operational efficiencies.

However, data privacy and security (Jain et al., 2016; Li and Qin, 2017) have garnered widespread attention, inevitably resulting in the so-called data silos, meaning that large-scale data distributed across numerous organizations cannot be centrally accessed, that is, organizations can only use their own local data but cannot obtain relevant data from elsewhere. For example, a large amount of medical data are stored in fragmented forms in different medical institutions but cannot be effectively aggregated; massive amounts of operational data are distributed among various companies but cannot be centrally accessed; and numerous consumer behavior data are collected by different platforms but cannot become public resources due to privacy factors. Data silos are a significant challenge (Fan et al., 2014) for the use of big data, requiring ingenious multi-party collaboration methods to increase the willingness of data holders to cooperate and improve their efficiency of data analysis without leaking their sensitive information. Designing and developing feasible methods to avoid data silos is a recent focus of machine learning, which not only determines the role that machine learning plays in the era of big data but also guides the future direction of machine learning development.

Distributed learning (Balcan et al., 2012; Lea and Nicoll, 2013) is a promising approach for addressing the data silos, as it enables multiple parties to collaborate and learn from each other's data without having to share their own data. As shown in Figure 1, there are generally three ingredients in a distributed learning scheme. The first one is local processing, in which each local machine (party) runs a specific learning algorithm with its own algorithmic parameters and data to yield a local estimator. The second one is communication, where several useful but non-sensitive pieces of information are communicated with each other to improve the quality of local estimators. To protect data privacy, neither data nor information that could lead to data disclosure is permitted to be communicated. The last one is synthesization, in which all local estimators are communicated to the global machine to synthesize a global estimator. In this way, multiple parties can collaborate on solving problems that require access to the whole data from different sources while also addressing privacy concerns, as sensitive data are kept in their original locations and only non-sensitive information is shared.

Due to the success in circumventing the data silos, numerous distributed learning schemes with solid theoretical verification have been developed, including distributed linear regression (Zhang et al., 2013), distributed online learning (Dekel et al., 2012), distributed conditional maximum entropy learning (Mcdonald et al., 2009), distributed kernel ridge regression (Zhang et al., 2015), distributed local average regression (Chang et al., 2017a), distributed kernel-based gradient descent (Lin and Zhou, 2018), distributed spectral algorithms (Mücke and Blanchard, 2018), distributed multi-penalty regularization algorithms (Guo et al., 2019), distributed multi-pass stochastic gradient methods (Lin and Cevher, 2018), distributed coefficient regularization algorithms (Shi, 2019), distributed learning in the overparameterized regimes (Mücke et al., 2022; Nguyen et al., 2024), distributed robust regression algorithm (Hu and Guo, 2024), and distributed linear approaches in high dimension (Rosenblatt and Nadler, 2016; Sheng and Dobriban, 2020). In particular, these algorithms have been proven to achieve optimal rates of generalization error bounds for their batch counterparts, as long as the algorithm parameters are properly selected and the number of local machines is not too large. However, how to choose appropriate algorithm parameters without sharing the data to achieve the theoretically optimal generalization performance of these distributed learning schemes is still open, because all the existing provable parameter selection strategies, such as the logarithmic mechanism for cross-validation (Liu et al., 2022), generalized cross-validation (Xu et al., 2019), and the discrepancy principle (Celisse and Wahl, 2021), need to access the whole data. This naturally raises the following problem:

Problem 1 *How to develop a feasible parameter selection strategy without communicating the individual data of local machines with each other to equip distributed learning to realize its theoretically optimal generalization performance and successfully circumvent the data silos?*

In this paper, taking distributed kernel ridge regression (DKRR) as an example, we develop an adaptive parameter selection strategy based on communicating non-sensitive information to solve the above problem. Our basic idea is to find a fixed basis, and each local machine computes an approximation of its derived rule (the relationship between the input and output) based on the basis and transfers the coefficients of the basis to the global machine. The global machine then synthesizes all the collected coefficients through a specific synthesis scheme and communicates the synthesized coefficients back to each local machine. In this way, each local machine obtains a good approximation of the global rule and uses this rule for cross-validation to determine its algorithm parameters. The road map of our approach is shown in Figure 2. Using the developed parameter selection strategy, we propose a novel adaptive distributed kernel ridge regression (AdaDKRR) to address the data silos. Our main contributions can be concluded as follows:

- *Methodology novelty:* Since data stored in different local machines cannot be communicated, developing an adaptive parameter selection strategy based on local data to equip distributed learning is not easy. The main novelty of our approach is a nonparametric-to-parametric model transition method, which determines the algorithm parameters of dis-

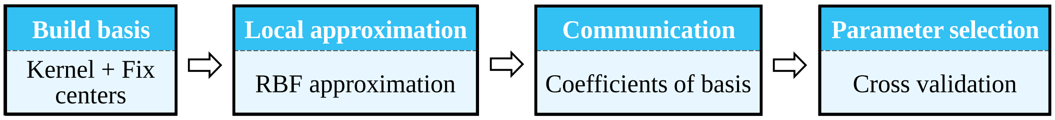


Figure 2: Road map of the proposed parameter selection strategy

tributed nonparametric learning schemes by communicating the coefficients of fixed basis functions without leaking any sensitive information about the local data. With such a novel design, we develop a provable and effective parameter selection strategy for distributed learning based on cross-validation to address the data silos. As far as we know, this is the first attempt at designing provable parameter selection strategies for distributed learning to address the data silos.

- *Theoretical assessments:* Previous theoretical research (Zhang et al., 2015; Lin et al., 2017; Mücke and Blanchard, 2018; Shi, 2019) on distributed learning was carried out with three crucial assumptions: 1) the sizes of data in local machines are almost the same; 2) the parameters selected by different local machines are almost the same; 3) the number of local machines is not so large. In this paper, we present a detailed theoretical analysis by considering the role of the synthesization strategy and removing the assumption of the same data size. Furthermore, we borrow the idea of low-discrepancy sequences (Dick and Pillichshammer, 2010) and the classical radial basis function approximation (Wendland and Rieger, 2005; Rudi et al., 2015) to prove the feasibility of the proposed parameter selection strategy and remove the above-mentioned same parameter assumption. Finally, we provide an optimal generalization rate for AdaDKRR in the framework of statistical learning theory (Cucker and Zhou, 2007; Steinwart and Christmann, 2008), which shows that if the number of local machines is not so large, the performance of AdaDKRR is similar to running kernel ridge regression (KRR) on the whole data. This provides a solid theoretical verification for the feasibility of AdaDKRR to address the data silos.

- *Experimental verification:* We conduct both toy simulations and real-world data experiments to illustrate the excellent performance of AdaDKRR and verify our theoretical assertions. The numerical results show that AdaDKRR is robust to the number of basis functions, which makes the selection of the basis functions easy, thus obtaining satisfactory results. In addition, AdaDKRR shows stable and effective learning performances in parameter selection for distributed learning, regardless of whether the numbers of samples allocated to local machines are the same or not. We also apply AdaDKRR to two real-world data sets, including ones designed to help determine car prices and GPU acceleration models, to test its usability in practice.

The rest of this paper is organized as follows. In the next section, we introduce the challenges, motivations, and some related work of parameter selection in distributed learning. In Section 3, we propose AdaDKRR and introduce some related properties. In Section 4, we provide theoretical evidence of the effectiveness of the proposed adaptive parameter selection strategy and present an optimal generalization error bound for AdaDKRR. In Section 5, we numerically analyze the learning performance of AdaDKRR in toy simulations and two real-world applications. Finally, we draw a simple conclusion. The proofs of all theoretical results and some other relevant information about AdaDKRR are postponed to the Appendix.

2. Challenges, Our Approaches, and Related Work

Let $(\mathcal{H}_K, \|\cdot\|_K)$ be a reproducing kernel Hilbert space (RKHS) induced by a Mercer kernel K (Cucker and Zhou, 2007) on a compact input space $\mathcal{X}$. Suppose that there is a data set $D_j = \{(x_{i,j}, y_{i,j})\}_{i=1}^{|D_j|} \subset \mathcal{X} \times \mathcal{Y}$ stored in the j-th local machine with $1 \leq j \leq m$ and $\mathcal{Y} \subseteq \mathbb{R}$ is the output space. Without loss of generality, we assume that there are no common samples of local machines, i.e. $D_j \cap D_{j'} = \varnothing$ for $j \neq j'$. DKRR with regularization parameters $\vec{\lambda} := \{\lambda_1, \ldots, \lambda_m\}$ is defined by (Zhang et al., 2015; Lin et al., 2017)

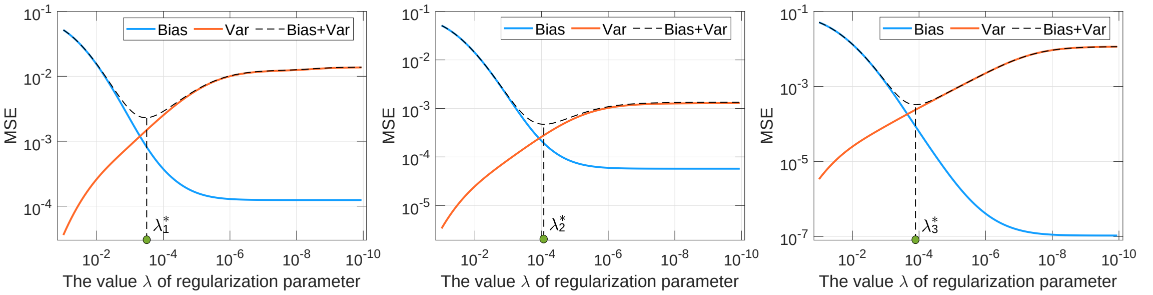
$$\overline{f}_{D,\vec{\lambda}} = \sum_{j=1}^{m} \frac{|D_j|}{|D|} f_{D_j,\lambda_j}, \tag{1}$$

where $\lambda_j > 0$ is a regularization parameter for $j = 1, \ldots, m$, $D = \cup_{j=1}^{m} D_j$, $|D|$ denotes the cardinality of the data set D, and the local estimator f_{D_j,λ_j} is defined by

$$f_{D_j,\lambda_j} = \arg\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{|D_j|} \sum_{(x,y)\in D_j} (f(x) - y)^2 + \lambda_j \|f\|_K^2 \right\}. \tag{2}$$

Therefore, in DKRR defined by (1), each local machine runs KRR (2) on its own data D_j with a specific regularization parameter λ_j to generate a local estimator, and the global machine synthesizes the global estimator $\overline{f}_{D,\vec{\lambda}}$ by using a weighted average based on data sizes. If λ_j is given, then it does not need additional communications in DKRR to handle the data silos. However, how to choose λ_j to optimize DKRR is important and difficult, as data are distributively stored across different local machines and cannot be shared.

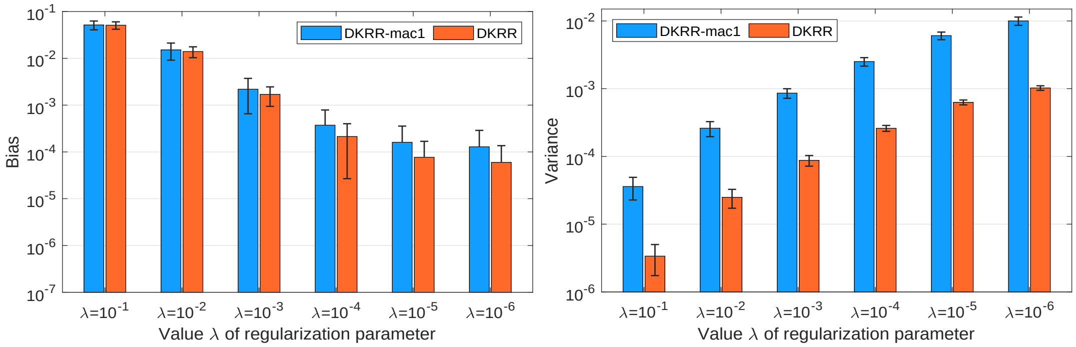
2.1 Challenges and road map for parameter selection in distributed learning



(a) DKRR-machine 1 (b) DKRR (c) KRR

Figure 3: Relationship between bias (variance) and regularization parameter values. Training data $\{(x_i, y_i)\}_{i=1}^{5000}$ are generated by drawing $\{x_i\}_{i=1}^{5000}$ i.i.d. according to the uniform distribution on $[0,1]^3$ and $y_i = g_1(x_i) + \varepsilon$, where $g_1(x)$ is defined by (20) and $\varepsilon \sim \mathcal{N}(0, 0.1^2)$; testing data $\{(x'_i, y'_i)\}_{i=1}^{1000}$ are generated similarly to the training data but with a promise that $y'_i = g_1(x'_i)$. The training samples are uniformly distributed to m local machines, and the number m is set to 10. "DKRR-machine1" represents running KRR on a local machine with a data subset of size $5000/m$.

Recalling that the introduction of the regularization term in (2) is to avoid the well-known over-fitting phenomenon (Cucker and Zhou, 2007) that the derived estimator fits the training data well but fails to predict other queries, the optimal regularization parameter is frequently selected when the bias is close to the variance. However, as shown in Figure 3, if we choose the theoretically optimal regularization parameter based on its own data in each local machine, it is usually larger than the optimal parameter of the global estimator, i.e., $\lambda_1^* > \lambda_2^* \sim \lambda_3^*$ with λ_1^*, λ_2^*, and λ_3^* representing the regularization parameters corresponding to the minimum total errors for the first local machine in DKRR, DKRR, and KRR for all data, respectively, resulting in the derived global estimator under-fitting. This is not surprising, as the weighted average in the definition of (1) helps to reduce the variance but has little influence on the bias, just as Figure 4 purports to show.



(a) Bias

(b) Variance

Figure 4: Comparisons of bias and variance with different regularization parameter values. The data and simulation settings are the same as in Figure 3.

Therefore, a smaller regularization parameter than the theoretically optimal one is required for each local machine based on its own data, leading to over-fitting for each local estimator. The weighted average in (1) then succeeds in reducing the variance of DKRR and avoids over-fitting. The problem is, however, that each local machine only accesses its own data, making it difficult to determine the extent of over-fitting needed to optimize the performance of distributed learning. This refers to the over-fitting problem of parameter selection in distributed learning, and it is also the main challenge of our study.

Generally speaking, there are two ways to settle the over-fitting problem of parameter selection in distributed learning. One is to modify the existing parameter selection strategies, such as the cross-validation (Györfi et al., 2002; Caponnetto and Yao, 2010), balancing principle (De Vito et al., 2010; Lu et al., 2020), discrepancy principle (Raskutti et al., 2014; Celisse and Wahl, 2021), and Lepskii principle (Blanchard et al., 2019; Lin, 2024), to force the local estimator in each local machine to over-fit their own data. A typical example is the logarithmic mechanism (Liu et al., 2022), which uses $\lambda_j^{\log_{|D_j|}|D|}$ to reduce the regularization parameter λ_j selected by D_j. Recalling that it is unknown what the extent of over-fitting should be, it is difficult for this approach to get appropriate regularization parameters to achieve the theoretically optimal learning performance established in (Zhang et al., 2015; Lin et al., 2017) of DKRR. The other is to modify the learned functions for parameter selection in each local machine so that the existing strategies can directly find the optimal regularization parameter for distributed learning. We adopt the latter in this paper since it is feasible to modify the learned functions by designing delicate communication strategies. In particular, it is possible to find a good approximation of the global estimator $\overline{f}_{D,\vec{\lambda}}$ by communicating non-private information.

Our approach is motivated by four interesting observations. First, it can be seen in Figure 3 that the optimal regularization parameter for the local estimator f_{D_j,λ_j} is not the optimal one for the global estimator. If we find an approximation of the global estimator $\overline{f}_{D,\vec{\lambda}}$ and use this approximation instead of the local estimator f_{D_j,λ_j} as the target of parameter selection in the j-th local machine, then it is not difficult to determine a nearly optimal regularization parameter for the global estimator $\overline{f}_{D,\vec{\lambda}}$ through the existing parameter selection strategies. Second, due to data privacy, it is impossible to communicate the local estimator f_{D_j,λ_j} directly since such a communication requires not only the coefficients of linear combinations of shifts of kernels but also the centers of the kernel that should be inputs of data D_j. However, these local estimators can be well approximated by linear combinations of some fixed basis functions, which is a classical research topic in approximation theory (Narcowich and Ward, 2004; Wendland and Rieger, 2005; Narcowich et al., 2006). Third, the well-developed sampling approaches including Monte Carlo sampling and Quasi-Monte Carlo sampling (Dick and Pillichshammer, 2010; Leobacher and Pillichshammer, 2014) introduced several low-discrepancy sequences, such as Sobol sequences, Niederreiter sequences, and Halton sequences, to improve the efficiency of the above approximation. Based on this, each local machine can generate the same centers of the kernel to establish a set of fixed basis functions, thus realizing the communication of functions by transmitting coefficients. Finally, though data cannot be communicated, some other non-private information, such as predicted values of queries, gradients, and coefficients of some basis functions, is communicable in distributed learning (Li et al., 2014; Lee et al., 2017; Jordan et al., 2019). According to the above four important observations, we design the road map for parameter selection in distributed learning, as shown in Figure 2.

As stated above, there are five crucial ingredients in our approach: basis functions generation, local approximation, communications, global approximation, and local parameter selection. For the first issue, we focus on searching low-discrepancy sequences (Dick and Pillichshammer, 2010; Leobacher and Pillichshammer, 2014) to form the centers of the kernel and then obtain a linear space spanned by these basis functions. For the second issue, we use the radial basis function approximation approach (Narcowich and Ward, 2004; Narcowich et al., 2006; Rudi et al., 2015) with the noise-free data $\left\{\left(x_{i,j}, f_{D_j,\lambda_j}(x_{i,j})\right)\right\}$ to provide a local approximation of the local estimator. For the third issue, each local machine transmits the coefficients of its local approximation to the global machine without leaking sensitive information about its own data. For the fourth issue, the global machine synthesizes these coefficients by weighted average like (1) and transmits the synthesized coefficients back to all local machines. For the last issue, each local machine executes a specific parameter selection strategy to determine the regularization parameter of the global approximation. Note that besides the coefficients of some fixed basis functions, the sensitive information of the data in local machines is not communicated, which implies that the proposed approach provides a feasible scheme to settle the data silos.

2.2 Related work

Since data silos caused by data privacy and interoperability impede the effective integration and management of data, it is highly desirable to develop feasible machine learning schemes to settle them and sufficiently explore the value of big data. Federated learning (Li et al., 2020) is a popular approach to handling the data silos. It starts with a pre-training model that all data holders know and aims at collaborative training through multiple rounds of communications of non-sensitive information from the data holders to aggregate a golden model. Though it has been numerically verified that federated learning is excellent in some specific application areas (Tuor et al., 2021; Li et al., 2022; Wei et al., 2023), the exploration of pre-training models and multiple rounds of communications leads to essential weaknesses of the current defense against privacy attacks, such as data probing, data poisoning, model poisoning, and inference attacks (Li et al., 2020; Lyu et al., 2020). More importantly, the lack of solid theoretical verifications restricts the use of federated learning in high-risk areas such as natural disaster prediction, financial market prediction, medical diagnosis prediction, and crime prediction.

Theoretically, nonparametric distributed learning based on a divide-and-conquer strategy (Zhang et al., 2015; Zhou and Tang, 2020) is a more promising approach for addressing the data silos. As shown in Figure 1, it requires neither a pre-training model nor multiple rounds of communications. Furthermore, solid theoretical verification has been established for numerous distributed learning schemes, including DKRR (Zhang et al., 2015; Lin et al., 2017), distributed gradient-based approaches (Lin and Zhou, 2018; Lin and Cevher, 2018), and distributed spectral-based algorithms (Guo et al., 2017a; Mücke and Blanchard, 2018), in the sense that such a distributed learning scheme performs almost the same as running the corresponding algorithms on the whole data under the same conditions. These results seem to show that distributed learning can successfully address the data silos while realizing the benefits of big data without communicating sensitive information about the data. However, such exciting results are based on the assumption of the selection of optimal (hyper-)parameters for distributed learning, which is challenging in reality when the data cannot be shared. This is the main reason why nonparametric distributed learning has not been practically used for settling the data silos, though its design flow is suitable for this purpose.

As an open question in numerous papers (Zhang et al., 2015; Lin et al., 2017; Mücke and Blanchard, 2018; Zhao et al., 2019), parameter selection of distributed learning has already been noticed by (Xu et al., 2019) and (Liu et al., 2022). In particular, Xu et al. (2019) proposed a distributed generalized cross-validation (DGCV) for DKRR and provided some solid theoretical analysis. It should be noted that the proposed DGCV essentially requires the communication of data, making it suffer from the data silos. Liu et al. (2022) proposed a logarithmic mechanism to force the over-fitting of local estimators without communicating sensitive information about local data and theoretically analyzed the efficacy of the logarithmic mechanism. However, their theoretical results are based on the assumption that the optimal parameter is algebraic with respect to the data size, which is difficult to verify in practice.

Different from the aforementioned results, our main novelty is to propose an adaptive parameter selection strategy to equip non-parametric distributed learning schemes to settle the data silos. It should be highlighted that our proposed approach needs only two rounds of communications of non-sensitive information. We provide the optimality guarantee in theory and the feasibility evidence in applications.

Table 1 compares the related work based on key concerns in addressing data silos, covering the three stages of distributed learning: local processing, communication, and synthesization. From Table 1, we see that existing schemes primarily focus on communication efficiency and generalization performance after synthesization. However, addressing the data silo challenge requires a stronger emphasis on local machines, particularly in preserving local data privacy against data probing attacks and ensuring autonomy in local model design. Given these concerns, the proposed AdaDKRR is, to the best of our knowledge, novel in the realm of distributed learning.

Table 1: Comparison of different distributed learning schemes

Methods	**Concerns in addressing data silos**				
	Local processing stage		*Communication stage*		*Synthesization stage*
	Autonomous local model design	**Adaptive parameter tuning**	**Low communication cost**	**No local data leakage (resist data probing attacks)**	**Optimal generalization performance**
FL	×	×	×	✓	✓
DGA	✓	×	✓	×	✓
DSA	✓	×	✓	×	✓
DGCV	×	✓	✓	×	✓
DL with Logarithmic mechanism	✓	✓	✓	✓	×
AdaDKRR	✓	✓	✓	✓	✓

Note: "DGA" refers to distributed gradient descents (Lin and Zhou, 2018; Lin and Cevher, 2018), "DSA" to distributed spectral algorithms (Guo et al., 2017a; Mücke and Blanchard, 2018), and "DL with Logarithmic mechanism" to a distributed learning scheme using the logarithmic mechanism (Liu et al., 2022). All methods listed are from related work mentioned above.

3. Adaptive Distributed Kernel Ridge Regression

In this section, we propose the adaptive parameter selection strategy for distributed kernel ridge regression, which is named AdaDKRR, to address the data silos. As discussed in the previous section, our approach includes five important ingredients: basis generation, local approximation, communications, global approximation, and parameter selection. To ease the description, we use the "hold-out" approach (Caponnetto and Yao, 2010) in each local machine to adaptively select the parameter, though our approach can be easily designed for other strategies. The detailed implementation of AdaDKRR is shown in Algorithm 1.

Algorithm 1 AdaDKRR with hold-out

```
Input: Training data subset $D_j = \{(x_{ij}, y_{ij})\}_{i=1}^{|D_j|}$ with $x_{ij} \in \mathcal{X}$ and $|y_{ij}| \leq M$ stored in
    the $j$-th local machine for $j = 1, \cdots, m$, a candidate set of the regularization parameter
    $\Lambda = \{\lambda_\ell\}_{\ell=1}^{L}$, and a query point $x$. Divide $D_j = \{(x_{ij}, y_{ij})\}_{i=1}^{|D_j|}$ into training and
    validation sets, and denote them as $D_j^{tr}$ and $D_j^{val}$, respectively.
```

```
1: Local machines: given $\lambda_\ell \in \Lambda$ and $j$, run KRR with data $D_j^{tr}$ to obtain a local estimator
       $f_{D_j^{tr},\lambda_\ell} = \arg\min_{f\in\mathcal{H}_K} \left\{ \frac{1}{|D_j^{tr}|} \sum_{(x,y)\in D_j^{tr}} (f(x)-y)^2 + \lambda_\ell \|f\|_K^2 \right\}.$   (3)
                                                                                    ▷ Local Processing
2: Local machines: generate the same set of centers $\Xi_n = \{\xi_k\}_{k=1}^n$ and define a set of basis
   functions $B_{n,K} := \{\sum_{k=1}^n a_k K_{\xi_k} : a_k \in \mathbb{R}\}$ with $K_\xi(x) = K(\xi, x)$.      ▷ Basis Generation
3: Local machines: for some $s \in \mathbb{N}$, generate a set of points $\{x_{i,j}^*\}_{i=1}^s \subseteq \mathcal{X}$ and define an
   approximation of $f_{D_j^{tr},\lambda_\ell}$ by running KRR on data $\left\{\left(x_{i,j}^*, f_{D_j^{tr},\lambda_\ell}(x_{i,j}^*)\right)\right\}_{i=1}^s$, that is,
       $f_{D_j^{tr},\lambda_\ell,n,\mu,s}^{local} = \arg\min_{f\in B_{n,K}} \frac{1}{s} \sum_{i=1}^s \left(f(x_{i,j}^*) - f_{D_j^{tr},\lambda_\ell}(x_{i,j}^*)\right)^2 + \mu \|f\|_K^2$   (4)
   for some $\mu > 0$, and denote $f_{D_j^{tr},\lambda_\ell,n,\mu,s}^{local} = \sum_{k=1}^n a_{j,k,\ell}^{local} K_{\xi_k}$.      ▷ Local Approximation
4: Local machines: transmit the coefficient matrix $\left(a_{j,k,\ell}^{local}\right)_{k=1,\ell=1}^{n,L}$ to the global machine.
                                                                                    ▷ Communication(I)
5: Global machine: synthesize the coefficients by $a_{k,\ell}^{global} = \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} a_{j,k,\ell}^{local}$ and communicate $\left(a_{k,\ell}^{global}\right)_{k=1,\ell=1}^{n,L}$ to each local machine.      ▷ Synthesization and
   Communication(II)
6: Local machines: obtain a global approximation $f_{D^{tr},\lambda_\ell,n,\mu,s}^{global}$ as
       $f_{D^{tr},\lambda_\ell,n,\mu,s}^{global} := \sum_{k=1}^n a_{k,\ell}^{global} K_{\xi_k} = \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} \sum_{k=1}^n a_{j,k,\ell}^{local} K_{\xi_k} = \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} f_{D_j^{tr},\lambda_\ell,n,\mu,s}^{local}$   (5)
   and define
       $\lambda_j^* = \arg\min_{\lambda_\ell\in\Lambda} \frac{1}{|D_j^{val}|} \sum_{(x,y)\in D_j^{val}} \left(\pi_M f_{D^{tr},\lambda_\ell,n,\mu,s}^{global}(x) - y\right)^2$   (6)
   with $\pi_M f(x) = \mathrm{sign}(f(x)) \min\{|f(x)|, M\}$.                            ▷ Local Validation
7: Local machines: calculate $\pi_M f_{D^{tr},\lambda_j^*,n,\mu,s}^{global}(x)$ and communicate it to the global machine.
                                                                                    ▷ Communication(III)
8: Global machine: synthesize the AdaDKRR estimator as                          ▷ Global Estimator
       $\overline{f}_{D,\vec{\lambda}^*}^{Ada}(x) := \overline{f}_{D,\vec{\lambda}^*,n,\mu,s}^{Ada}(x) = \sum_{j=1}^m \frac{|D_j|}{|D|} \pi_M f_{D^{tr},\lambda_j^*,n,\mu,s}^{global}(x).$   (7)
Output: The global estimator $\overline{f}_{D,\vec{\lambda}^*}^{Ada}(x)$.
```

Compared with the classical DKRR (Zhang et al., 2015; Lin et al., 2017), AdaDKRR presented in Algorithm 1 requires five additional steps (Steps 2–6) that include basis generation, local approximation, global approximation, and two rounds of communications with $\mathcal{O}(mnL)$ communication complexity. Algorithm 1 actually presents a feasible framework for selecting parameters of distributed learning, as the basis functions, local approximation, and global approximation are not unique. It should be highlighted that Algorithm 1 uses the “hold-out” method in selecting the parameters, while our approach is also available for cross-validation (Györfi et al., 2002), which requires a random division of the training data D_j. We refer the readers to Algorithm 2 in the Appendix for the detailed training and testing flows of the cross-validation version of AdaDKRR.

In the basis generation step (Step 2), we generate the same set of basis functions in all local machines so that the local estimators defined in (3) can be well approximated by linear combinations of these basis functions. Noting that the local estimators are smooth and in $\mathcal{H}_K$, numerous basis functions, such as polynomials, splines, and kernels, can approximate them well from the viewpoint of approximation theory (Wendland and Rieger, 2005). Since we have already obtained a kernel K, we use the kernel to build up the basis functions, and then the problem boils down to selecting a suitable set of centers $\Xi_n := \{\xi_k\}_{k=1}^n$ so that $\mathrm{span}\{K_{\xi_k}\}$ can well approximate functions in $\mathcal{H}_K$. There are roughly two approaches to determining Ξ_n. One is to generate a set of fixed low-discrepancy sequences, such as Sobol sequences and Halton sequences (Dick and Pillichshammer, 2010), with the same size. The other method is to generate n points (in a random manner according to a uniform distribution) in the global machine, and then the global machine transmits this set of points to all local machines.

In this paper, we focus on the first method, though the second one is also feasible, for the following three reasons: (1) *Advantages of using low-discrepancy sequences in numerical integration approximation.* As discussed in (Asmussen and Glynn, 2007), the convergence rate of integration error using low-discrepancy sequences (with a convergence rate of $\mathcal{O}(n^{-1})$) is typically faster than that of uniform random sampling (with a convergence rate of $\mathcal{O}(n^{-\frac{1}{2}})$), which is crucial in reducing the communication cost in AdaDKRR. Furthermore, it can be found in (Dick and Pillichshammer, 2010; Dick, 2011; Feng et al., 2024) that there exist constants $c, \beta > 0$ such that

$$\sup_{\|f\|_K \leq 1, \|g\|_K \leq 1} \left| \int f(x)g(x)dP_u(x) - \frac{1}{n}\sum_{k=1}^{n} f(\xi_k)g(\xi_k) \right| \leq cn^{-\beta}, \tag{8}$$

where P_u denotes a uniform distribution. (2) *Reproducibility on local machines.* As discussed in (Leobacher and Pillichshammer, 2014), low-discrepancy sequences, such as Sobol and Halton sequences, can be independently regenerated on distributed local machines. Their construction depends primarily on the number of points, allowing them to define the same basis functions without requiring communication, thereby reducing communication costs further. (3) *Low computational cost.* The cost of generating Sobol and Halton sequences is $\mathcal{O}(n \log n)$ and $\mathcal{O}(n \log d)$, respectively, where d denotes the dimensionality. These costs are not much higher than that of uniform random sampling, which is $\mathcal{O}(n)$. A detailed explanation of the rationale behind using fixed low-discrepancy sequences in this paper is provided in Appendix B.

In the local approximation step (Step 3), we aim to find a good approximation of the local estimator $f_{D_j^{tr},\lambda_\ell}$. The key is to select a suitable set of points $\{x_{i,j}^*\}_{j=1}^s$ and a suitable parameter μ so that the solution to (4) can well approximate $f_{D_j^{tr},\lambda_\ell}$. Since there are already two point sets, $\{x_{i,j}\}_{i=1}^{|D_j^{tr}|}$ and $\{\xi_k\}_{k=1}^n$, we can select one of them as $\{x_{i,j}^*\}_{j=1}^s$. In this paper, we use the former, but choosing the latter is also reasonable because the solution to (4) is a good approximation of the local estimator (Wendland and

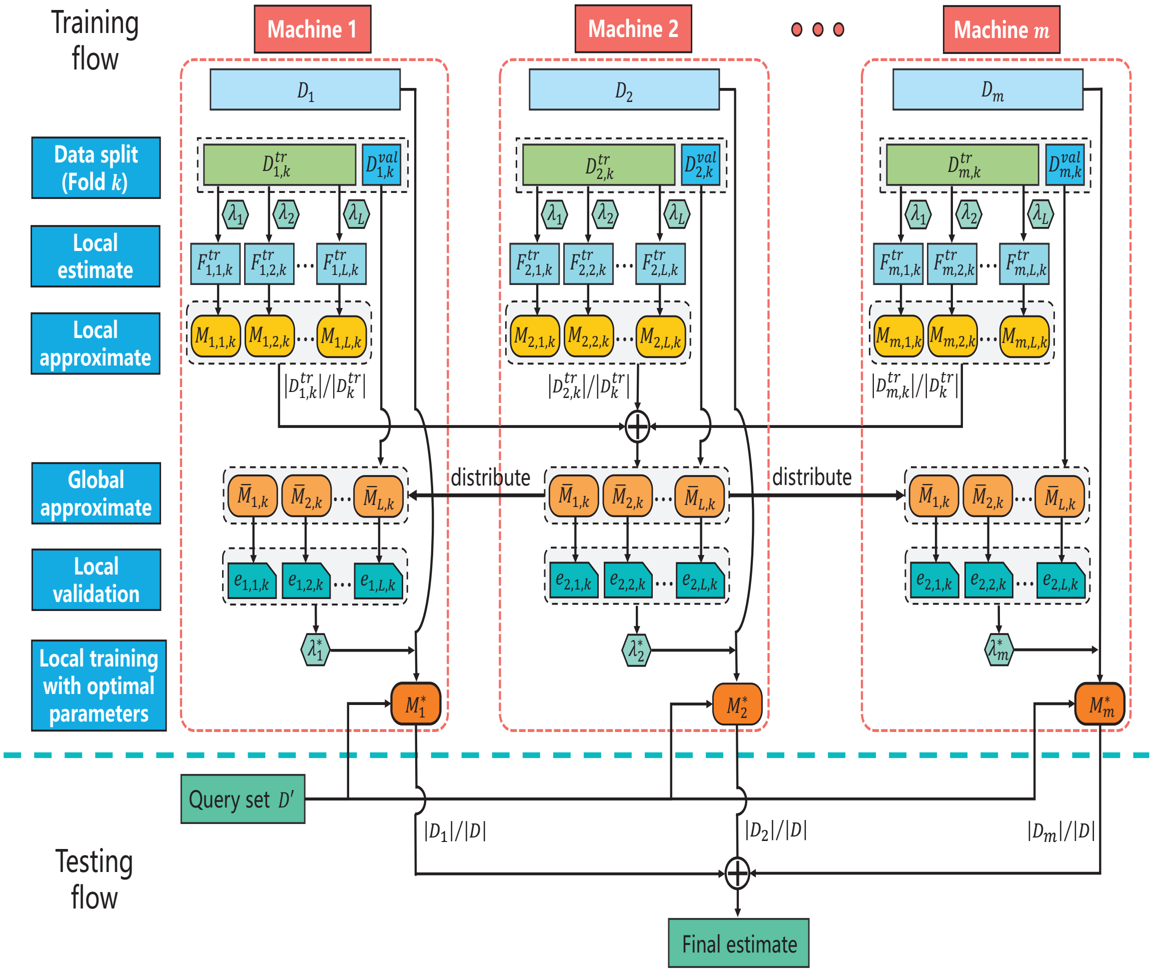


Figure 5: Training and testing flows of the proposed method

Rieger, 2005). Noting $s = |D_j^{tr}|$, we write $f^{local}_{D_j^{tr},\lambda_\ell,n,\mu,s}$ as $f^{local}_{D_j^{tr},\lambda_\ell,n,\mu}$. Recalling the idea of Nyström regularization (Rudi et al., 2015; Sun et al., 2022) and regarding (4) as a Nyström regularization scheme with the noise-free data $\left\{\left(x_{i,j}, f_{D_j^{tr},\lambda_\ell}(x_{i,j})\right)\right\}_{i=1}^{|D_j^{tr}|}$, we obtain from (4) that $f^{local}_{D_j^{tr},\lambda_\ell,n,\mu}(\cdot) = \sum_{k=1}^{n} a^{local}_{j,k,\ell} K_{\xi_k}(\cdot)$, where

$$\vec{a}^{local}_{j,\ell} := \left(a^{local}_{j,1,\ell}, \cdots, a^{local}_{j,n,\ell}\right)^T = \left(\mathbb{K}^T_{|D_j^{tr}|,n}\mathbb{K}_{|D_j^{tr}|,n} + \mu\left|D_j^{tr}\right|\mathbb{K}_{n,n}\right)^\dagger \mathbb{K}^T_{|D_j^{tr}|,n}\vec{f}_{D_j^{tr},\lambda_\ell}, \qquad (9)$$

$A^\dagger$ and A^T denote the Moore-Penrose pseudo-inverse and transpose of a matrix A, respectively, $\left(\mathbb{K}_{|D_j^{tr}|,n}\right)_{i,k} = K(x_{i,j},\xi_k)$, $(\mathbb{K}_{n,n})_{k,k'} = K(\xi_k,\xi_{k'})$, and $\vec{f}_{D_j^{tr},\lambda_\ell} = \Big(f_{D_j^{tr},\lambda_\ell}(x_{1,j}),\ldots,$ $f_{D_j^{tr},\lambda_\ell}(x_{|D_j^{tr}|,j})\Big)^T$. Therefore, it requires $\mathcal{O}\left(|D_j^{tr}|n^2 + n^3\right)$ floating computations to derive the local approximation. Since $\left\{\left(x_{i,j}, f_{D_j^{tr},\lambda_\ell}(x_{i,j})\right)\right\}_{i=1}^{|D_j^{tr}|}$ is noise-free, the parameter μ in

(4) is introduced to overcome the ill-conditionedness of the linear least squares problems and thus can be set to be small, for which $8\max\{(\kappa^2+1)/3, 2\sqrt{\kappa^2+1}\}\max_{j=1,\dots,m}\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|}$ is a suitable candidate for μ, where $\kappa := \sqrt{\sup_{x\in\mathcal{X}} K(x,x)}$. If $\mathcal{X}$ is known to possess some specific structure such as admitting a star-shaped structure (Wendland and Rieger, 2005), then the verifications of the effectiveness of local approximation can be derived from (Wendland and Rieger, 2005, Proposition 3.6) directly, in which μ can be set to be extremely small and 10^{-4} is a good candidate.

In the global approximation step (Step 6), the global approximation is obtained through a weighted average. The optimal parameters of local machines are then searched for the global approximation via the validation set. If $f^{local}_{D_j^{tr},\lambda_\ell,n,\mu}$ is a good approximation of $f_{D_j^{tr},\lambda_\ell}$, then $f^{global}_{D^{tr},\lambda_\ell,n,\mu}$ is a good approximation of the global estimator $\overline{f}_{D^{tr},\lambda_\ell}$ defined by (1). Therefore, the optimal parameters selected for the global approximation are close to those of the global estimator. It should be noted that introducing the truncation operator π_M in parameter selection is to ease the theoretical analysis and does not require additional computation. It requires $\mathcal{O}\left(|D_j^{val}|nL\right)$ floating computations in this step. The flows of AdaDKRR adopted in this paper can be found in Figure 5.

4. Theoretical Verifications

In this section, we study the generalization performance of AdaDKRR defined by (7) in a framework of statistical learning theory (Cucker and Zhou, 2007; Steinwart and Christmann, 2008), where samples in D_j for $j=1,2,\dots,m$ are assumed to be independently and identically drawn according to an unknown joint distribution $\rho := \rho(x,y) = \rho_X(x)\rho(y|x)$ with the marginal distribution ρ_X and the conditional distribution $\rho(\cdot|x)$. The regression function $f_\rho(x) = E[y|X=x]$ minimizes the generalization error $\mathcal{E}(f) := \int_{\mathcal{Z}}(f(x)-y)^2 d\rho$ for $f \in L^2_{\rho_X}$, where $L^2_{\rho_X}$ denotes the Hilbert space of ρ_X-square integrable functions on $\mathcal{X}$, with the norm denoted by $\|\cdot\|_\rho$. Therefore, the purpose of learning is to obtain an estimator f_D based on D_j for $j=1,2,\dots,m$ to approximate the regression function f_ρ without leaking the privacy information of D_j. In this way, the performance of the global estimator $\overline{f}^{Ada}_{D,\vec{\lambda}^*}$ is quantitatively measured by the excess generalization error

$$\mathcal{E}(\overline{f}^{Ada}_{D,\vec{\lambda}^*}) - \mathcal{E}(f_\rho) = \left\|\overline{f}^{Ada}_{D,\vec{\lambda}^*} - f_\rho\right\|_\rho^2, \tag{10}$$

which describes the relationship between the prediction error and the data size.

4.1 Generalization error for DKRR

Before presenting the generalization error analysis for AdaDKRR, we study theoretical behaviors of DKRR, which have been conducted in (Zhang et al., 2015; Lin et al., 2017) to show that DKRR performs similarly to running KRR on the whole data stored in a large enough machine, provided that m is not excessively large, $|D_1| \sim \cdots \sim |D_j|$, and the regularization parameters $\lambda_1 \sim \cdots \sim \lambda_m$ are similar to the optimal regularization parameter of KRR with the whole data.

The restriction on the number of local machines is natural since it is impossible to derive satisfactory generalization error bounds for DKRR when $m = |D|$, i.e., there is only one sample in each local machine. However, the assumptions $|D_1| \sim \cdots \sim |D_j|$ and $\lambda_1 \sim \cdots \sim \lambda_m$ are somewhat unrealistic. On one hand, local agents attending to the distributed learning system frequently have different data sizes, making it unrealistic to assume that the data sizes of local machines are the same. On the other hand, it is difficult to develop a parameter selection strategy for local machines so that $\lambda_1 \sim \cdots \sim \lambda_m$ are similar to the optimal regularization parameter of KRR with the whole data, as local agents only access their own data.

Noticing these, we derive optimal generalization error bounds for DKRR without the assumptions $|D_1| \sim \cdots \sim |D_j|$ and $\lambda_1 \sim \cdots \sim \lambda_m$ the same as the theoretically optimal one. For this purpose, we introduce several standard assumptions on the data D_j, regression function f_ρ, and kernel K. Our first assumption is the boundedness assumption of the output.

Assumption 1 *There exists a $M > 0$ such that $|y| \leq M$ almost surely.*

Assumption 1 is quite mild since we are always faced with finitely many data, whose outputs are naturally bounded. It should be mentioned that Assumption 1 implies $\|f_\rho\|_{L^\infty} \leq M$ directly. To present the second assumption, we introduce the integral operator L_K on $\mathcal{H}_K$ (or $L^2_{\rho_X}$) given by (Smale and Zhou, 2005)

$$L_K f := \int_{\mathcal{X}} K_x f(x) d\rho_X, \qquad f \in \mathcal{H}_K \quad (\text{or } f \in L^2_{\rho_X}).$$

The following assumption shows the regularity of the regression function f_ρ.

Assumption 2 *For some $r > 0$, assume*

$$f_\rho = L_K^r h_\rho, \quad \text{for some } h_\rho \in L^2_{\rho_X}, \tag{11}$$

where L_K^r denotes the r-th power of $L_K : L^2_{\rho_X} \to L^2_{\rho_X}$ as a compact and positive operator.

According to the no-free-lunch theory (Györfi et al., 2002, Chap.3), it is impossible to derive a satisfactory rate for the excess generalization error if there is no restriction on the regression functions. Assumption 2 actually connects f_ρ with the adopted kernel K, where the index r in (11) quantifies the relationship. Indeed, (11) with $r = 1/2$ implies $f_\rho \in \mathcal{H}_K$, $0 < r < 1/2$ implies $f_\rho \notin \mathcal{H}_K$, and $r > 1/2$ implies that f_ρ is in an RKHS generated by a smoother kernel than K.

Our third assumption is on the property of the kernel, measured by the effective dimension (Caponnetto and De Vito, 2007),

$$\mathcal{N}(\lambda) = \text{Tr}((\lambda I + L_K)^{-1} L_K), \qquad \lambda > 0.$$

Assumption 3 *There exists some $s \in (0, 1]$ such that*

$$\mathcal{N}(\lambda) \leq C_0 \lambda^{-s}, \tag{12}$$

where $C_0 \geq 1$ is a constant independent of λ.

It is obvious that (12) always holds with $s = 0$ and $C_0 = \kappa$. As discussed in (Fischer and Steinwart, 2020), Assumption 3 is equivalent to the eigenvalue decay assumption employed in (Caponnetto and De Vito, 2007; Steinwart et al., 2009; Zhang et al., 2015). It quantifies the smoothness of the kernel and the structure of the marginal distribution ρ_X. For example, if ρ_X is the uniform distribution on the unit cube in the d-dimensional space $\mathbb{R}^d$ (i.e., $\mathcal{X} = \mathbb{I}^d$), and K is a Sobolev kernel of order $\tau > d/2$, then Assumption 3 holds with $s = \frac{d}{2\tau}$ (Steinwart et al., 2009). The above three assumptions have been widely used to analyze generalization errors for kernel-based learning algorithms (Blanchard and Krämer, 2016; Chang et al., 2017b; Dicker et al., 2017; Guo et al., 2017a,b; Lin et al., 2017; Lin and Zhou, 2018; Mücke and Blanchard, 2018; Shi, 2019; Fischer and Steinwart, 2020; Lin et al., 2020; Sun et al., 2022), and optimal rates of excess generalization error for numerous learning algorithms have been established under these assumptions.

Under these well-developed assumptions, we provide the following theorem that DKRR can achieve the optimal rate of excess generalization error established for KRR with the whole data (Caponnetto and De Vito, 2007; Lin et al., 2017; Fischer and Steinwart, 2020), even when different local machines possess different data sizes.

Theorem 1 *Under Assumption 1, Assumption 2 with $\frac{1}{2} \leq r \leq 1$, and Assumption 3 with $0 < s \leq 1$, if*

$$\lambda_j = C_1 \left\{ \begin{array}{cc} |D|^{-\frac{1}{2r+s}}, & \text{if} \quad |D_j| \geq |D|^{\frac{1}{2r+s}} \log^4 |D|, \\ |D_j|^{-1} \log^4 |D|, & \text{otherwise}, \end{array} \right. \tag{13}$$

and

$$m \leq |D|^{\frac{s}{2r+s}} (\log |D|)^{-8r}, \tag{14}$$

then

$$E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right] \leq C_2 |D|^{-\frac{2r}{2r+s}}, \tag{15}$$

where C_1 and C_2 are constants independent of $|D|$ or m.

Under Assumptions 1–3, it can be found in (Caponnetto and De Vito, 2007) that the derived learning rates in (15) are optimal in the sense that there is a regression function f_ρ^* satisfying the above three assumptions such that

$$E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho^*\|_\rho^2\right] \geq C_3 |D|^{-\frac{2r}{2r+s}}$$

for a constant C_3 depending only on r and s. Unlike the existing results on distributed learning (Zhang et al., 2015; Lin et al., 2017; Mücke and Blanchard, 2018; Lin et al., 2020) that imposed strict restrictions on the data sizes of local machines, i.e., $|D_1| \sim |D_2| \sim \cdots \sim |D_m|$, Theorem 1 removes this condition since it is difficult to guarantee the same data size for all participants in the distributed learning system. As a result, it requires completely different mechanisms (13) to select the regularization parameter and stricter restriction on the number of local machines (14). The main reason for the stricter restriction on m is that the distributed learning system accommodates local machines with little data, i.e., $|D_j| \leq |D|^{\frac{1}{2r+s}} \log^4 |D|$. If we impose a qualification requirement that each participant in the distributed learning system has at least $|D|^{\frac{1}{2r+s}} \log^4 |D|$ samples, then the restriction can be greatly relaxed, just as the following corollary shows.

Corollary 2 *Under Assumption 1, Assumption 2 with* $\frac{1}{2} \leq r \leq 1$*, and Assumption 3 with* $0 < s \leq 1$*, if* $|D_j| \geq |D|^{\frac{1}{2r+s}} \log^4 |D|$*,* $\lambda_j = C_1 |D|^{-\frac{1}{2r+s}}$ *for all* $j = 1, 2, \dots, m$*, and*

$$m \leq |D|^{\frac{2r+s-1}{2r+s}} \log^{-4} |D|, \tag{16}$$

then

$$E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right] \leq C_2 |D|^{-\frac{2r}{2r+s}}, \tag{17}$$

where C_1 *and* C_2 *are constants independent of* $|D|$ *or* m*.*

Theorem 1 and Corollary 2 provide a baseline for the analysis of AdaDKRR in terms that the generalization error of AdaDKRR should be similar to (15).

4.2 Learning performance of AdaDKRR

In this subsection, we study the theoretical behavior of AdaDKRR (7) by estimating its generalization error in the following theorem.

Theorem 3 *Under Assumption 1, Assumption 2 with* $1/2 \leq r \leq 1$*, and Assumption 3 with* $0 < s \leq 1$*, if* ρ_X *is a uniform distribution,* $|D_j| \geq (8C_1^*(\log(1+\kappa)+2))^2 |D|^{\frac{1}{2r+s}} \log^4 |D|$*,* Λ *contains a* $\bar{\lambda} \sim |D|^{-\frac{1}{2r+s}}$*, and*

$$m \leq \min\left\{|D|^{\frac{2r+s-1}{4r+2s}} \log^{-4} |D|, |D|^{\frac{s}{2r+s}} \log^{-1} L\right\}, \tag{18}$$

then for any $\mu \in \left[(8C_1^*(\log(1+\kappa)+2))^2 \max_{j=1,\dots,m} \frac{\log^4 |D_j^{tr}|}{|D_j^{tr}|}, |D|^{-\frac{1}{2r+s}}\right]$ *and* Ξ_n *satisfying* (8) *for some* $c, \beta > 0$ *with* $\mu n^\beta \geq 2c$*, there holds*

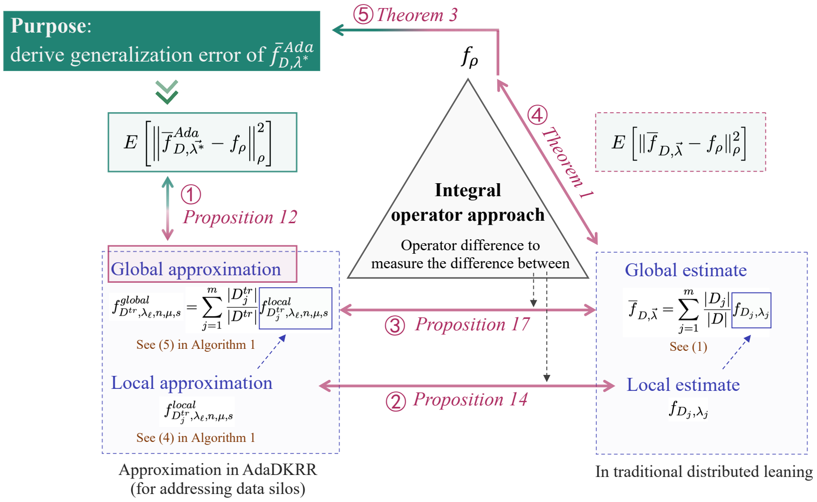
$$E\left[\left\|\overline{f}_{D,\vec{\lambda}^*}^{Ada} - f_\rho\right\|_\rho^2\right] \leq C|D|^{-\frac{2r}{2r+s}}, \tag{19}$$

where C, C_1^* *are constants depending only on* $\|h_\rho\|_\rho, M, r, C_0, c,$ *and* β*.*

Compared with Theorem 1, it can be found that AdaDKRR possesses the same generalization error bounds under some additional restrictions, implying that the proposed parameter selection strategy is optimal in the sense that no other strategies always perform better. There are five additional restrictions that may prohibit the wide use of the proposed approach: (I) m satisfies (18); (II) $|D_j| \geq (8C_1^*(\log(1+\kappa)+2))^2 |D|^{\frac{1}{2r+s}} \log^4 |D|$; (III) Λ contains a $\bar{\lambda} \sim |D|^{-\frac{1}{2r+s}}$; (IV) ρ_X is a uniform distribution; (V) $\mu \in \Big((8C_1^*(\log(1+\kappa)+2))^2 \max_{j=1,\dots,m} (\log^4 |D_j^{tr}|)/|D_j^{tr}|, |D|^{-\frac{1}{2r+s}}\Big)$ and Ξ_n satisfies (8) for some $c, \beta > 0$ with n satisfying $\mu n^\beta \geq 2c$.

Condition (I) is necessary since it is impossible to derive a satisfactory distributed learning estimator when each local machine has only one sample. Condition (II) presents a qualification requirement for the local machines participating in the distributed learning system, indicating that their data sizes should not be so small. Condition (III) means that the candidate set Λ should include the optimal parameter. Noting (18), the restriction on m is logarithmic with respect to L, and we can set $\Lambda = \{\lambda_k\}_{k=1}^L$ with $\lambda_k = q^k$ for some $q \in (0,1)$ and $L \sim |D|$. Conditions (IV) and (V) are mainly due to setting $\{x_{i,j}^*\}_{i=1}^s$ to $\{x_{i,j}\}_{i=1}^{|D_j|}$ in the local approximation step (Step 3 in Algorithm 1). Therefore, we have to use the quadrature property (8) of the low-discrepancy property, which requires the samples to be drawn i.i.d. according to the uniform distribution. Furthermore, the well-conditionedness of the local approximation imposes a lower bound of μ. Since $|D_j| \geq (8C_1^*(\log(1+\kappa)+2))^2|D|^{\frac{1}{2r+s}}\log^4|D|$, it is easy to check that $(8C_1^*(\log(1+\kappa)+2))^2\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|} \leq |D|^{-\frac{1}{2r+s}}$, and there are numerous feasible values for μ. The restriction on μ is to theoretically verify the well-conditionedness of the local approximation in the worst case. In practice, it can be set to $8\max\{(\kappa^2+1)/3, 2\sqrt{\kappa^2+1}\}\max_{j=1,\dots,m}\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|}$ directly without compromising the adaptive nature of AdaDKRR. It would also be interesting to set a suitable $\{x_{i,j}^*\}_{i=1}^s$ to remove or relax conditions (IV) and (V).

As shown in Theorem 3, under some assumptions, we prove that AdaDKRR performs similarly to running the optimal learning algorithms on the whole data $D = \cup_{j=1}^m D_j$ without considering data privacy. Recalling in Algorithm 1 that AdaDKRR only requires communicating non-sensitive information, it is thus a feasible strategy to address the data silos. We conclude this section with three important remarks on the proof ideas, uniform distribution assumptions, and data privacy.



Purpose:
derive generalization error of $\bar{f}_{D,\lambda^*}^{Ada}$
⑤ Theorem 3
f_ρ
$E\left[\left\|\bar{f}_{D,\vec{\lambda}^*}^{Ada} - f_\rho\right\|_\rho^2\right]$
① Proposition 12
④ Theorem 1
$E\left[\|\bar{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right]$
Integral operator approach
Operator difference to measure the difference between
Global approximation
$f_{D^{tr},\lambda_\ell,n,\mu,s}^{global} = \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} f_{D_j^{tr},\lambda_\ell,n,\mu,s}^{local}$
See (5) in Algorithm 1
③ Proposition 17
Global estimate
$\bar{f}_{D,\vec{\lambda}} = \sum_{j=1}^m \frac{|D_j|}{|D|} f_{D_j,\lambda_j}$
See (1)
Local approximation
$f_{D_j^{tr},\lambda_\ell,n,\mu,s}^{local}$
See (4) in Algorithm 1
② Proposition 14
Local estimate
f_{D_j,λ_j}
Approximation in AdaDKRR (for addressing data silos)
In traditional distributed leaning

Figure 6: The illustration of the proof ideas

Remark 4 *The proof of Theorem 3 is postponed to Appendix C. We sketch the proof ideas in this remark. The main tool of our analysis is the recently developed integral operator theory (Lin et al., 2017, 2021) and there are mainly four steps to prove the theorem. First, we build a bridge between the generalization error of $\bar{f}_{D,\vec{\lambda}^*}^{Ada}$ and the global approximation defined by (5) in Proposition 12 below by using perturbation inequalities established in (Györfi et al., 2002, Chap.7) and (Caponnetto and Yao, 2010). Then, an integral operator approach based on operator projection was developed to derive the difference between the local approximation*

$f^{loc}_{D,\lambda,\mu}$ and the local estimate $f_{D,\lambda}$ in Proposition 14 below. Here, we borrow the ideas from (Rudi et al., 2015) and (Sun et al., 2022) to divide the above difference into approximation error and computation error and use operator differences to quantify these errors. Third, we measure the difference between the global approximation (5) *and the global estimate in Theorem 1 by operator differences in Proposition 17. Finally, after deriving tight bounds for the mentioned operator differences, we derive Theorem 3 by combining the above three steps and Theorem 1. Figure 6 provides a clearer illustration of our proof ideas.*

Remark 5 *Besides the popular Assumptions 1-3, a technical assumption that ρ_X is the uniform distribution is imposed in Theorem 3. This is due to the fact that the marginal distribution ρ_X is unknown in practice, making it quite difficult to generate low-discrepancy sequences according to ρ_X directly. Since the popular Sobol sequences or Halton sequences, which have been proven to be quasi-uniform in the Lebesgue measure, are suggested in the approaches, we have to impose the uniformness assumption on ρ_X in our theoretical analysis. In practice, it is possible to first estimate the marginal distribution ρ_X and then generate some sequences accordingly. Because the main purpose of this paper is the parameter selection approach for DKRR rather than the sampling mechanism, we leave this interesting topic for future work.*

Remark 6 *This study aims to develop distributed methods that settle data silos by ensuring privacy preservation while maintaining optimal generalization performance. However, it is unrealistic to discuss privacy preservation without figuring out the privacy attack. In fact, data probing attack that aims to access the data from other organizations is one of the most important issues in distributed learning. As shown in Algorithm 1 and Theorem 3, AdaDKRR succeeds in defending against the data probing attack without sacrificing the utility of data measured by the generalization error. Table 2 presents the privacy issues concerning AdaDKRR and data silos.*

Table 2: Privacy issues for AdaDKRR

Attack	Privacy measure	Approach	Utility
Data probing	Local data are not leaked	AdaDKRR	Optimal generalization

5. Experimental Results

In this section, we use the following parameter selection methods for distributed learning to conduct experiments on synthetic and real-world data sets:

1) On each local machine, the parameters are selected by cross-validation, and DKRR is executed with these selected parameters; we call this method DKRR with cross-validation (“DKRR” for short).

2) On the j-th local machine, we first select parameters by cross-validation, and then transform the selected regularization parameter λ_j by $\lambda_j \leftarrow \lambda_j^{\log|D|/\log|D_j|}$, and transform the selected kernel width σ_j by $\sigma_j \leftarrow \sigma_j^{\log|D|/\log|D_j|}$ if the Gaussian kernel is used;

DKRR is executed with these transformed parameters; we call this method DKRR with cross-validation and logarithmic transformation ("DKRRLog" for short).

3) The proposed adaptive parameter selection method is applied to distributed learning and is denoted by "AdaDKRR".

All the experiments are run on a desktop workstation equipped with an Intel(R) Core(TM) i9-10980XE 3.00 GHz CPU, 128 GB of RAM, and Windows 10. The results are recorded by averaging the results from multiple individual trials with the best parameters.[1]

5.1 Synthetic results

In this part, the performance of the proposed method is verified by four simulations. The first one studies the influence of the number and type of center points for local approximation on generalization ability. The second one exhibits the robustness of AdaDKRR to the number of center points. The third simulation presents comparisons of the generalization ability of the three mentioned methods with changing the number of local machines, provided that all training samples are uniformly distributed to local machines. The last simulation focuses on comparisons of generalization ability for the three methods when the training samples are unevenly distributed on local machines.

Before carrying out experiments, we describe the generating process of the synthetic data and some important settings of the simulations. The inputs $\{x_i\}_{i=1}^N$ of training samples are independently drawn according to the uniform distribution on the (hyper-)cube $[0,1]^d$ with $d=3$ or $d=10$. The corresponding outputs $\{y_i\}_{i=1}^N$ are generated from the regression models $y_i = g_j(x_i)+\varepsilon$ with the Gaussian noise $\mathcal{N}(0,0.2)$ for $j=1,2$, where

$$g_1(x) = \begin{cases} (1-\|x\|_2)^6(35\|x\|_2^2+18\|x\|_2+3) & \text{if } 0<\|x\|_2\le 1, \\ 0 & \text{if } \|x\|_2>1, \end{cases} \tag{20}$$

for the 3-dimensional data, and

$$g_2(x) = (\|x\|_2-1)\,(\|x\|_2-2)\,(\|x\|_2-3) \tag{21}$$

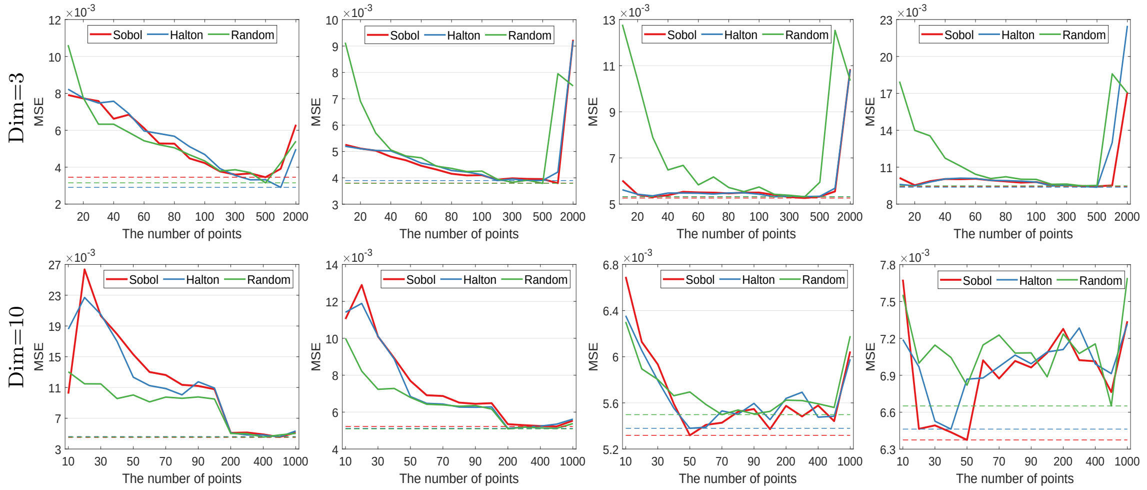
for the 10-dimensional data. The generation of test sets $\{(x_i',y_i')\}_{i=1}^{N'}$ is similar to that of training sets, but it has the promise of $y_i'=g_j(x_i')$.

For the 3-dimensional data, we use the kernel function $K_1(x_1,x_2)=h(\|x_1-x_2\|_2)$ with

$$h(r) = \begin{cases} (1-r)^4(4r+1) & \text{if } 0<r\le 1, \\ 0 & \text{if } r>1, \end{cases} \tag{22}$$

and the regularization parameter λ is chosen from the set $\{\frac{1}{2^q}|\frac{1}{2^q}\ge 10^{-10}, q=0,1,2,\cdots\}$. For the 10-dimensional data, we use the Gaussian kernel $K_2(x_1,x_2)=\exp\left(-\frac{\|x_1-x_2\|_2^2}{2\sigma^2}\right)$, the regularization parameter λ is chosen from the set $\{\frac{1}{3^q}|\frac{1}{3^q}\ge 10^{-10}, q=0,1,2,\cdots\}$, and the kernel width σ is chosen from 10 values that are drawn in a logarithmic, equally spaced interval $[0.1,10]$. In the simulations, we generate 10000 samples for training and 1000 samples for testing, and the regularization parameter μ for local approximation is fixed as 10^{-4}.

[1] The MATLAB code, as well as the data sets, can be downloaded from https://github.com/18357710774/AdaDKRR.

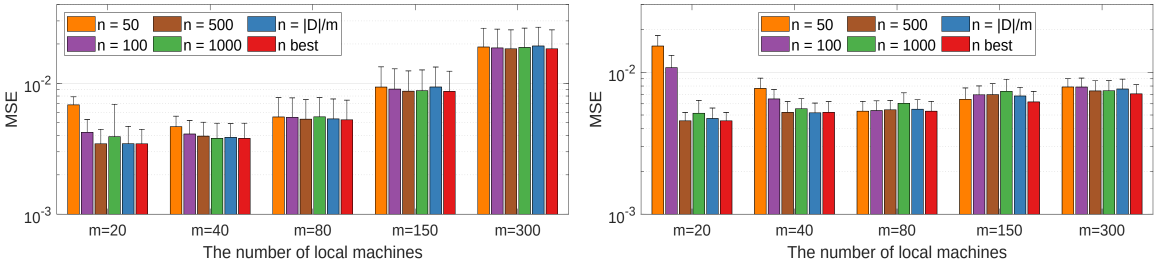


(a) $m = 20$ (b) $m = 40$ (c) $m = 80$ (d) $m = 160$

Figure 7: Relationship between test MSE and the number of center points in local approximation using the three low-discrepancy sequences for AdaDKRR with different numbers of local machines

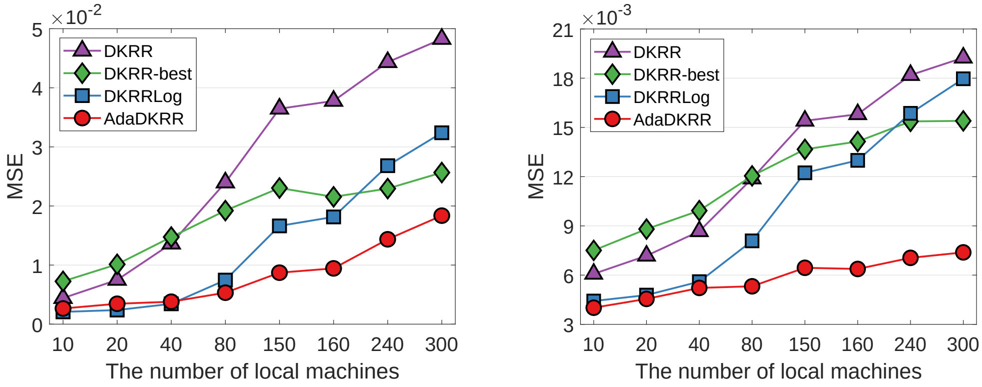
Simulation 1: In this simulation, we select three types of center points for local approximation, including two QMCS (Sobol points and Halton points) and one MCS (random points). The number m of local machines varies from the set $\{20, 40, 80, 160\}$. For each fixed m, the relation between the test MSE and the number n of center points is shown in Figure 7, in which the dashed lines exhibit the best test MSEs with the optimal numbers of the three types of center points. From the above results, we have the following observations: 1) As the number of center points increases, the curves of test MSE have a trend of descending first and then ascending. This is because very few center points cannot provide satisfactory accuracy for local approximation, resulting in the approximate function based on the center points having a large deviation from the ground truth, while a large number of center points put the estimator at risk of over-fitting. 2) The optimal number of center points generally decreases as the number of local machines increases. Because a smaller m indicates that there are more training samples on each local machine, more center points are required to cover these samples to obtain a satisfactory local approximation. 3) The three types of center points perform similarly on the 3-dimensional data, but Sobol points and Halton points are obviously better than random points on the 10-dimensional data, especially for larger numbers of local machines. In addition, the optimal number of random points is usually larger than the number of Sobol points and Halton points. The reason is that the discrepancy of QMCS is smaller than that of MCS, indicating that the sample distribution of QMCS is more uniform than that of MCS. Therefore, QMCS can better describe the structural information of the data and is more effective than MCS in local approximation. Since Sobol points perform similarly to Halton points, we take Sobol points as an example to demonstrate the superiority of the proposed method in the following experiments.

Simulation 2: In this simulation, we check the robustness of the proposed method concerning the number n of center points, as n determines the accuracy of the local ap-



(a) Dim=3 (b) Dim=10

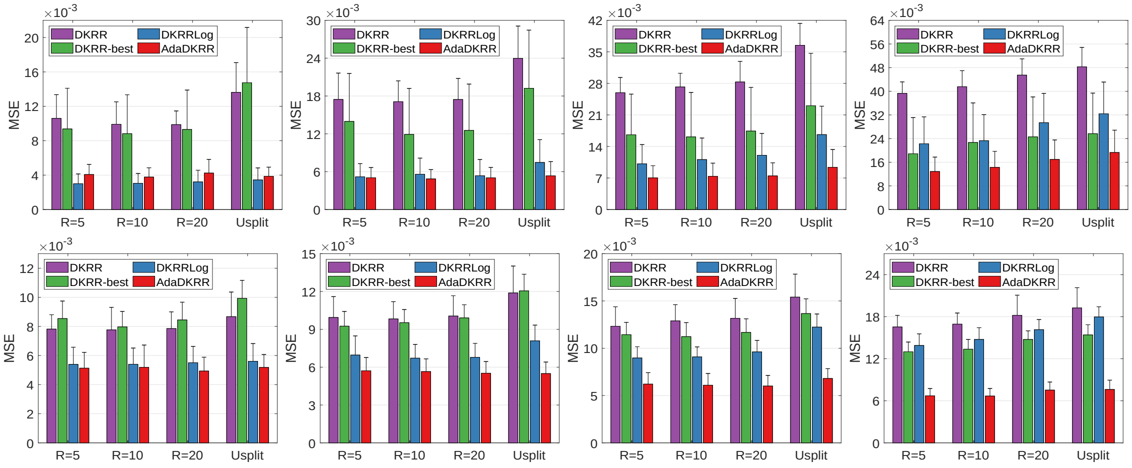
Figure 8: Relationship between test MSE and the number of Sobol points for AdaDKRR with different numbers of local machines



(a) Dim=3 (b) Dim=10

Figure 9: Comparisons of test MSE among the three parameter selection approaches with different numbers of local machines

proximation. We set the number n in two ways: 1) by fixing n as a constant (denoted by "$n = \#$"), and 2) by adaptively adjusting n as the average number of training samples in each local machine (denoted by "$n = |D|/m$"). We vary the number of Sobol points from the sets $\{10, 20, \cdots, 100, 200, \cdots, 500, 1000, 2000\}$ and $\{10, 20, \cdots, 100, 200, \cdots, 500, 1000\}$ for the 3-dimensional and 10-dimensional data, respectively, and vary the number of local machines from the set $\{20, 40, 80, 150, 300\}$. The testing RMSEs with respect to different orders of magnitude n under different numbers of local machines are shown in Figure 8, where "n best" represents the optimal MSE corresponding to the best n chosen from the candidate set and provides a baseline for assessing the performance of the proposed method. From the results, it can be seen that the generalization performances with different orders of n are all comparable with the best n when m is large (e.g., $m \geq 80$). Even when m is small, we can also obtain a satisfactory result by simply varying a few different orders of magnitude of n. In addition, for different numbers of local machines, the proposed method with an adaptive number of center points shows stable performance that is comparable to the best n. All these results demonstrate that the proposed method is robust to the number of center points.



(a) $m = 40$ (b) $m = 80$ (c) $m = 150$ (d) $m = 300$

Figure 10: Comparisons of test MSE among the three parameter selection approaches with different distribution patterns of data in local machines

Simulation 3: This simulation compares the proposed method with DKRR and DKRRLog under the condition that all training samples are uniformly distributed to local machines. The number of local machines changes from the set $\{10, 20, 40, 80, 150, 160, 240, 300\}$. For AdaDKRR, the number n of Sobol points is chosen from the set $\{50, 100, 500, 1000\}$. The results of test MSE as a function of the number of local machines are shown in Figure 9, where "DKRR-best" denotes the best performance of local machines in DKRR. Based on the above results, we have the following observations: 1) The test MSE grows as the number of local machines increases for all methods, but the growth of AdaDKRR is much slower than that of other methods. 2) When the number of local machines is small (e.g., $m \leq 40$), DKRR-best has the worst performance; the MSE values of DKRR are smaller than those of DKRR-best, which verifies that distributed learning can fuse the information of local machines and achieve better generalization performance than each local machine; AdaDKRR performs similarly to DKRRLog, and both of them are significantly better than DKRR, which provides evidence that it is not a good choice to select parameters only based on the data in each local machine. 3) When the number of local machines increases, the generalization performance of DKRR and DKRRLog deteriorates dramatically, even worse than that of a single local machine (e.g., $m \geq 200$), whereas the test MSE of AdaDKRR grows slowly and has obvious superiority to other methods. These results show that the proposed method is effective and stable in parameter selection for distributed learning.

Simulation 4: In this simulation, we compare the generalization performance of the three methods under non-uniform distributions of the number of training samples in local machines. Specifically, all training samples are randomly distributed to local machines, meaning the numbers of training samples in local machines are also random. In addition, we set the minimum number of training samples on each local machine so that cross-validation could be performed. For example, "$R = 5$" means that the minimum number of training samples on each local machine is no less than 5. AdaDKRR uses the adaptive number of

Sobol points (i.e., $n = |D|/m$) in local approximation for convenience. The number m of local machines varies from the set $\{40, 80, 150, 300\}$. For each fixed number m, we compare the generalization performance in four cases, including $R \in \{5, 10, 20\}$ and uniform split (denoted as "Usplit"), and the results are shown in Figure 10. Note that the case "Usplit" can be considered as "$R = |D|/m$". From the results, we can see that the performance differences among the three methods in each case of the non-uniform split are similar to those in the case of the uniform split, as described in Simulation 3. Additionally, the test MSE usually increases from the case "$R = 5$" to the case "Usplit". This is because a smaller value of R means that the distribution of the number of samples is more uneven, and the generalization performance of a single local machine with a large number of training samples is much better than that of a combination of several local machines with the same total number of training samples, as distributed learning shows. Compared with DKRR and DKRRLog, AdaDKRR is more robust to the split of training samples distributed to local machines, especially for the 10-dimensional data. The above results demonstrate that AdaDKRR is suitable for distributed learning with different data sizes of participants.

5.2 Real-world applications

The mentioned parameter selection methods are tested on two real-world data sets: used car price forecasting and graphics processing unit (GPU) performance prediction for single-precision general matrix multiplication (SGEMM). Before discussing the experiments, it is important to clarify some implementation details: 1) For each data set, half of the data samples are randomly chosen as training samples, and the other half are used as testing samples to evaluate the performance of the mentioned methods. Following the typical evaluation procedure, 10 independent sets of training and testing samples are generated by running 10 random divisions on the data set. 2) For each data set, min-max normalization is performed for each attribute except the target attribute. Specifically, the minimum and maximum values of the i-th attribute of training samples are calculated and denoted by $F_{min}^{(i)}$ and $F_{max}^{(i)}$, respectively. The i-th attribute of samples is rescaled using the formula $\hat{\mathbf{x}}_i = \left(\mathbf{x}_i - F_{min}^{(i)}\right) / \left(F_{max}^{(i)} - F_{min}^{(i)}\right)$, where $\mathbf{x}_i$ is the i-th attribute vector. 3) Based on the numerical results provided in Simulation 3 of the previous subsection, we vary the number of Sobol points in the set $\{10, 50, 100, 500\}$ and record the best result for AdaDKRR; the Gaussian kernel is used for the two data sets. 4) We consider the case that all training samples are uniformly distributed to local machines.

5.2.1 Used car price forecasting

As the number of private cars increases and the used car industry develops, more and more buyers are making used cars their primary choice due to their cost-effectiveness and practicality. Car buyers usually purchase used cars from private sellers and auctions aside from dealerships, and there is no manufacturer-suggested retail price for used cars. Car sellers want to reasonably evaluate the residual value of used cars to ensure sufficient profit margins. Car buyers hope the car they buy is economical, or at least they won't buy overpriced cars due to their unfamiliarity with the pricing of used cars. Therefore, pricing a used car can be regarded as a very important decision problem that is related to the

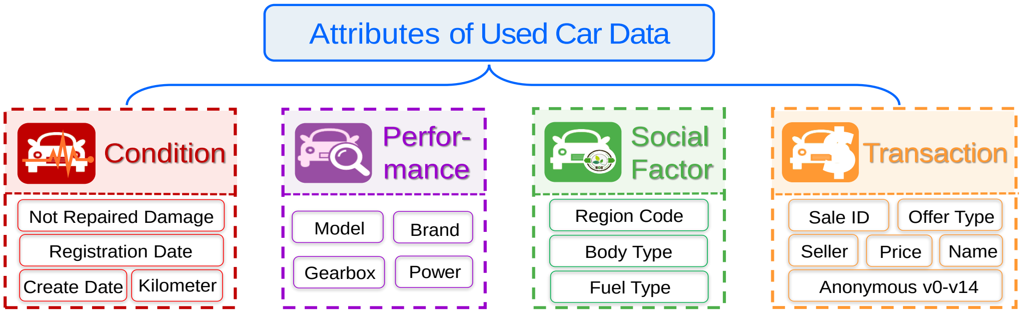


Figure 11: Details of the attributes of the data set `CarTianchi`

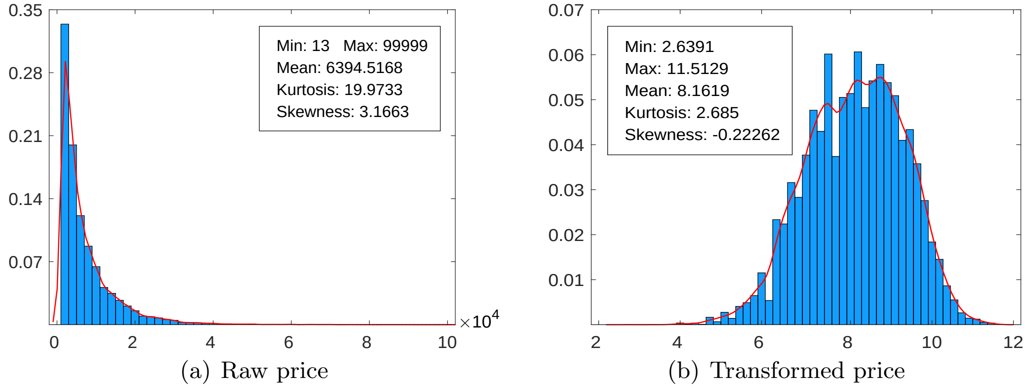
Table 3: The details of data binning on the data set `CarTianchi`

Attribute	Data binning
Usage time (day)	0: [0,90], 1: (90,180], 2: (180, 365], 3: (365,730], 4: (730,1095], 5: (1095, 1460], 6: (1460,2190], 7: (2190,3650], 8: (3650,5475], 9: (5475,+∞)
Power	0: [-19.3,1931.2], 1: (1931.2,3862.4], 2: (3862.4,5793.6], 3: (5793.6,7724.8], 4: (7724.8,9656.0], 5: (9656.0,11587.2], 6: (11587.2,13518.4], 7: (13518.4,15449.6], 8: (15449.6,17380.8], 9: (17380.8,19312]

success of the transaction between buyers and sellers. However, the sale price of used cars is very complicated, because it depends not only on the wear and tear of the car, such as usage time, mileage, and maintenance, but also on the performance of the car, such as brand, gearbox, and power, as well as on some social factors, such as car type, fuel type, and sale region. Sellers and buyers usually spend a lot of time and effort negotiating the price of used cars, so it is desirable to develop an effective pricing model from a collection of existing transaction data to provide a reliable reference for sellers and buyers and promote the success of transactions.

The used car data on Tianchi (`CarTianchi` for short) [2] provided by Alibaba Cloud comes from the used car transaction records of a trading platform, and its goal is to establish models to predict the price of used cars. The data set contains more than $400,000$ samples; each sample is described by 31 attributes, of which 15 are anonymous and 4 are masked to protect the confidentiality of the data. The attributes are described in detail in Figure 11, and they are grouped into four categories, including car condition, performance, social factor, and transaction. Note that the 15 anonymous attributes are classified into the category of transaction for convenience. In the experiment, we selected a subset of $129,710$ samples with price attributes after removing samples with missing values to train and evaluate models. We use the time difference between the sales date (i.e., the create date) and the registration date as an approximation of the usage time and remove the attributes of the sale ID, create date, and registration date. Data binning is applied to the attributes of usage time and power because their values are highly dispersed, and the details are listed in Table 3. The histogram of the target attribute of prices, as well as the skewness and kurtosis, are shown in Figure 12 (a), from which it can be seen that the data distribution does not obey the normal distribution, with a sharp peak and a long tail dragging on the right.

[2] https://tianchi.aliyun.com/competition/entrance/231784/information



(a) Raw price (b) Transformed price

Figure 12: The histogram of historical transaction prices

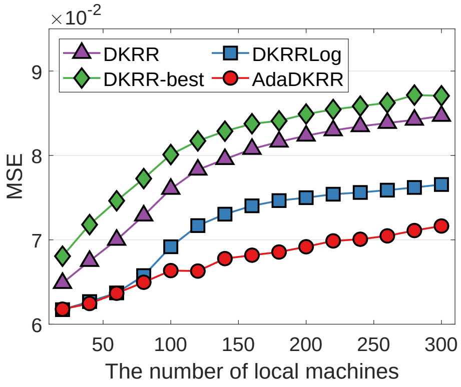


Figure 13: Comparisons of test MSE among the three parameter selection approaches with different numbers of local machines on the data set `CarTianchi`

Therefore, the target attribute of price is transformed by a logarithmic operation, and the histogram of the transformed price is close to a normal distribution, as shown in Figure 12 (b). The regularization parameter λ is chosen from the set $\{\frac{1}{3^q}|\frac{1}{3^q} \geq 10^{-10}, q = 0, 1, 2, \cdots\}$, and the kernel width σ is chosen from 10 values that are drawn in a logarithmic, equally spaced interval $[1, 10]$.

The relationship between test MSE and the number m of local machines for the compared methods is shown in Figure 13, where m varies from the set $\{20, 40, \cdots, 300\}$. From the results, we can see that DKRR-best has the worst generalization performance due to the limited number of training samples in local machines; DKRR synthesizes the estimators of local machines and thus achieves better performance than each local estimator; although the logarithmic transformation on the parameters makes DKRRLog superior to DKRR, it can still be significantly improved by AdaDKRR, especially for large numbers of local machines (e.g., $m \geq 100$). The above results demonstrate the effectiveness of the proposed parameter selection approach in distributed learning.

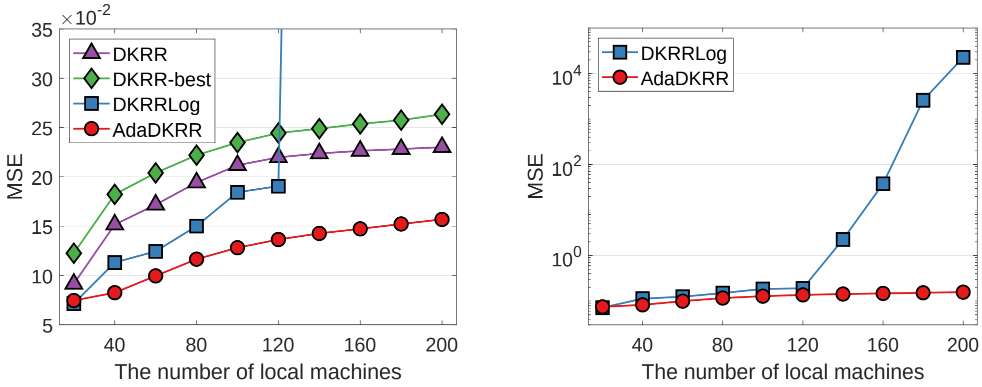


Figure 14: Comparisons of test MSE among the three parameter selection approaches with different numbers of local machines on the data set `SGEMM GPU`

5.2.2 SGEMM GPU performance prediction

Over the past decade, GPUs have delivered considerable performance in multi-disciplinary areas such as bioinformatics, astronomy, and machine learning. However, it is still a challenging task to achieve close-to-peak performance on GPUs, as professional programmers must carefully tune their code for various device-specific problems, each of which has its own optimal parameters such as workgroup size, vector data type, tile size, and loop unrolling factor. Therefore, it is important to design effective GPU acceleration models to automatically perform parameter tuning with the data collected from the device.

The data set `SGEMM GPU` (Nugteren and Codreanu, 2015) considers the running time of dense matrix-matrix multiplication $C = \alpha A^T B + \beta C$, as matrix multiplication is a fundamental building block in deep learning and other machine learning methods, where $A \in \mathbb{R}^{K\times M}$, $B \in \mathbb{R}^{K\times N}$, $C \in \mathbb{R}^{M\times N}$, $M = N = K = 2048$, α and β are constants, and A^T is the transpose of A. The data set contains $241,600$ samples; each sample includes a possible combination of 14 parameters of the SGEMM kernel and 4 running times for this parameter combination. The 14 parameters and their corresponding domains are as follows:

- Per-matrix 2D tiling at workgroup-level uses the parameters $M_{wg}, N_{wg} \in \{16, 32, 64, 128\}$, which correspond to the matrix dimensions of M and N, respectively.
- The inner dimension of 2D tiling at workgroup-level uses the parameter $K_{wg} \in \{16, 32\}$, which corresponds to the dimension of K.
- Local workgroup size uses the parameters $M_{dimC}, N_{dimC} \in \{8, 16, 32\}$.
- Local memory shape uses the parameters $M_{dimA}, N_{dimB} \in \{8, 16, 32\}$.
- The kernel loop unrolling factor is denoted by $K_{wi} \in \{2, 8\}$.
- Per-matrix vector widths for loading and storage use parameters $M_{vec}, N_{vec} \in \{1, 2, 4, 8\}$, where M_{vec} is for matrices A and C, and N_{vec} is for matrix B.
- The enabling stride for accessing off-chip memory within a single thread is denoted by $M_{stride}, N_{stride} \in \{0, 1\}$, where M_{stride} is for matrices A and C, and N_{stride} is for matrix B.
- Per-matrix manual caching of the 2D workgroup tile can be controlled by parameters $L\$_A, L\$_B \in \{0, 1\}$.

In the experiment, the first 14 columns of the data are used as data input, and the average time of the 4 runs is regarded as data output. Similar to the data set `CarTianchi`, the distribution of the average running time has a sharp peak and a long tail dragging on the right. Therefore, as suggested by Nugteren and Codreanu (2015), we also perform a logarithmic operation on the average running time. The regularization parameter λ is chosen from the set $\{\frac{1}{5^q}|\frac{1}{5^q} \geq 10^{-10}, q = 0, 1, 2, \cdots\}$, and the kernel width σ is chosen from 10 values that are drawn in a logarithmic, equally spaced interval $[1, 100]$.

Figure 14 records the relationship between test MSE and the number of local machines for the compared methods. It can be seen that the performance of these methods on the data set `SGEMM GPU` is similar to that on the data set `CarTianchi`. The only difference is that DKRRLog performs extremely poorly in generalization when the number of local machines is larger than 120. This is because the transformed parameters are far from the optimal parameters in some local machines. These results provide another piece of evidence that the proposed AdaDKRR is stable and effective in selecting parameters.

6. Conclusion

This paper proposed an adaptive parameter selection strategy for distributed learning to settle the data silos. Specifically, by communicating the coefficients of fixed basis functions, we obtained a good approximation of the global estimator and thus determined the algorithm parameters for the global approximation without leaking any sensitive information about the data. From a theoretical perspective, we established optimal rates of excess generalization error for the proposed method in a framework of statistical learning theory by utilizing the idea of low-discrepancy sequences and the classical radial basis function approximation. According to the theoretical findings, as long as the number of local machines is not too large, the proposed method is similar to running KRR on the whole data. The theoretical results demonstrate the efficacy of the proposed method for parameter selection in distributed learning. From an application point of view, we also applied the proposed method to several simulations and two real-world data sets, used car price forecasting and GPU performance prediction. The numerical results verify our theoretical assertions and demonstrate the feasibility and effectiveness of the proposed method in applications.

Acknowledgment

The work of Shao-Bo Lin, Xiaotong Liu is partially supported by the Major Key Project of PCL under Grant PCL2024A06 and the National Natural Science Foundation of China (Grant No. 2276209). The work of Di Wang is partially supported by the National Natural Science Foundation of China (Grant No. 12471486). The work of Hai Zhang is partially supported by the Major Key Project of PCL under Grant PCL2024A06, and the National Natural Science Foundation of China (Grant No. 12326615). The work of Ding-Xuan Zhou was partially supported by InnoHK initiative, the Government of the HKSAR, and Laboratory for AI-Powered Financial Technologies. The authors would like to thank the anonymous reviewers for their encouraging comments and valuable suggestions.

References

Søren Asmussen and Peter W Glynn. *Stochastic Simulation: Algorithms and Analysis*, volume 57. Springer, 2007.

Maria Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory*, pages 26–1. JMLR Workshop and Conference Proceedings, 2012.

Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer Science & Business Media, 2013.

Gilles Blanchard and Nicole Krämer. Convergence rates of kernel conjugate gradient for random design regression. *Analysis and Applications*, 14(06):763–794, 2016.

Gilles Blanchard, Peter Mathé, and Nicole Mücke. Lepskii principle in supervised learning. *arXiv preprint arXiv:1905.10764*, 2019.

Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

Andrea Caponnetto and Yuan Yao. Cross-validation based adaptation for regularization operators in learning theory. *Analysis and Applications*, 8(02):161–183, 2010.

Alain Celisse and Martin Wahl. Analyzing the discrepancy principle for kernelized spectral filter learning algorithms. *Journal of Machine Learning Research*, 22:76–1, 2021.

Xiangyu Chang, Shao-Bo Lin, and Yao Wang. Divide and conquer local average regression. *Electronic Journal of Statistics*, 11:1326–1350, 2017a.

Xiangyu Chang, Shao-Bo Lin, and Ding-Xuan Zhou. Distributed semi-supervised learning with kernel ridge regression. *Journal of Machine Learning Research*, 18(1):1493–1514, 2017b.

Felipe Cucker and Ding-Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*, volume 24. Cambridge University Press, 2007.

Thomas H. Davenport, Paul Barth, and Randy Bean. *How 'Big Data' Is Different.* MIT Sloan Management Review, 2012.

Ernesto De Vito, Sergei Pereverzyev, and Lorenzo Rosasco. Adaptive kernel methods using the balancing principle. *Foundations of Computational Mathematics*, 10(4):455–479, 2010.

Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(6):165–202, 2012.

Josef Dick. Higher order scrambled digital nets achieve the optimal rate of the root mean square error for smooth integrands. *The Annals of Statistics*, 39(3):1372–1398, 2011.

Josef Dick and Friedrich Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and quasi–Monte Carlo Integration.* Cambridge University Press, 2010.

Lee H Dicker, Dean P Foster, and Daniel Hsu. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electronic Journal of Statistics*, 11(1):1022–1047, 2017.

Jianqing Fan, Fang Han, and Han Liu. Challenges of big data analysis. *National Science Review*, 1(2):293–314, 2014.

Han Feng, Shao-Bo Lin, and Ding-Xuan Zhou. Radial basis function approximation with distributively stored data on spheres. *Constructive Approximation*, 60(1):1–31, 2024.

Simon Fischer and Ingo Steinwart. Sobolev norm learning rates for regularized least-squares algorithms. *Journal of Machine Learning Research*, 21(205):1–38, 2020.

Zheng-Chu Guo, Shao-Bo Lin, and Ding-Xuan Zhou. Learning theory of distributed spectral algorithms. *Inverse Problems*, 33(7):074009, 2017a.

Zheng-Chu Guo, Lei Shi, and Qiang Wu. Learning theory of distributed regression with bias corrected regularization kernel network. *Journal of Machine Learning Research*, 18 (1):4237–4261, 2017b.

Zheng-Chu Guo, Shao-Bo Lin, and Lei Shi. Distributed learning with multi-penalty regularization. *Applied and Computational Harmonic Analysis*, 46(3):478–499, 2019.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-free Theory of Nonparametric Regression*, volume 1. Springer, 2002.

Ting Hu and Renjie Guo. Distributed robust regression with correntropy losses and regularization kernel networks. *Analysis and Applications*, 22(04):689–725, 2024.

Priyank Jain, Manasi Gyanchandani, and Nilay Khare. Big data privacy: a technological perspective and review. *Journal of Big Data*, 3:1–25, 2016.

Michael I. Jordan, Jason D. Lee, and Yun Yang. Communication-efficient distributed statistical inference. *Journal of the American Statistical Association*, 114(526):668–681, 2019.

Mary R Lea and Kathy Nicoll. *Distributed Learning: Social and Cultural Approaches to Practice*. Routledge, 2013.

Jason D Lee, Qiang Liu, Yuekai Sun, and Jonathan E Taylor. Communication-efficient sparse regression. *Journal of Machine Learning Research*, 18(1):115–144, 2017.

Gunther Leobacher and Friedrich Pillichshammer. *Introduction to Quasi-Monte Carlo Integration and Applications*. Springer, 2014.

Mu Li, David G Andersen, Alexander J Smola, and Kai Yu. Communication efficient distributed machine learning with the parameter server. *Advances in Neural Information Processing Systems*, 27, 2014.

Qinbin Li, Yiqun Diao, Quan Chen, and Bingsheng He. Federated learning on non-iid data silos: An experimental study. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 965–978. IEEE, 2022.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3): 50–60, 2020.

Xiao-Bai Li and Jialun Qin. Anonymizing and sharing medical text records. *Information Systems Research*, 28(2):332–352, 2017.

Junhong Lin and Volkan Cevher. Optimal distributed learning with multi-pass stochastic gradient methods. In *International Conference on Machine Learning*, pages 3092–3101. PMLR, 2018.

Junhong Lin, Alessandro Rudi, Lorenzo Rosasco, and Volkan Cevher. Optimal rates for spectral algorithms with least-squares regression over hilbert spaces. *Applied and Computational Harmonic Analysis*, 48(3):868–890, 2020.

Shao-Bo Lin. Adaptive parameter selection for kernel ridge regression. *Applied and Computational Harmonic Analysis*, page 101671, 2024.

Shao-Bo Lin and Ding-Xuan Zhou. Distributed kernel-based gradient descent algorithms. *Constructive Approximation*, 47(2):249–276, 2018.

Shao-Bo Lin, Xin Guo, and Ding-Xuan Zhou. Distributed learning with regularized least squares. *Journal of Machine Learning Research*, 18(1):3202–3232, 2017.

Shao-Bo Lin, Yu Guang Wang, and Ding-Xuan Zhou. Distributed filtered hyperinterpolation for noisy data on the sphere. *SIAM Journal on Numerical Analysis*, 59(2):634–659, 2021.

Shao-Bo Lin, Di Wang, and Ding-Xuan Zhou. Sketching with spherical designs for noisy data fitting on spheres. *SIAM Journal on Scientific Computing*, 46(1):A313–A337, 2024.

Xiaotong Liu, Yao Wang, Shaojie Tang, and Shao-Bo Lin. Enabling collaborative diagnosis through novel distributed learning system with autonomy. *Available at SSRN 4128032*, 2022.

Shuai Lu, Peter Mathé, and Sergei V Pereverzev. Balancing principle in supervised learning for a general regularization scheme. *Applied and Computational Harmonic Analysis*, 48 (1):123–148, 2020.

Lingjuan Lyu, Han Yu, and Qiang Yang. Threats to federated learning: A survey. *arXiv preprint arXiv:2003.02133*, 2020.

Ryan Mcdonald, Mehryar Mohri, Nathan Silberman, Dan Walker, and Gideon Mann. Efficient large-scale distributed training of conditional maximum entropy models. *Advances in Neural Information Processing Systems*, 22, 2009.

Nicole Mücke and Gilles Blanchard. Parallelizing spectrally regularized kernel algorithms. *Journal of Machine Learning Research*, 19(1):1069–1097, 2018.

Nicole Mücke, Enrico Reiss, Jonas Rungenhagen, and Markus Klein. Data-splitting improves statistical performance in overparameterized regimes. In *International Conference on Artificial Intelligence and Statistics*, pages 10322–10350. PMLR, 2022.

Francis J Narcowich and Joseph D Ward. Scattered-data interpolation on $\mathbb{R}^n$: Error estimates for radial basis and band-limited functions. *SIAM Journal on Mathematical Analysis*, 36(1):284–300, 2004.

Francis J Narcowich, Joseph D Ward, and Holger Wendland. Sobolev error estimates and a bernstein inequality for scattered data interpolation via radial basis functions. *Constructive Approximation*, 24(2):175–186, 2006.

Mike Nguyen, Charly Kirst, and Nicole Mücke. Distributed SGD in overparametrized linear regression. *Analysis and Applications*, 22(03):425–466, 2024.

Cedric Nugteren and Valeriu Codreanu. Cltune: A generic auto-tuner for opencl kernels. In *2015 IEEE 9th International Symposium on Embedded Multicore/Many-core Systems-on-Chip*, pages 195–202, 2015. doi: 10.1109/MCSoC.2015.10.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Early stopping and non-parametric regression: An optimal data-dependent stopping rule. *Journal of Machine Learning Research*, 15(1):335–366, 2014.

Jonathan D Rosenblatt and Boaz Nadler. On the optimality of averaging in distributed statistical learning. *Information and Inference: A Journal of the IMA*, 5(4):379–404, 2016.

Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *NIPS*, pages 1657–1665, 2015.

Yue Sheng and Edgar Dobriban. One-shot distributed ridge regression in high dimensions. In *International Conference on Machine Learning*, pages 8763–8772. PMLR, 2020.

Lei Shi. Distributed learning with indefinite kernels. *Analysis and Applications*, 17(06): 947–975, 2019.

Steve Smale and Ding-Xuan Zhou. Shannon sampling ii: Connections to learning theory. *Applied and Computational Harmonic Analysis*, 19(3):285–302, 2005.

Steve Smale and Ding-Xuan Zhou. Learning theory estimates via integral operators and their approximations. *Constructive Approximation*, 26(2):153–172, 2007.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer Science & Business Media, 2008.

Ingo Steinwart, Don R Hush, Clint Scovel, et al. Optimal rates for regularized least squares regression. In *COLT*, pages 79–93, 2009.

Zirui Sun, Mingwei Dai, Yao Wang, and Shao-Bo Lin. Nystrom regularization for time series forecasting. *Journal of Machine Learning Research*, 23(312):1–42, 2022.

Prasanna Tambe. Big data investment, skills, and firm value. *Management Science*, 60(6): 1452–1469, 2014.

Tiffany Tuor, Joshua Lockhart, and Daniele Magazzeni. Asynchronous collaborative learning across data silos. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–8, 2021.

Le-Yin Wei, Zhan Yu, and Ding-Xuan Zhou. Federated learning for minimizing nonsmooth convex loss functions. *Mathematical Foundations of Computing*, 6(4):753–770, 2023.

Holger Wendland and Christian Rieger. Approximate interpolation with applications to selecting smoothing parameters. *Numerische Mathematik*, 101(4):729–748, 2005.

Ganggang Xu, Zuofeng Shang, and Guang Cheng. Distributed generalized cross-validation for divide-and-conquer kernel ridge regression and its asymptotic optimality. *Journal of Computational and Graphical Statistics*, 28(4):891–908, 2019.

Yuchen Zhang, John C Duchi, and Martin J Wainwright. Communication-efficient algorithms for statistical optimization. *Journal of Machine Learning Research*, 14(1):3321–3363, 2013.

Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Weihua Zhao, Fode Zhang, and Heng Lian. Debiasing and distributed estimation for high-dimensional quantile regression. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2569–2577, 2019.

Yaqin Zhou and Shaojie Tang. Differentially private distributed learning. *INFORMS Journal on Computing*, 32(3):779–789, 2020.

Appendix A: Training and Testing Flows of AdaDKRR

In this part, we give a detailed implementation of AdaDKRR (with cross-validation) as shown in Algorithm 2, where steps 1–19 are the training process and steps 20–24 are the testing process for a query set. In Algorithm 2, $\mathbb{K}_{|D_j|,n} \in \mathbb{R}^{|D_j|\times n}$ is a kernel matrix with the element in the i-th row and the k-th column being $K(x_{ij},\xi_k)$, $\mathbb{K}_{n,n}\in\mathbb{R}^{n\times n}$ is a kernel matrix with the element in the k-th row and the k'-th column being $K(\xi_k,\xi'_k)$, and $\mathbb{I}_D$ is the identity matrix with size $|D|$. $\vec{y}_D$ is a vector composed of the outputs of data D. In Step 8, $\vec{a}^{local}_{j,l,\ell} := \left(a^{local}_{j,l,1,\ell},\cdots,a^{local}_{j,l,n,\ell}\right)^T = \left(\mathbb{K}^T_{|D^{tr}_{j,l}|,n}\mathbb{K}_{|D^{tr}_{j,l}|,n} + \mu\left|D^{tr}_{j,l}\right|\mathbb{K}_{n,n}\right)^{\dagger}\mathbb{K}^T_{|D^{tr}_{j,l}|,n}\vec{f}^{tr}_{j,l,\ell}$, where $\vec{f}^{tr}_{j,l,\ell}$ is a vector composed of the outputs of data $F^{tr}_{j,l,\ell}$. In Step 11, $\left|D^{tr}_l\right| = \sum_{j=1}^m \left|D^{tr}_{j,l}\right|$. In Step 13, $\vec{a}^{global}_{l,\ell} := \left(a^{global}_{l,1,\ell},\cdots,a^{global}_{l,n,\ell}\right)^T$. In Step 18, $\vec{\alpha}^*_{D_j} := \left(\alpha^*_{1j},\cdots,\alpha^*_{|D_j|j}\right)^T = \left(\mathbb{K}_{D_j,D_j}+\lambda^*_j|D_j|\mathbb{I}_{D_j}\right)^{-1}\vec{y}_{D_j}$. Note that we retrain the local estimator f_{D_j,λ^*_j} with all training samples D_j under the selected optimal regularization parameter λ^*_j in the j-th local machine for $j=1,\cdots,m$ to improve the generalization performance of AdaDKRR.

Algorithm 2 Training and testing flows of AdaDKRR (with cross-validation)

```
Input: Training data subset $D_j = \{(x_{ij},y_{ij})\}_{i=1}^{|D_j|}$ stored in the $j$-th local machine for
    $j=1,\cdots,m$, query data $D'$, and a candidate set of the regularization parameter
    $\Lambda=\{\lambda_\ell\}_{\ell=1}^L$.
# Training process
1: for $j=1,\cdots,m$ do
2:     Generate $\iota$ groups of training and validation sets via $\iota$-fold cross validation on data
       $D_j$, and denote them as $\left\{\left(D^{tr}_{j,l},D^{val}_{j,l}\right)\right\}_{l=1}^{\iota}$.                ▷ Data Split
3:     Generate the same centers $\Xi_n=\{\xi_k\}_{k=1}^n$ satisfying (8) and define a set of functions
       $B_{n,K} := \{\sum_{k=1}^n a_k K_{\xi_k} : a_k\in\mathbb{R}\}$, where $K_\xi(x)=K(\xi,x)$.        ▷ Basis Generation
4: end for
5: for $l=1,\cdots,\iota$ do                                   ▷ Operate on the $l$-th Split
6:     for $j=1,\cdots,m$ do
7:         Given $\lambda_\ell\in\Lambda$, run KRR with data $D^{tr}_{j,l}$ to obtain        ▷ Local Processing
               $$f_{D^{tr}_{j,l},\lambda_\ell} = \arg\min_{f\in\mathcal{H}_K}\left\{\frac{1}{\left|D^{tr}_{j,l}\right|}\sum_{(x,y)\in D^{tr}_{j,l}}(f(x)-y)^2+\lambda_\ell\|f\|_K^2\right\}.$$
8:         Generate a set of points $F^{tr}_{j,l,\ell}=\left\{\left(x_{i,j},f_{D^{tr}_{j,l},\lambda_\ell}(x_{i,j})\right)\right\}_{i=1}^{\left|D^{tr}_{j,l}\right|}$, and run KRR with
           $F^{tr}_{j,l,\ell}$ to obtain an approximation of $f_{D^{tr}_{j,l},\lambda_\ell}$ by       ▷ Local Approximation
               $$f^{local}_{D^{tr}_{j,l},\lambda_\ell,n,\mu} = \arg\min_{f\in B_{n,K}}\frac{1}{\left|D^{tr}_{j,l}\right|}\sum_{(x,y)\in F^{tr}_{j,l,\ell}}(f(x)-y)^2+\mu\|f\|_K^2=\sum_{k=1}^n a^{local}_{j,l,k,\ell}K_{\xi_k}.$$
9:         Transmit $\left(a^{local}_{j,l,k,\ell}\right)_{k=1,\ell=1}^{n,L}$ to the global machine.          ▷ Communication(I)
```

```
        end for
        Synthesize the coefficients by $a^{global}_{l,k,\ell} = \sum_{j=1}^{m} \frac{|D^{tr}_{j,l}|}{|D^{tr}_{l}|} a^{local}_{j,l,k,\ell}$, and communicate the matrix
        $\left(a^{global}_{l,k,\ell}\right)_{k=1,\ell=1}^{n,L}$ to each local machine.    ▷ Synthesization and Communication(II)
        for $j = 1, \cdots, m$ do
            Compute $L$ MSEs on the validation set $D^{val}_{j,l}$ by    ▷ Local Validation
                $e_{j,l,\ell} = \frac{1}{|D^{val}_{j,l}|} \left\| \vec{y}_{D^{val}_{j,l}} - \pi_M \left( \mathbb{K}_{|D^{val}_{j,l}|,n} \vec{a}^{global}_{l,\ell} \right) \right\|_2^2 \quad \text{for} \quad \ell = 1, \cdots, L,$
            and denote the MSE vector with size $L$ as $\vec{e}_{j,l} := (e_{j,l,1}, \cdots, e_{j,l,L})^T$.
        end for
    end for
    for $j = 1, \cdots, m$ do
        Average $\vec{e}_{j,l}$ over the $\iota$-fold, i.e., $\bar{\vec{e}}_j = \frac{1}{\iota} \sum_{l=1}^{\iota} \vec{e}_{j,l}$, and select the optimal parameter
        $\lambda_j^* \in \Lambda$ with the minimum element in the vector $\bar{\vec{e}}_j$.    ▷ Parameter Selection
        Run KRR with data $D_j$ under the optimal parameter $\lambda_j^*$ to obtain    ▷ Retraining
            $f_{D_j,\lambda_j^*} = \arg\min_{f \in \mathcal{H}_K} \left\{ \frac{1}{|D_j|} \sum_{(x,y) \in D_j} (f(x) - y)^2 + \lambda_j^* \|f\|_K^2 \right\} = \sum_{i=1}^{|D_j|} \alpha^*_{ij} K_{x_{ij}}.$
    end for
    # Testing process
    Distribute the query data $D'$ to $m$ local machines.
    for $j = 1, \cdots, m$ do
        Compute a vector of size $|D'|$ by $\pi_M f_{D_j,\lambda_j^*}(D') = \pi_M \left( \mathbb{K}_{|D'|,|D_j|} \vec{\alpha}^*_{D_j} \right)$, and communicate it to the global machine.    ▷ Communication(III)
    end for
    Synthesize the final estimate by
        $\overline{f}^{Ada}_{D,\vec{\lambda}^*}(D') := \sum_{j=1}^{m} \frac{|D_j|}{|D|} \pi_M f_{D_j,\lambda_j^*}(D').$
                                                                        ▷ Global Estimates
Output: The global estimate $\overline{f}^{Ada}_{D,\vec{\lambda}^*}(D')$ for the query data $D'$.
```

Appendix B: Introduction to Low-Discrepancy Sequences: Halton Sequences and Sobol Sequences

In this appendix, we present four parts. First, we introduce quasi-Monte Carlo integration. Second, we discuss the discrepancy of a point set and its connection to quasi-Monte Carlo integration. Third, we provide a detailed explanation of two low-discrepancy sequences: Halton sequences and Sobol sequences, focusing on their generation processes and computational costs. Finally, we highlight the effectiveness of Halton and Sobol sequences in numerical integration and the advantages of using them within the AdaDKRR. We refer the readers to Dick and Pillichshammer (2010) and Leobacher and Pillichshammer (2014) for further details on quasi-Monte Carlo integration and the content presented here.

6.1 Quasi-Monte Carlo (QMC) integration

To approximate the integral of a function $f : [0,1]^s \to \mathbb{R}$, we can use an equal-weight quadrature rule of the form

$$\frac{1}{n}\sum_{\tilde{n}=0}^{n-1} f(x_{\tilde{n}}),$$

where the quadrature points $\mathcal{P} = \{x_0, \ldots, x_{n-1}\}$ are chosen from $[0,1)^s$. This approach is known as the quasi-Monte Carlo (QMC) rule. To achieve higher accuracy in numerical integration, we focus on the integration error

$$e(f,\mathcal{P}) := \int_{[0,1]^s} f(x)\,\mathrm{d}x - \frac{1}{n}\sum_{\tilde{n}=0}^{n-1} f(x_{\tilde{n}}).$$

The estimator

$$Q_{n,s}(f) := \frac{1}{n}\sum_{\tilde{n}=0}^{n-1} f(X_{\tilde{n}})$$

serves as an approximation for the integral, with the integration error offering valuable insights into the performance of QMC integration. Our objective is to deterministically select quadrature points to construct an estimator that minimizes the integration error for specific classes of functions.

6.2 Discrepency: assessing the quality of uniform distribution

For a sequence $\mathcal{S} = (\boldsymbol{x}_{\tilde{n}})_{\tilde{n}\in\mathbb{N}_0}$ in $[0,1)^d$ and an interval $[\boldsymbol{a},\boldsymbol{b}) \subseteq [0,1)^d$, the number of indices $\tilde{n} \in \{0,\ldots,n-1\}$ for which $\boldsymbol{x}_{\tilde{n}} \in [\boldsymbol{a},\boldsymbol{b})$ is denoted by $A([\boldsymbol{a},\boldsymbol{b}),\mathcal{S},n)$. Precisely,

$$A([\boldsymbol{a},\boldsymbol{b}),\mathcal{S},n) = \#\left\{\tilde{n}\in\mathbb{N}_0 : 0 \le \tilde{n} \le n-1 \text{ and } \boldsymbol{x}_{\tilde{n}} \in [\boldsymbol{a},\boldsymbol{b})\right\}.$$

For finite point sets $\mathcal{P} = \{\boldsymbol{x}_0,\ldots,\boldsymbol{x}_{n-1}\}$, we have

$$A([\boldsymbol{a},\boldsymbol{b}),\mathcal{P},n) = \#\left\{\tilde{n}\in\mathbb{N}_0 : 0 \le \tilde{n} \le n-1 \text{ and } \boldsymbol{x}_{\tilde{n}} \in [\boldsymbol{a},\boldsymbol{b})\right\}.$$

Leobacher and Pillichshammer (2014) provides the following definition:

Definition 7 (Uniformly Distributed Modulo One) *An infinite sequence $\mathcal{S}$ in $[0,1)^d$ is said to be uniformly distributed modulo one (or equidistributed) if, for every interval of the form $[\boldsymbol{a},\boldsymbol{b}) \subseteq [0,1)^d$, the following holds:*

$$\lim_{n\to\infty}\frac{A([\boldsymbol{a},\boldsymbol{b}),\mathcal{S},n)}{n} = \lambda_d([\boldsymbol{a},\boldsymbol{b})),$$

where $\lambda_d([\boldsymbol{a},\boldsymbol{b})) = \prod_{j=1}^{d}(b_j - a_j)$ denotes the d-dimensional Lebesgue measure.

According to (Leobacher and Pillichshammer, 2014, Theorem 2.2), for every Riemann integrable function $f : [0,1]^d \to \mathbb{R}$, a sequence $(\boldsymbol{x}_{\tilde{n}})_{\tilde{n}\in\mathbb{N}_0}$ in $[0,1)^d$ that satisfies Definition 7 enables the following:

$$\lim_{n\to\infty}\frac{1}{n}\sum_{\tilde{n}=0}^{n-1} f\left(\boldsymbol{x}_{\tilde{n}}\right) = \int_{[0,1]^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.$$

This implies that, to obtain a QMC rule that converges to the actual value of the integral, the quadrature points must be uniformly distributed modulo one.

We now introduce a sequence called the van der Corput sequence, which satisfies uniform distribution modulo one. Let $b \in \mathbb{N}, b \geq 2$. The b-adic radical inverse function is defined as $\phi_b : \mathbb{N}_0 \to [0, 1)$ by

$$\phi_b(\tilde{n}) = \frac{n_0}{b} + \frac{n_1}{b^2} + \frac{n_2}{b^3} + \cdots ,$$

where $\tilde{n} \in \mathbb{N}_0$ has the b-adic digit expansion $\tilde{n} = n_0+n_1b+n_2b^2+\cdots$, with $n_i \in \{0, 1, \ldots, b-1\}$. The van der Corput sequence in base b is defined as $(x_{\tilde{n}})_{\tilde{n}\in\mathbb{N}_0}$, where $x_{\tilde{n}} = \phi_b(\tilde{n})$, and is uniformly distributed modulo one (Leobacher and Pillichshammer, 2014, Theorem 2.2). For example, for $b = 2$, the b-adic digit expansion is $0, 1, 10, 11, 100, \ldots$ for $\tilde{n} = 0, 1, 2, 3, 4, \ldots$, leading to the van der Corput sequence in base 2: $0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \ldots$.

Next, we introduce quantitative measures for assessing the quality of uniform distribution.

Definition 8 (Discrepancy) *For an n-element point set $\mathcal{P}$ with each element in $[0, 1)^d$, the discrepancy D_n is defined as*

$$D_n(\mathcal{P}) = \sup_{\substack{\boldsymbol{a},\boldsymbol{b}\in[0,1)^d \\ \boldsymbol{a}\leq\boldsymbol{b}}} \left| \frac{A([\boldsymbol{a}, \boldsymbol{b}), \mathcal{P}, n)}{n} - \lambda_s([\boldsymbol{a}, \boldsymbol{b})) \right| ,$$

where $\boldsymbol{a} \leq \boldsymbol{b}$ means $a_j \leq b_j$ for all $j = 1, \ldots, d$.

The discrepancy $D_n(\mathcal{S})$ of an infinite sequence $\mathcal{S}$ is defined as the discrepancy of the first N elements of $\mathcal{S}$. Often, it suffices to consider the star discrepancy, a slightly simplified notion of discrepancy:

$$D_n^*(\mathcal{P}) = \sup_{\boldsymbol{a}\in[0,1]^d} \left| \frac{A([\mathbf{0}, \boldsymbol{a}), \mathcal{P}, n)}{n} - \lambda_d([\mathbf{0}, \boldsymbol{a})) \right| .$$

Based on these notions of discrepancy, the following two assertions are equivalent to $\mathcal{S}$ being uniformly distributed modulo one (Leobacher and Pillichshammer, 2014, Theorem 2.15): (a) $\lim_{n\to\infty} D_n^*(\mathcal{S}) = 0$; (b) $\lim_{n\to\infty} D_n(\mathcal{S}) = 0$. This indicates that, to construct a QMC rule that converges to the exact value of the integral, the quadrature points or sequences must exhibit low discrepancy.

6.3 Low-discrepancy sequences: Halton sequences and Sobol sequences

In this section, we introduce two low-discrepancy sequences, Halton sequences and Sobol sequences, focusing on their generation processes, computational costs, and effectiveness in numerical integration.

6.3.1 Halton sequences

In the second part, we introduce the one-dimensional van der Corput sequence in base b, which is uniformly distributed modulo one and serves as a prototype for many uniformly distributed sequences and point sets with low star discrepancy, even in higher dimensions. The

Halton sequence, one of its classical generalizations, is an infinite sequence constructed by concatenating component-wise elements of van der Corput sequences generated in different bases.

Definition 9 (Halton sequence) *Let $s \in \mathbb{N}$ and let $b_1, \dots, b_d \geq 2$ be integers. The Halton sequence in bases $b_1, \dots, b_d$ is the sequence $\mathcal{S}_{b_1.....b_d} = (\boldsymbol{x}_{\tilde{n}})_{\tilde{n} \in \mathbb{N}_0}$ whose nth element is given by*

$$\boldsymbol{x}_{\tilde{n}} := (\phi_{b_1}(\tilde{n}), \phi_{b_2}(\tilde{n}), \dots, \phi_{b_d}(\tilde{n})) \text{ for } \tilde{n} \in \mathbb{N}_0$$

For $d = 1$ we recover the van der Corput sequence $(\phi_b(\tilde{n}))_{\tilde{n} \in \mathbb{N}_0}$ in base $b = b_1$.

We previously showed that the van der Corput sequence in base 2 is $0, \frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8} \dots$. Following its generating process, the van der Corput sequence in base 3 is $0, \frac{1}{3}, \frac{2}{3}, \frac{1}{9}, \frac{4}{9}, \dots$. Consequently, a Halton sequence in dimension $d = 2$ can be constructed with bases $b_1 = 2$ and $b_2 = 3$: $\boldsymbol{x}_0 = (0, 0), \boldsymbol{x}_1 = (1/2, 1/3), \boldsymbol{x}_2 = (1/4, 2/3), \boldsymbol{x}_3 = (3/4, 1/9), \boldsymbol{x}_4 = (1/8, 4/9), \dots$. Figure 15(a) illustrates the first 8 elements of the Halton sequence with specific coordinates, while Figure 16(a) shows the first 128 elements of this sequence. Compared with the uniform random sampling shown in Figure 16(c), Halton sequence exhibits a more evenly distributed pattern.

Computational cost: The generation of Halton sequences relies on base conversion and reverse computation. For each number $\tilde{n}$, converting it into its base-b representation requires $\mathcal{O}(\log_b \tilde{n})$ operations for modulo and division operations, as $\tilde{n}$ has $\lfloor \log_b \tilde{n} \rfloor + 1$ digits in base b. Once the base-b representation is obtained, calculating the reverse value also requires $\mathcal{O}(\log_b \tilde{n})$ operations. The total complexity of generating the first n values of the Halton sequence is $\mathcal{O}(n \log_b n)$, since the values are generated sequentially from $\tilde{n} = 1$ to $\tilde{n} = n$.

6.3.2 Sobol sequences

The Sobol' sequence can be constructed as follows: Let $p_1, \dots, p_d \in \mathbb{F}_2[x]$ be primitive polynomials ordered according to their degree, and let

$$p_j(x) = x^{e_j} + a_{1,j}x^{e_j-1} + a_{2,j}x^{e_j-2} + \cdots + a_{e_j-1}x + 1 \quad \text{for } j = 1, 2, \dots, d,$$

where $\mathbb{F}_b$ is a finite field consisting of b elements, typically $\{0, 1, \dots, b-1\}$, where addition and multiplication are defined modulo b. $\mathbb{F}_b[x]$ is the set of all polynomials over the finite field $\mathbb{F}_b$, containing all polynomials whose coefficients are elements of $\mathbb{F}_b$.

Select odd natural numbers $1 \leq m_{1,j}, \dots, m_{e_j,j}$ such that $m_{k,j} < 2^k$ for $1 \leq k \leq e_j$, and for all $k > e_j$ define $m_{k,j}$ recursively by

$$m_{k,j} = 2a_{1,j}m_{k-1,j} \oplus \cdots \oplus 2^{e_j-1}a_{e_j-1}m_{k-e_j+1,j} \oplus 2^{e_j}m_{k-e_j,j} \oplus m_{k-e_j,j}$$

where $\oplus$ denotes the bitwise exclusive-or (XOR) operator. The values $v_{k,j} := \frac{m_{k,j}}{2^k}$ are referred to as direction numbers.

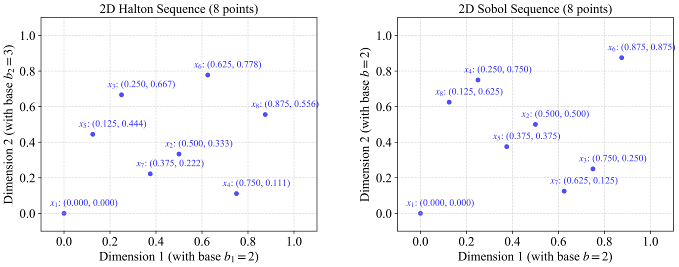
For $\tilde{n} \in \mathbb{N}_0$ with a binary expansion $\tilde{n} = n_0 + 2n_1 + \cdots + 2^{r-1}n_{r-1}$, the corresponding sequence elements are defined as

$$x_{\tilde{n},j} = n_0v_{1,j} \oplus n_1v_{2,j} \oplus \cdots \oplus n_{r-1}v_{r,j} \quad \text{and} \quad \boldsymbol{x}_{\tilde{n}} = (x_{\tilde{n},1}, \dots, x_{\tilde{n},d})$$

The Sobol' sequence is then the sequence of points $(\boldsymbol{x}_{\tilde{n}})_{\tilde{n}\in\mathbb{N}_0}$.

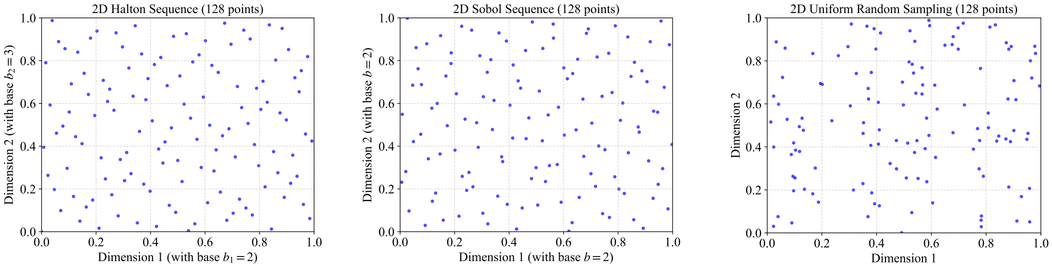
Figure 15(b) illustrates the first 8 elements of the Sobol sequence with specific coordinates, while Figure 16(b) shows the first 128 elements of the sequence. Compared with the uniformly random sampled elements shown in Figure 16(c), Sobol sequence exhibits a more evenly distributed pattern.

Computational cost: The Sobol sequence generation involves three key steps: binary expansion, direction number calculations, and bitwise XOR computation. In practice, precomputed direction numbers are used to simplify the process. Thus, we consider only the costs of binary expansion and bitwise XOR computation. Converting $\tilde{n}$ to its binary representation and reversing it requires $\mathcal{O}(\log_b n)$ operations. For a d-dimensional Sobol sequence, each dimension performs a bitwise XOR with the corresponding direction numbers, with a complexity of $\mathcal{O}(d)$. Hence, generating a single point has a complexity of $\mathcal{O}(\log_b n + d)$, which simplifies to $\mathcal{O}(\log_b n)$ when d is small and independent of n. Consequently, the total complexity of generating the first n points of the Sobol sequence is $\mathcal{O}(n \log_b n)$.



(a) Halton sequence (b) Sobol sequence

Figure 15: Low-discrepency sequences



(a) Halton sequence (b) Sobol sequence (c) Uniform random sampling

Figure 16: Low-discrepency sequences and uniform random sampling

6.4 Benefits of using low-discrepancy sequences in AdaDKRR

Based on the definition of discrepancy, a point set $\mathcal{P}$ is referred to as a low-discrepancy point set if its star-discrepancy $D_n^*(\mathcal{P})$ satisfies an upper bound of the form $D_n^*(\mathcal{P}) = \mathcal{O}\left(\frac{(\log n)^{\alpha_d}}{n}\right)$ as $n \to \infty$, where $\alpha_d \geq 0$ (Dick and Pillichshammer, 2010). Both the Halton sequence and the Sobol sequence are low-discrepancy, which enhances the accuracy of QMC integration, as discussed in Sections 6.1-6.2. Specifically, the $D_n^*(\mathcal{P})$ of Halton sequence is asymptotically of the order $\mathcal{O}((\log n)^d/n)$ with $\alpha_d = d$, as shown in (Leobacher and Pillichshammer, 2014, Theorem 2.35). Although the Sobol and Halton sequences share the theoretical discrepancy upper bound $\mathcal{O}((\log n)^d/n)$, the Sobol sequence's carefully designed direction numbers reduce inter-dimensional correlations, making the α_d value often smaller than d in practice, resulting in a more uniform distribution and making it more effective for numerical integration compared to the Halton sequence in high-dimensional applications. However, the asymptotic expression for the star discrepancy of uniform random sampling is $D_n^* = \mathcal{O}\left(\sqrt{\frac{\log n}{n}}\right)$, which is slower than the asymptotic expression $\mathcal{O}((\log n)^{\alpha_d}/n)$. Consequently, a disadvantage of Monte Carlo methods (based on uniform random sampling) is their slow convergence rate for integration error $e(f, \mathcal{P})$, which is of the order $\mathcal{O}(1/\sqrt{n})$ (Leobacher and Pillichshammer, 2014), significantly slower than the $\mathcal{O}(1/n)$ convergence rate of low-discrepancy methods (ignoring logarithmic factors).

Table 4: Comparison of Halton sequences, Sobol sequences, and uniform random sampling

Methods	**Concerns in addressing data silos during local processing stage**		
	Star discrepency D_n^* *(Local approximation accuracy concern)*	**Point generation cost** *(Computation efficiency concern)*	**Reproducibility on local machines** *(Communication efficiency concern)*
Halton sequences	$\mathcal{O}((\log n)^d/n)$	$\mathcal{O}(n \log_b n)$	✓
Sobol sequences	$\mathcal{O}((\log n)^d/n)$	$\mathcal{O}(n \log_b d)$	✓
Uniform random sampling	$\mathcal{O}(\sqrt{\frac{\log n}{n}})$	$\mathcal{O}(n)$	×

Note: n represents the number of points, and d denotes their dimensionality.

Table 4 compares Halton sequences, Sobol sequences, and uniform random sampling across different evaluation metrics. From the comparison, we observe that Halton and Sobol sequences exhibit low discrepancy, enabling more accurate numerical integration approximations. Additionally, they achieve this without incurring significant computational cost compared to uniform random sampling. Furthermore, these sequences can be reproduced independently by distributed local machines, allowing local machines to use the same basis functions without requiring communication about the set of centers. Overall, Halton and Sobol sequences are well-suited for addressing data silos during the local processing.

Appendix C: Proofs

In this part, we use the well-developed integral operator approach (Smale and Zhou, 2007; Lin et al., 2017; Rudi et al., 2015) to prove our main results. Our main novelty in the proof is the detailed analysis of the role of the weight $\frac{|D_j|}{|D|}$ in (1) and a tight bound on local approximation. Our proofs are divided into four steps: error decomposition, local approximation and global approximation, generalization error of DKRR, and generalization error of AdaDKRR.

6.5 Error decomposition based on integral operators

Let D be a set of data drawn i.i.d. according to a distribution ρ. Denote by $S_D : \mathcal{H}_K \to \mathbb{R}^{|D|}$ the sampling operator on $\mathcal{H}_K$

$$S_D f := \{f(x_i)\}_{(x_i,y_i)\in D}.$$

Its scaled adjoint $S_D^T : \mathbb{R}^{|D|} \to \mathcal{H}_K$ is

$$S_D^T \mathbf{c} := \frac{1}{|D|}\sum_{(x_i,y_i)\in D} c_i K_{x_i}, \qquad \mathbf{c} \in \mathbb{R}^{|D|}.$$

Let $L_{K,D}$ be the empirical version of L_K and it is defined by

$$L_{K,D} f := S_D^T S_D f = \frac{1}{|D|}\sum_{(x,y)\in D} f(x) K_x.$$

Then we have (Smale and Zhou, 2005)

$$f_{D,\lambda} = (L_{K,D} + \lambda I)^{-1} S_D^T y_D, \tag{23}$$

where $y_D := (y_1, \ldots, y_{|D|})^T$.

To present the error decomposition of DKRR, we need the following lemma that can be easily deduced from (Guo et al., 2017a, Prop.4) or (Chang et al., 2017b, Prop.5).

Lemma 10 *Let $\overline{f}_{D,\vec{\lambda}}$ be defined by (1). We have*

$$E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right] \leq 2\sum_{j=1}^m \frac{|D_j|^2}{|D|^2} E\left[\left\|f_{D_j,\lambda_j} - f^\diamond_{D_j,\lambda_j}\right\|_\rho^2\right] + 2\sum_{j=1}^m \frac{|D_j|}{|D|} E\left[\left\|f^\diamond_{D_j,\lambda_j} - f_\rho\right\|_\rho^2\right], \tag{24}$$

where

$$f^\diamond_{D,\lambda} := (L_{K,D} + \lambda I)^{-1} L_{K,D} f_\rho \tag{25}$$

is the noise-free version of $f_{D,\lambda}$.

With the help of the above lemma, we can derive the following error decomposition for DKRR based on integral operators.

Proposition 11 *Let $\overline{f}_{D,\vec{\lambda}}$ be defined by (1). If Assumption 2 holds with $\frac{1}{2} \leq r \leq 1$, then*

$$E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right] \leq 2\sum_{j=1}^m \frac{|D_j|}{|D|}\lambda_j^{2r} E\left[\mathcal{Q}^{4r}_{D_j,\lambda}\right] \|h_\rho\|_\rho^2 + 2\sum_{j=1}^m \frac{|D_j|^2}{|D|^2} E\left[\mathcal{Q}^4_{D_j,\lambda_j}\mathcal{P}^2_{D_j,\lambda_j}\right], \tag{26}$$

where

$$\mathcal{Q}_{D,\lambda} := \left\|(L_{K,D} + \lambda I)^{-1/2}(L_K + \lambda I)^{1/2}\right\|, \tag{27}$$

$$\mathcal{P}_{D,\lambda} := \left\|(L_K + \lambda I)^{-1/2}(L_{K,D} f_\rho - S_D^T y_D)\right\|_K. \tag{28}$$

Proof of Proposition 11. Due to (23) and (25), we obtain

$$f^{\diamond}_{D,\lambda} - f_\rho = ((L_{K,D}+\lambda I)^{-1}L_{K,D} - I)f_\rho = \lambda(L_{K,D}+\lambda I)^{-1}f_\rho \tag{29}$$

and

$$f^{\diamond}_{D,\lambda} - f_{D,\lambda} = (L_{K,D}+\lambda I)^{-1}(L_{K,D}f_\rho - S_D^T y_D). \tag{30}$$

Then, we get from (27), (28), (30), (29) with $D = D_j$, $\lambda = \lambda_j$, and Assumption 2 with $\frac{1}{2} \le r \le 1$ that for any $1 \le j \le m$, there holds

$$\begin{aligned} &\|f^{\diamond}_{D_j,\lambda_j} - f_\rho\|_\rho \le \lambda_j \|(L_K+\lambda_j I)^{1/2}(L_{K,D_j}+\lambda_j I)^{-1}f_\rho\|_K \\ \le \quad & \lambda_j \mathcal{Q}_{D_j,\lambda_j}\|(L_{K,D_j}+\lambda_j I)^{-1/2}L_K^{r-1/2}\|\|h_\rho\|_\rho \le \lambda_j^r \mathcal{Q}^{2r}_{D_j,\lambda}\|h_\rho\|_\rho \end{aligned}$$

and

$$\|f_{D_j,\lambda_j} - f^{\diamond}_{D_j,\lambda_j}\|_\rho \le \|(L_K+\lambda_j I)^{1/2}(L_{K,D_j}+\lambda_j I)^{-1}(S^T_{D_j}y_{D_j} - L_{K,D}f_\rho)\|_K \le \mathcal{Q}^2_{D_j,\lambda_j}\mathcal{P}_{D_j,\lambda_j}.$$

Plugging the above two estimates into (24), we then obtain

$$\frac{1}{2}E\left[\|\overline{f}_{D,\vec{\lambda}} - f_\rho\|_\rho^2\right] \le \sum_{j=1}^m \frac{|D_j|}{|D|}\lambda_j^{2r}E\left[\mathcal{Q}^{4r}_{D_j,\lambda}\right]\|h_\rho\|_\rho^2 + \sum_{j=1}^m \frac{|D_j|^2}{|D|^2}E\left[\mathcal{Q}^4_{D_j,\lambda_j}\mathcal{P}^2_{D_j,\lambda_j}\right].$$

This completes the proof of Proposition 11. ■

We next derive an error decomposition for AdaDKRR in the following proposition.

Proposition 12 *For any $\delta > 0$ and any $\lambda \in \Lambda$, if Assumption 1 holds and $|D_j^{tr}| \sim |D_j^{val}|$, then*

$$\begin{aligned} E\left[\left\|\overline{f}^{Ada}_{D,\vec{\lambda}^*} - f_\rho\right\|_\rho^2\right] &\le \frac{C_1'(1+\log|\Lambda|)m}{|D|} \\ + 4\min_{\lambda\in\Lambda}&\left\{\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}E\left[\left\|f^{global}_{D^{tr},\lambda,n,\mu} - \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda}\right\|_\rho^2\right] + E\left[\left\|\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda} - f_\rho\right\|_\rho^2\right]\right\}, \end{aligned} \tag{31}$$

where C_1' is an absolute constant.

To prove the above proposition, we need the following lemma which can be found in (Györfi et al., 2002, Theorem 7.1) (see also (Caponnetto and Yao, 2010) for a probabilistic argument).

Lemma 13 *For a given data set D, let D^{tr} and D^{val} be the training set and validation set respectively. Under Assumption 1, for any $\delta > 0$ and any $\lambda \in \Lambda$, there holds*

$$E\left[\left\|\pi_M f_{D^{tr},\hat{\lambda}} - f_\rho\right\|_\rho^2\right] \le (1+\delta)\min_{\lambda\in\Lambda}E\left[\left\|\pi_M f_{D^{tr},\lambda} - f_\rho\right\|_\rho^2\right] + C'\frac{1+\log|\Lambda|}{|D^{val}|},$$

where $C' = M^2(16/\delta + 35 + 19\delta)$.

With the help of the above lemma, we can prove Proposition 12 as follows.

Proof of Proposition 12. Due to (7), we have from Lemma 13, $|D_j^{val}| \sim |D_j^{tr}|$, (6), and Jensen's inequality that

$$\begin{aligned}
& E\left[\left\|\overline{f}^{Ada}_{D,\vec{\lambda}^*}-f_\rho\right\|_\rho^2\right] \leq \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} E\left[\left\|\pi_M f^{global}_{D^{tr},\lambda_j^*,n,\mu}-f_\rho\right\|_\rho^2\right] \\
\leq & \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}\left((1+\delta)\min_{\lambda\in\Lambda} E\left[\left\|\pi_M f^{global}_{D^{tr},\lambda,n,\mu}-f_\rho\right\|_\rho^2\right]+\frac{C'(1+\log|\Lambda|)}{|D_j^{val}|}\right) \\
\leq & (1+\delta)\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}\min_{\lambda\in\Lambda} E\left[\left\|f^{global}_{D^{tr},\lambda,n,\mu}-f_\rho\right\|_\rho^2\right]+\frac{C_1'(1+\log|\Lambda|)m}{|D|} \\
\leq & \frac{C_1'm(1+\log|\Lambda|)}{|D|}+2(1+\delta)\min_{\lambda\in\Lambda}\left\{\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} E\left[\left\|f^{global}_{D^{tr},\lambda,n,\mu}-\sum_{j=1}^m\frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda}\right\|_\rho^2\right]\right. \\
& \left. +E\left[\left\|\sum_{j=1}^m\frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda}-f_\rho\right\|_\rho^2\right]\right\},
\end{aligned}$$

where we use $(a+b)^2 \leq 2a^2+2b^2$ in the last inequality and C', C_1 are absolute constants. This completes the proof of Proposition 12 by setting $\delta=1$. ■

6.6 Local approximation and global approximation

Due to Proposition 12, it is crucial to derive the error of the global approximation (5) as well as the local approximation (4). In this appendix only, we denote by $f^{loc}_{D,\lambda,\mu}$ the local approximation (4) with $D_j^{tr}=D$, $s=|D|$, and $\lambda_\ell=\lambda$ for the sake of brevity. The derived error of local approximation is shown in the following proposition.

Proposition 14 *If (11) holds with $1/2 \leq r \leq 1$, then we have*

$$\begin{aligned}
& \left\|f^{loc}_{D,\lambda,\mu}-f_{D,\lambda}\right\|_\rho \leq \mu^r\|h_\rho\|_\rho\left(\left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\mathcal{Q}^{2r}_{\Xi_n,\mu}+\mathcal{Q}^{2r}_{D,\mu}\right) \\
+ & \mu^{1/2}\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right)\left(\mathcal{Q}_{D,\mu}+\left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\mathcal{Q}_{\Xi_n,\mu}\right), \quad (32)
\end{aligned}$$

where $\|\cdot\|$ denotes the operator norm and

$$\mathcal{Q}^*_{D,\mu} := \left\|(L_K+\mu I)^{-1/2}(L_{K,D}+\mu I)^{1/2}\right\|. \quad (33)$$

Before providing the proof of the above proposition, we introduce several interesting tools. Let P_{Ξ_n} be the projection from $\mathcal{H}_K$ to $B_{n,K}$. Then for an arbitrary $\nu>0$, there holds

$$(I-P_{\Xi_n})^\nu = I-P_{\Xi_n}. \quad (34)$$

Let $T_{\Xi_n}: B_{n,K} \to \mathbb{R}^n$ be a sampling operator defined by $T_{\Xi_n} f := \{f(x_i)\}_{x_i \in \Xi_n}$ such that the range of its adjoint operator $T_{\Xi_n}^T$ is exactly in $B_{n,K}$, and let $U\Sigma V^T$ be the SVD of T_{Ξ_n}. Then we have

$$V^T V = I, \qquad VV^T = P_{\Xi_n}. \tag{35}$$

Write

$$g_{\Xi_n,\mu}(L_{K,D}) := V(V^T L_{K,D} V + \mu I)^{-1} V^T. \tag{36}$$

Then it can be found in (9) and (Rudi et al., 2015) that

$$f^{loc}_{D,\lambda,\mu} = g_{\Xi_n,\mu}(L_{K,D}) L_{K,D} f_{D,\lambda}. \tag{37}$$

For an arbitrary bounded linear operator B, it follows from (35) and (36) that

$$g_{\Xi_n,\mu}(L_{K,D})(L_{K,D} + \mu I) V B V^T = V B V^T. \tag{38}$$

Inserting $B = (V^T L_{K,D} V + \mu I)^{-1}$ into (38), it is easy to derive (Sun et al., 2022)

$$\|(L_{K,D} + \mu I)^{1/2} g_{\Xi_n,\mu}(L_{K,D})(L_{K,D} + \mu I)^{1/2}\| \leq 1. \tag{39}$$

Besides the above tools, we also need the following two lemmas that can be found in (Rudi et al., 2015, Proposition 3) and (Rudi et al., 2015, Proposition 6), respectively.

Lemma 15 *Let $\mathcal{H}$, $\mathcal{K}$, and $\mathcal{F}$ be three separable Hilbert spaces. Let $Z: \mathcal{H} \to \mathcal{K}$ be a bounded linear operator and P be a projection operator on $\mathcal{H}$ such that $rangeP = \overline{rangeZ^T}$. Then for any bounded linear operator $F: \mathcal{F} \to \mathcal{H}$ and any $\lambda > 0$, we have*

$$\|(I - P)F\| \leq \lambda^{1/2} \left\|(Z^T Z + \lambda I)^{-1/2} F\right\|.$$

Lemma 16 *Let $\mathcal{H}$ and $\mathcal{K}$ be two separable Hilbert spaces, $A: \mathcal{H} \to \mathcal{H}$ be a positive linear operator, $V_{\mathcal{H},\mathcal{K}}: \mathcal{H} \to \mathcal{K}$ be a partial isometry, and $B: \mathcal{K} \to \mathcal{K}$ be a bounded operator. Then for all $0 \leq r^*, s^* \leq 1/2$, there holds*

$$\left\|A^{r^*} V_{\mathcal{H},\mathcal{K}} B V_{\mathcal{H},\mathcal{K}}^T A^{s^*}\right\| \leq \left\|(V_{\mathcal{H},\mathcal{K}}^T A V_{H,K})^{r^*} B (V_{\mathcal{H},\mathcal{K}}^T A V_{\mathcal{H},\mathcal{K}})^{s^*}\right\|.$$

With the help of the above lemmas and the important properties of $g_{D_m,\mu}$ in (38) and (39), we prove Proposition 14 as follows.

Proof of Proposition 14. The triangle inequality yields

$$\|f^{loc}_{D,\lambda,\mu} - f_{D,\lambda}\|_\rho \leq \mathcal{A}_n(D,\lambda,\mu) + \mathcal{C}_n(D,\lambda,\mu), \tag{40}$$

where

$$\begin{aligned}\mathcal{A}_n(D,\lambda,\mu) &= \|(g_{\Xi_n,\mu}(L_{K,D})L_{K,D} - I)P_{\Xi_n} f_{D,\lambda}\|_\rho, \\ \mathcal{C}_n(D,\lambda,\mu) &= \|(g_{\Xi_n,\mu}(L_{K,D})L_{K,D} - I)(I - P_{\Xi_n}) f_{D,\lambda}\|_\rho.\end{aligned}$$

We first bound $\mathcal{A}_n(D,\lambda,\mu)$. It follows from (38) with $B = I$ that

$$P_{\Xi_n} = g_{\Xi_n,\mu}(L_{K,D})(L_{K,D} + \mu I) P_{\Xi_n},$$

which together with the definition of $\mathcal{A}_n(D,\lambda,\mu)$ yields

$$\mathcal{A}_n(D,\lambda,\mu)=\mu\|g_{\Xi_n,\mu}(L_{K,D})P_{\Xi_n}f_{D,\lambda}\|_\rho.$$

Then we have from the triangle inequality that

$$\begin{aligned}\mathcal{A}_n(D,\lambda,\mu) &\leq \mu\left\|g_{\Xi_n,\mu}(L_{K,D})P_{\Xi_n}f_\rho\right\|_\rho+\mu\left\|g_{\Xi_n,\mu}(L_{K,D})P_{\Xi_n}(f_{D,\lambda}-f_\rho)\right\|_\rho\\ &=: \mathcal{A}_{n,1}(D,\mu)+\mathcal{A}_{n,2}(D,\lambda,\mu).\end{aligned}\tag{41}$$

To bound $\mathcal{A}_{n,1}(D,\mu)$, noting that $L_{K,D}$ is a positive operator, we have $\|(V^T(L_{K,D}+\mu I)V)^{r-1}\|\leq\mu^{r-1}$ for $r\leq 1$ (Rudi et al., 2015). Recalling further (35) and the Cordes inequality (Bhatia, 2013)

$$\|A^uB^u\|\leq\|AB\|^u,\qquad 0<u\leq 1\tag{42}$$

for arbitrary positive operators A and B, we get from Lemma 16 with $A=(L_{K,D}+\mu I)$, $V_{\mathcal{H},\mathcal{K}}=V$, $B=(V^TL_{K,D}V+\mu I)^{-1}$, $r^*=1/2$, and $s^*=r-1/2$ that

$$\begin{aligned}&\mathcal{A}_{n,1}(D,\mu)\leq\mu\left\|L_K^{1/2}g_{\Xi_n,\mu}(L_{K,D})VV^TL_K^{r-1/2}\right\|\|h_\rho\|_\rho\\ \leq\ &\mu\mathcal{Q}_{D,\mu}^{2r}\|h_\rho\|_\rho\left\|(L_{K,D}+\mu I)^{1/2}g_{\Xi_n,\mu}(L_{K,D})VV^T(L_{K,D}+\mu I)^{r-1/2}\right\|\\ \leq\ &\mu\mathcal{Q}_{D,\mu}^{2r}\left\|(V^T(L_{K,D}+\mu I)V)^{1/2}(V^T(L_{K,D}+\mu I)V)^{-1}(V^T(L_{K,D}+\mu I)V)^{r-1/2}\right\|\|h_\rho\|_\rho\\ \leq\ &\mu\mathcal{Q}_{D,\mu}^{2r}\left\|(V^T(L_{K,D}+\mu I))^{r-1}V\right\|\|h_\rho\|_\rho\\ \leq\ &\mu^r\mathcal{Q}_{D,\mu}^{2r}\|h_\rho\|_\rho.\end{aligned}\tag{43}$$

To bound $\mathcal{A}_{n,2}(D,\lambda,\mu)$, we need the following standard estimate of $\|f_{D,\lambda}-f_\rho\|_K$ (Caponnetto and De Vito, 2007; Steinwart et al., 2009; Lin et al., 2017; Chang et al., 2017b), which can be derived in a way similar to that used to prove Proposition 11. Under (11) with $r\leq 1$, there holds

$$\|f_{D,\lambda}-f_\rho\|_K\leq\lambda^{-1/2}\mathcal{Q}_{D,\lambda}^2\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}_{D,\lambda}^{2r-1}\|h\|_\rho.\tag{44}$$

Then, it follows from (35), (42), and Lemma 16 with $A=(L_{K,D}+\mu I)$, $V_{H,K}=V$, $B=(V^TL_{K,D}V+\mu I)^{-1}$, $r^*=1/2$, and $s^*=0$ that

$$\begin{aligned}&\mathcal{A}_{n,2}(D,\lambda,\mu)\leq\mu\left\|L_K^{1/2}g_{\Xi_n,\mu}(L_{K,D})VV^T\right\|\|f_{D,\lambda}-f_\rho\|_K\\ \leq\ &\mu\mathcal{Q}_{D,\mu}\left(\lambda^{-1/2}\mathcal{Q}_{D,\lambda}^2\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}_{D,\lambda}^{2r-1}\|h\|_\rho\right)\left\|(L_{K,D}+\mu I)^{1/2}g_{\Xi_n,\mu}(L_{K,D})VV^T\right\|\\ \leq\ &\mu\mathcal{Q}_{D,\mu}\left(\lambda^{-1/2}\mathcal{Q}_{D,\lambda}^2\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}_{D,\lambda}^{2r-1}\|h\|_\rho\right)\left\|(V^T(L_{K,D}+\mu I)V)^{-1/2}\right\|\\ \leq\ &\mu^{1/2}\mathcal{Q}_{D,\mu}\left(\lambda^{-1/2}\mathcal{Q}_{D,\lambda}^2\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}_{D,\lambda}^{2r-1}\|h\|_\rho\right).\end{aligned}\tag{45}$$

Plugging (43) and (45) into (41), we obtain

$$\mathcal{A}_n(D,\lambda,\mu)\leq\mu^r\mathcal{Q}_{D,\mu}^{2r}\|h_\rho\|_\rho+\mu^{1/2}\mathcal{Q}_{D,\mu}\left(\lambda^{-1/2}\mathcal{Q}_{D,\lambda}^2\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}_{D,\lambda}^{2r-1}\|h\|_\rho\right).\tag{46}$$

We next bound $\mathcal{C}_n(D,\lambda,\mu)$. Similar to the above, we have

$$\begin{aligned} \mathcal{C}_n(D,\lambda,\mu) &\leq \|(g_{\Xi_n,\mu}(L_{K,D})L_{K,D}-I)(I-P_{\Xi_n})f_\rho\|_\rho \\ &+ \|(g_{\Xi_n,\mu}(L_{K,D})L_{K,D}-I)(I-P_{\Xi_n})(f_{D,\lambda}-f_\rho)\|_\rho \\ &=: \mathcal{C}_{n,1}(D,\mu)+\mathcal{C}_{n,2}(D,\lambda,\mu). \end{aligned} \tag{47}$$

Due to Lemma 15, we have

$$\left\|(I-P_{\Xi_n})(L_K+\mu I)^{1/2}\right\| \leq \mu^{1/2}\left\|(L_{K,\Xi_n}+\mu I)^{-1/2}(L_K+\mu I)^{1/2}\right\| \leq \mu^{1/2}\mathcal{Q}_{\Xi_n,\mu}. \tag{48}$$

Then, it follows from (39), (42), and (34) with $\tau=2r$ that

$$\begin{aligned} &\mathcal{C}_{n,1}(D,\mu) \leq \left\|L_K^{1/2}g_{\Xi_n,\mu}(L_{K,D})L_{K,D}(I-P_{\Xi_n})L_K^{r-1/2}\right\|\|h_\rho\|_\rho \\ +\ & \left\|L_K^{1/2}(I-P_{\Xi_n})L_K^{r-1/2}\right\|\|h_\rho\|_\rho \\ \leq\ & \mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}\left\|(L_{K,D}+\mu I)^{1/2}g_{\Xi_n,\mu}(L_{K,D})(L_{K,D}+\mu I)^{1/2}\right\| \\ \times\ & \left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})^{2r}L_K^{r-1/2}\right\|\|h_\rho\|_\rho \\ +\ & \left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})^{2r}L_K^{r-1/2}\right\|\|h_\rho\|_\rho \\ \leq\ & \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\|h_\rho\|_\rho\left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})\right\|\left\|(I-P_{\Xi_n})^{2r-1}L_K^{r-1/2}\right\| \\ \leq\ & \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\|h_\rho\|_\rho\,\mu^r\mathcal{Q}^{2r}_{\Xi_n,\mu}. \end{aligned} \tag{49}$$

Noting (44), we get from (34), (39), (42), and (48) again that

$$\mathcal{C}_{n,2}(D,\lambda,\mu) \tag{50}$$

$$\begin{aligned} \leq\ & \left\|L_K^{1/2}g_{\Xi_n,\mu}(L_{K,D})L_{K,D}(I-P_{\Xi_n})\right\|\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right) \\ +\ & \left\|L_K^{1/2}(I-P_{\Xi_n})\right\|\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right) \\ \leq\ & \mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}\left\|(L_{K,D}+\mu I)^{1/2}g_{\Xi_n,\mu}(L_{K,D})(L_{K,D}+\mu I)^{1/2}\right\| \\ \times\ & \left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})\right\|\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right) \\ +\ & \left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})\right\|\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right) \\ \leq\ & \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right)\left\|(L_K+\mu I)^{1/2}(I-P_{\Xi_n})\right\| \\ \leq\ & \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right)\mu^{1/2}\mathcal{Q}_{\Xi_n,\mu}. \end{aligned} \tag{51}$$

Inserting (49) and (50) into (47), we have

$$\begin{aligned} \mathcal{C}_n(D,\lambda,\mu) &\leq \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right)\mu^{1/2}\mathcal{Q}_{\Xi_n,\mu} \\ &+ \left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\|h_\rho\|_\rho\mu^r\mathcal{Q}^{2r}_{\Xi_n,\mu}. \end{aligned} \tag{52}$$

Plugging (46) and (52) into (40), we get

$$\begin{aligned}&\left\|f^{loc}_{D,\lambda,\mu}-f_{D,\lambda}\right\|_\rho \le \mu^r\left\|h_\rho\right\|_\rho\left(\left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\mathcal{Q}^{2r}_{\Xi_n,\mu}+\mathcal{Q}^{2r}_{D,\mu}\right)\\ +\ \ &\mu^{1/2}\left(\lambda^{-1/2}\mathcal{Q}^2_{D,\lambda}\mathcal{P}_{D,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D,\lambda}\|h\|_\rho\right)\left(\mathcal{Q}_{D,\mu}+\left(\mathcal{Q}_{D,\mu}\mathcal{Q}^*_{D,\mu}+1\right)\mathcal{Q}_{\Xi_n,\mu}\right).\end{aligned}$$

This completes the proof of Proposition 14. ■

Based on Proposition 14, we can derive an error estimate for the global approximation directly.

Proposition 17 *If (11) holds with* $1/2 \le r \le 1$*, then*

$$\begin{aligned}&\left\|f^{global}_{D^{tr},\lambda,\mu}-\sum_{j=1}^m\frac{|D^{tr}_j|}{|D^{tr}|}f_{D_j,\lambda}\right\|_\rho \le \sum_{j=1}^m\frac{|D^{tr}_j|}{|D^{tr}|}\left[\mu^r\left\|h_\rho\right\|_\rho\left(\left(\mathcal{Q}_{D^{tr}_j,\mu}\mathcal{Q}^*_{D^{tr}_j,\mu}+1\right)\mathcal{Q}^{2r}_{\Xi_n,\mu}+\mathcal{Q}^{2r}_{D^{tr}_j,\mu}\right)\right.\\ &\left.+\,\mu^{1/2}\left(\lambda^{-1/2}\mathcal{Q}^2_{D^{tr}_j,\lambda}\mathcal{P}_{D^{tr}_j,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D^{tr}_j,\lambda}\left\|h\right\|_\rho\right)\left(\mathcal{Q}_{D^{tr}_j,\mu}+\left(\mathcal{Q}_{D^{tr}_j,\mu}\mathcal{Q}^*_{D^{tr}_j,\mu}+1\right)\mathcal{Q}_{\Xi_n,\mu}\right)\right].\end{aligned}$$

Proof of Proposition 17. It follows from (5) and Jensen's inequality that

$$\left\|f^{global}_{D^{tr},\lambda,\mu}-\sum_{j=1}^m\frac{|D^{tr}_j|}{|D^{tr}|}f_{D_j,\lambda}\right\|_\rho = \left\|\sum_{j=1}^m\frac{|D^{tr}_j|}{|D^{tr}|}\left(f^{loc}_{D^{tr}_j,\lambda,\mu}-f_{D_j,\lambda}\right)\right\|_\rho \le \sum_{j=1}^m\frac{|D^{tr}_j|}{|D^{tr}|}\left\|f^{loc}_{D^{tr}_j,\lambda,\mu}-f_{D_j,\lambda}\right\|_\rho.$$

But Proposition 14 with $D=D^{tr}_j$ for $j=1,\dots,m$ yields

$$\begin{aligned}&\left\|f^{loc}_{D^{tr}_j,\lambda,\mu}-f_{D^{tr}_j,\lambda}\right\|_\rho \le \mu^r\left\|h_\rho\right\|_\rho\left(\left(\mathcal{Q}_{D^{tr}_j,\mu}\mathcal{Q}^*_{D^{tr}_j,\mu}+1\right)\mathcal{Q}^{2r}_{\Xi_n,\mu}+\mathcal{Q}^{2r}_{D^{tr}_j,\mu}\right)\\ +\ \ &\mu^{1/2}\left(\lambda^{-1/2}\mathcal{Q}^2_{D^{tr}_j,\lambda}\mathcal{P}_{D^{tr}_j,\lambda}+\lambda^{r-1/2}\mathcal{Q}^{2r-1}_{D^{tr}_j,\lambda}\|h\|_\rho\right)\left(\mathcal{Q}_{D^{tr}_j,\mu}+\left(\mathcal{Q}_{D^{tr}_j,\mu}\mathcal{Q}^*_{D^{tr}_j,\mu}+1\right)\mathcal{Q}_{\Xi_n,\mu}\right).\end{aligned}$$

Combining the above two estimates proves Proposition 17. ■

6.7 Proof of Theorem 1 and Corollary 2

To prove Theorem 1, it suffices to bound $E\left[\mathcal{Q}^{4r}_{D_j,\lambda}\right]$ and $E\left[\mathcal{Q}^4_{D_j,\lambda_j}\mathcal{P}^2_{D_j,\lambda_j}\right]$, which requires several auxiliary lemmas. The first one focuses on bounding $\mathcal{P}_{D,\lambda}$ derived in (Caponnetto and De Vito, 2007; Lin et al., 2017).

Lemma 18 *Let D be a set of samples drawn i.i.d. according to ρ and $0<\delta<1$. Under Assumption 1, with confidence at least $1-\delta$, there holds*

$$\mathcal{P}_{D,\lambda} \ \le\ 2M(\kappa+1)\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)\log\frac{2}{\delta}. \tag{53}$$

The second one aims to bound

$$\mathcal{R}_{D,\lambda} := \left\|(L_K+\lambda I)^{-1/2}(L_K - L_{K,D})(L_K+\lambda I)^{-1/2}\right\| \tag{54}$$

that can be deduced from (Lin et al., 2021, Lemma 6).

Lemma 19 *Let D be a set of samples drawn i.i.d. according to ρ. If $\frac{1}{|D|} < \lambda \leq 1$ and $\mathcal{N}(\lambda) \geq 1$, then*

$$P\left[\mathcal{R}_{D,\lambda} \geq \frac{1}{4}\right] \leq 4\exp\left\{-\frac{\sqrt{\lambda|D|}}{C_1^*(1+\log\mathcal{N}(\lambda))}\right\}, \tag{55}$$

where $C_1^* := 4\max\{(\kappa^2+1)/3, 2\sqrt{\kappa^2+1}\}$.

Due to the definition of $\mathcal{R}_{D,\lambda}$, the bounds of $\mathcal{Q}_{D,\lambda}$ and $\mathcal{Q}^*_{D,\lambda}$ can be given in the following lemma, whose proof is rather standard.

Lemma 20 *If $\mathcal{R}_{D,\lambda} \leq 1/4$, there holds*

$$\mathcal{Q}_{D,\lambda} \leq \frac{2\sqrt{3}}{3} \tag{56}$$

and

$$\mathcal{Q}^*_{D,\lambda} \leq \frac{\sqrt{6}}{2}. \tag{57}$$

Proof of Lemma 20. A direct computation yields

$$\begin{aligned}
& (L_K+\lambda I)^{1/2}(L_{K,D}+\lambda I)^{-1}(L_K+\lambda I)^{1/2} \\
= \; & (L_K+\lambda I)^{1/2}[(L_{K,D}+\lambda I)^{-1} - (L_K+\lambda I)^{-1}](L_K+\lambda I)^{1/2} + I = I \\
+ \; & (L_K+\lambda I)^{-1/2}(L_K - L_{K,D})(L_K+\lambda I)^{-1/2}(L_K+\lambda I)^{1/2}(L_{K,D}+\lambda I)^{-1}(L_K+\lambda I)^{1/2}.
\end{aligned}$$

Thus,

$$\begin{aligned}
& \left\|(L_K+\lambda I)^{1/2}(L_{K,D}+\lambda I)^{-1}(L_K+\lambda I)^{1/2}\right\| \\
\leq \; & 1 + \frac{1}{4}\left\|(L_K+\lambda I)^{1/2}(L_{K,D}+\lambda I)^{-1}(L_K+\lambda I)^{1/2}\right\|.
\end{aligned}$$

This implies

$$\|(L_K+\lambda I)^{1/2}(L_{K,D}+\lambda I)^{-1}(L_K+\lambda I)^{1/2}\| \leq \frac{4}{3}$$

and (56). Then,

$$\left\|(L_{K,D}+\lambda I)^{-1/2}(L_K - L_{K,D})(L_{K,D}+\lambda I)^{-1/2}\right\| \leq \mathcal{Q}^2_{D,\lambda}\mathcal{R}_{D,\lambda} \leq \frac{1}{3}.$$

But

$$\begin{aligned}
& \left\|(L_{K,D}+\lambda I)^{1/2}(L_K+\lambda I)^{-1}(L_{K,D}+\lambda I)^{1/2}\right\| \\
\leq \; & 1 + \frac{1}{3}\left\|(L_{K,D}+\lambda I)^{1/2}(L_K+\lambda I)^{-1}(L_{K,D}+\lambda I)^{1/2}\right\|.
\end{aligned}$$

Therefore, we have

$$\left\|(L_{K,D}+\lambda I)^{1/2}(L_K+\lambda I)^{-1}(L_{K,D}+\lambda I)^{1/2}\right\| \leq \frac{3}{2}.$$

This completes the proof of Lemma 20. ■

For further use, we also need the following probability to expectation formula (Lin et al., 2024).

Lemma 21 *Let $0<\delta<1$, and let $\xi \in \mathbb{R}_+$ be a random variable. If $\xi \leq \mathcal{A}\log^b \frac{c}{\delta}$ holds with confidence $1-\delta$ for some $\mathcal{A}, b, c > 0$, then*

$$E[\xi] \leq c\Gamma(b+1)\mathcal{A},$$

where $\Gamma(\cdot)$ is the Gamma function.

With the help of the above lemmas, we can derive the bounds of $E\left[\mathcal{Q}^4_{D,\lambda}\mathcal{P}^2_{D,\lambda}\right]$ and $E\left[\mathcal{Q}^{4r}_{D,\lambda}\right]$ as follows.

Lemma 22 *Under Assumption 1, if $\lambda|D| \geq (2vC_1^*(\log(1+\kappa)+2))^2\log^4|D|$ for some $v \geq 1$, then*

$$\begin{aligned} E\left[\mathcal{Q}^{2v}_{D,\lambda}\mathcal{P}^v_{D,\lambda}\right] &\leq C_0^*\left(\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)^v + |D|^{-v/2}\right), && (58)\\ E\left[\mathcal{Q}^u_{D,\lambda}\right] &\leq 5(2+\kappa)^{u/2}, \qquad \forall\ 0 \leq u \leq 4v, && (59)\\ E[(\mathcal{Q}^*_{D,\lambda})^u] &\leq 5(4+\kappa)^{u/2}, \qquad \forall\ 0 \leq u \leq 4v, && (60)\end{aligned}$$

where $C_0^* := 2(2(\kappa+1)M)^v \max\{\gamma(v+1)2^v, 2\kappa^v\}$.

Proof of Lemma 22. If $R_{D,\lambda} \leq 1/4$, it follows from Lemma 20 and Lemma 18 that, with confidence $1-\delta$, there holds

$$\mathcal{Q}^{2v}_{D,\lambda}\mathcal{P}^v_{D,\lambda} \leq \left(4M(\kappa+1)\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)\log\frac{2}{\delta}\right)^v.$$

Then Lemma 21 with $\mathcal{A} = \left(4M(\kappa+1)\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)\right)^v$, $b=v$, and $c=2$ implies

$$E\left[\mathcal{Q}^{2v}_{D,\lambda}\mathcal{P}^v_{D,\lambda}|R_{D,\lambda} \leq 1/4\right] \leq 2\Gamma(v+1)\left(4M(\kappa+1)\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)\right)^v. \qquad (61)$$

But $\lambda|D| \geq (2vC_1^*(\log(1+\kappa)+2))^2\log^4|D|$ together with Lemma 19, $\mathcal{N}(\lambda) \leq \kappa\lambda^{-1}$, and $\lambda \geq |D|^{-1}$ yields

$$\begin{aligned} &P\left[R_{D,\lambda} > 1/2\right] \leq 4\exp\left\{-\frac{2vC_1^*(\log(1+\kappa)+2)\log^2|D|}{C_1^*(2+\log((\kappa+1)|D|))}\right\}\\ \leq\ & 4\exp\left\{-2v\log|D|\right\} = 4|D|^{-2v}. \qquad (62)\end{aligned}$$

Therefore, we get from $\mathcal{Q}_{D,\lambda}^{2v}\mathcal{P}_{D,\lambda}^{v} \leq (2(\kappa+1)\kappa M\lambda^{-3/2})^v$ and (62) that

$$\begin{aligned}
& E\left[\mathcal{Q}_{D,\lambda}^{2v}\mathcal{P}_{D,\lambda}^{v}\right] = E\left[\mathcal{Q}_{D,\lambda_j}^{4}\mathcal{P}_{D,\lambda_j}^{2}|R_{D,\lambda}\leq 1/4\right]P[R_{D,\lambda}\leq 1/4] \\
+ \quad & E\left[\mathcal{Q}_{D,\lambda_j}^{4}\mathcal{P}_{D,\lambda_j}^{2}|R_{D,\lambda}> 1/4\right]P[R_{D,\lambda}> 1/4] \\
\leq \quad & 2\Gamma(v+1)\left(4M(\kappa+1)\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)\right)^v + 4(2(\kappa+1)\kappa M\lambda^{-3/2})^v|D|^{-2v} \\
\leq \quad & 2(2(\kappa+1)M)^v\max\{\Gamma(v+1)2^v, 2\kappa^v\}\left(\left(\frac{1}{|D|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D|}}\right)^v + |D|^{-v/2}\right).
\end{aligned}$$

This proves (58). Noting further

$$\mathcal{Q}_{D,\lambda} \leq \sqrt{1+\kappa}\lambda^{-1/2},$$

we have from $0\leq u\leq 4v$, Lemma 20, and (62) that

$$\begin{aligned}
E\left[\mathcal{Q}_{D,\lambda}^{u}\right] &\leq E\left[\mathcal{Q}_{D,\lambda}^{u}|R_{D,\lambda}>1/4\right]P[R_{D,\lambda}>1/4] + E\left[\mathcal{Q}_{D,\lambda}^{u}|R_{D,\lambda}\leq 1/4\right]P[R_{D,\lambda}\leq 1/4] \\
&\leq 4(1+\kappa)^{u/2}|D|^{u/2-2v} + 2^{u/2} \leq 5(2+\kappa)^{u/2}.
\end{aligned}$$

The bound of (60) can be derived in the same way. This completes the proof of Lemma 22. ■

We then turn to prove Theorem 1.

Proof of Theorem 1. Noting (13) with $C_1 := (4C_1^*(\log(1+\kappa)+2))^2$, we have $\lambda|D_j| \geq (4C_1^*(\log(1+\kappa)+2))^2\log^4|D_j|$ for all $j=1,\dots,m$. Then, it follows from Lemma 22 with $v=2$, $u=4r$, (13), $r\geq 1/2$, and (14) that

$$\begin{aligned}
& \sum_{j=1}^{m}\frac{|D_j|}{|D|}\lambda_j^{2r}\|h_\rho\|_\rho^2 E\left[\mathcal{Q}_{D_j,\lambda_j}^{2r}\right] \\
\leq \quad & C_2^*\left(\sum_{j:|D_j|\geq|D|^{\frac{1}{2r+s}}\log^4|D|}\frac{|D_j|}{|D|}|D|^{-\frac{2r}{2r+s}} + \sum_{j:|D_j|\leq|D|^{\frac{1}{2r+s}}\log^4|D|}\frac{|D_j|}{|D|}|D_j|^{-2r}\log^{8r}|D|\right) \\
\leq \quad & 2C_2^*|D|^{-\frac{2r}{2r+s}}, \qquad (63)
\end{aligned}$$

where $C_2^* := 5(2+\kappa)^{2r}(4C_1^*(\log(1+\kappa)+2))^{4r}\|h_\rho\|_\rho^2$. Furthermore, it follows from Lemma 22 with $v=2$, Assumption 3, and (13) that

$$\begin{aligned}
& \sum_{j=1}^m \frac{|D_j|^2}{|D|^2} E\left[\mathcal{Q}_{D_j,\lambda_j}^4 \mathcal{P}_{D_j,\lambda_j}^2\right] \\
\leq \ & 80M^2(\kappa+1)^4 \sum_{j=1}^m \frac{|D_j|^2}{|D|^2}\left(\left(\frac{1}{|D_j|\sqrt{\lambda_j}}+\sqrt{\frac{\mathcal{N}(\lambda_j)}{|D_j|}}\right)^2+|D_j|^{-1}\right) \\
\leq \ & 160M^2(\kappa+1)^4|D|^{-1}\sum_{j=1}^m\left(2|D|^{-1}\lambda_j^{-1}+\frac{|D_j|}{|D|}\mathcal{N}(\lambda_j)\right) \\
\leq \ & 160(C_0+2)M^2(\kappa+1)^4|D|^{-1}\sum_{j=1}^m\left(|D|^{-1}\lambda_j^{-1}+\frac{|D_j|}{|D|}\lambda_j^{-s}\right).
\end{aligned}$$

But (13) and (14) yield

$$\begin{aligned}
& \sum_{j=1}^m \left(\lambda_j^{-1}+\frac{|D_j|}{|D|}\lambda_j^{-s}\right) \\
\leq \ & \sum_{j:|D_j|\geq |D|^{\frac{1}{2r+s}}\log^4|D|}\left(|D|^{-1}\lambda_j^{-1}+\frac{|D_j|}{|D|}\lambda_j^{-s}\right) \\
+ \ & \sum_{j:|D_j|< |D|^{\frac{1}{2r+s}}\log^4|D|}\left(|D|^{-1}\lambda_j^{-1}+\frac{|D_j|}{|D|}\lambda_j^{-s}\right) \\
\leq \ & (4C_1^*(\log(1+\kappa)+2))^{-2s}\left(m|D|^{\frac{-2r-s+1}{2r+s}}+|D|^{\frac{s}{2r+s}}+\log^{-4}|D|+|D|^{\frac{s}{2r+s}}\right) \\
\leq \ & 4(4C_1^*(\log(1+\kappa)+2))^{-2s}|D|^{\frac{s}{2r+s}}.
\end{aligned}$$

Thus, we have

$$\sum_{j=1}^m \frac{|D_j|^2}{|D|^2} E\left[\mathcal{Q}_{D_j,\lambda_j}^4 \mathcal{P}_{D_j,\lambda_j}^2\right] \leq C_3^*|D|^{-\frac{2r}{2r+s}}, \tag{64}$$

where $C_3^* := 640(C_0+2)M^2(\kappa+1)^4(4C_1^*(\log(1+\kappa)+2))^{-2s}$. Plugging (63) and (64) into Proposition 11, we complete the proof of Theorem 1 with $C_2 = 4C_2^*+2C_3^*$. ■

The proof of Corollary 2 is almost the same as above. We present it for the sake of completeness.

Proof of Corollary 2. Setting $C_1 := (4C_1^*(\log(1+\kappa)+2))^2$, we have $\lambda|D_j| \geq (4C_1^*(\log(1+\kappa)+2))^2\log^4|D_j|$ for all $j=1,\dots,m$. Then, it follows from Lemma 22 with $u=4r$ and $v=2$ that

$$\sum_{j=1}^m \frac{|D_j|}{|D|}\lambda_j^{2r}\|h_\rho\|_\rho^2 E\left[\mathcal{Q}_{D_j,\lambda}^{2r}\right] \leq C_2^*\sum_{j=1}^m\frac{|D_j|}{|D|}|D|^{-\frac{2r}{2r+s}} = C_2^*|D|^{-\frac{2r}{2r+s}}.$$

Furthermore, it follows from Lemma 22 with $v = 2$, Assumption 3, and (16) that

$$\begin{aligned}
&\sum_{j=1}^{m}\frac{|D_j|^2}{|D|^2}E\left[\mathcal{Q}^4_{D_j,\lambda_j}\mathcal{P}^2_{D_j,\lambda_j}\right]\\
\leq\ & 160(C_0+2)M^2(\kappa+1)^4|D|^{-1}\sum_{j=1}^{m}\left(|D|^{-1}\lambda_j^{-1}+\frac{|D_j|}{|D|}\lambda_j^{-s}\right)\\
\leq\ & 160(C_0+2)M^2(\kappa+1)^4(4C_1^*(\log(1+\kappa)+2))^{-2s}|D|^{-1}\left(m|D|^{\frac{-2r-s+1}{2r+s}}+|D|^{\frac{s}{2r+s}}\right)\\
\leq\ & C_3^*|D|^{-\frac{2r}{2r+s}}.
\end{aligned}$$

Plugging the above two estimates into Proposition 11 completes the proof of Corollary 2. ■

6.8 Proof of Theorem 3

In this part, we use Theorem 1 and Proposition 12 to prove Theorem 3. Firstly, we present a detailed bound of the global approximation as follows.

Proposition 23 *Under Assumption 1 and Assumption 2 with $1/2 \leq r \leq 1$, if ρ_X is a uniform distribution, $\mu \geq (8C_1^*(\log(1+\kappa)+2))^2\max_{j=1,\dots,m}\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|}$, Ξ_n satisfies (8) for some $c, \beta > 0$, and n satisfies $\mu n^\beta \geq 2c$, then for any $\lambda \in (\mu, 1)$, there holds*

$$\begin{aligned}
&E\left[\left\|f^{global}_{D^{tr},\lambda,\mu}-\sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda}\right\|_\rho^2\right]\\
\leq\ & C_4\mu^{2r}+C_4\mu\left(\lambda^{2r-1}+\sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}\left(\left(\frac{1}{|D_j^{tr}|\lambda}+\sqrt{\frac{\mathcal{N}(\lambda)}{\lambda|D_j^{tr}|}}\right)^2+|D_j^{tr}|^{-1}\right)\right),
\end{aligned}$$

where C_4 is a constant depending only on $\|h_\rho\|_\rho, M, r, c,$ and β.

To prove the above proposition, we need the following preliminary lemma.

Lemma 24 *If ρ_X is a uniform distribution, Ξ_n satisfies (8) for some $c, \beta > 0$, and n satisfies $\mu n^\beta \geq 2c$, then*

$$\mathcal{Q}_{\Xi_n,\mu} \leq \sqrt{2}. \tag{65}$$

Proof of Lemma 24. From the definition of the operator norm $\|\cdot\|$ and the fact $\|f\|_K = \sup_{\|g\|_K\leq 1}\langle f, g\rangle_K$, we obtain

$$\begin{aligned}
\|L_K - L_{K,\Xi_n}\| = &\sup_{\|f\|_K\leq 1}\|((L_{K,\Xi_n}-L_K)f\|_K = \sup_{\|f\|_K\leq 1}\sup_{\|g\|_K\leq 1}\langle(L_{K,\Xi_n}-L_K)f, g\rangle_K \\
= &\sup_{\|g\|_K\leq 1,\|f\|_K\leq 1}\left|\left\langle\int_{\mathcal{X}} f(x')K_{x'}d\rho_X - \frac{1}{n}\sum_{k=1}^n f(\xi_k)K_{\xi_k}, g\right\rangle_K\right| \\
= &\sup_{\|g\|_K\leq 1,\|f\|_K\leq 1}\left|\int_{\mathcal{X}} f(x')\langle K_{x'}, g\rangle_K d\rho_X - \frac{1}{n}\sum_{k=1}^n f(\xi_k)\langle K_{\xi_k}, g\rangle_K\right| \\
= &\sup_{\|g\|_K\leq 1,\|f\|_K\leq 1}\left|\int_{\mathcal{X}} f(x')g(x')d\rho_X - \frac{1}{n}\sum_{k=1}^n f(\xi_k)g(x_i)\right|.
\end{aligned}$$

Then, it follows from (8) that

$$\|L_{K,\Xi_n} - L_K\| \leq cn^{-\beta}.$$

Therefore,

$$\|(L_K+\mu I)^{-1/2}(L_K - L_{K,\Xi_n})(L_K+\mu I)^{-1/2}\| \leq c\mu^{-1}n^{-\beta}.$$

Then it follows from $\mu n^\beta \geq 2c$ that

$$\|(L_K+\mu I)^{-1/2}(L_K - L_{K,\Xi_n})(L_K+\mu I)^{-1/2}\| \leq 1/2.$$

Noting further

$$\begin{aligned}
\mathcal{Q}_{D,\Xi_n}^2 = &\left\|(L_K+\mu I)^{1/2}(L_{K,\Xi_n}+\mu I)^{-1}(L_K+\mu I)^{1/2}\right\| \\
\leq &\left\|(L_K+\mu I)^{1/2}\left[(L_{K,\Xi_n}+\mu I)^{-1} - (L_K+\mu I)^{-1}\right](L_K+\mu I)^{1/2}\right\| + 1 \\
\leq &\frac{1}{2}\mathcal{Q}_{D,\Xi_n}^2 + 1,
\end{aligned}$$

we then obtain $\mathcal{Q}_{D,\Xi_n} \leq \sqrt{2}$. This completes the proof of Lemma 24. ■

With the help of the above two lemmas and Proposition 17, we are in a position to prove Proposition 23 as follows.

Proof of Proposition 23. Due to Proposition 17, Hölder's inequality, Jensen's inequality, and the basic inequalities $(a+b)^2 \leq 2a^2+2b^2$ and $(a+b)^4 \leq 8a^4+8b^4$ for $a,b \geq 0$, we have

$$\begin{aligned}
& E\left[\left\|f^{global}_{D^{tr},\lambda,\mu} - \sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}f_{D_j,\lambda}\right\|_{\rho}^{2}\right] \\
\leq\ & 4\mu^{2r}\|h_\rho\|_\rho^2\sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}E\left[\left(\mathcal{Q}_{D_j^{tr},\mu}\mathcal{Q}^*_{D_j^{tr},\mu}+1\right)^2\mathcal{Q}^{4r}_{\Xi_n,\mu}\right] + 4\mu^{2r}\|h_\rho\|_\rho^2\sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}E\left[\mathcal{Q}^{4r}_{D_j^{tr},\mu}\right] \\
+\ & 16\mu\sum_{j=1}^{m}\frac{|D_j^{tr}|}{|D^{tr}|}\left(E\left[\lambda^{-2}\mathcal{Q}^8_{D_j^{tr},\lambda}\mathcal{P}^4_{D_j^{tr},\lambda}+\lambda^{4r-2}\mathcal{Q}^{8r-4}_{D_j^{tr},\lambda}\|h\|_\rho^4\right]\right)^{1/2} \\
\times\ & \left(E\left[\mathcal{Q}^4_{D_j^{tr},\mu}+\left(\mathcal{Q}_{D_j^{tr},\mu}\mathcal{Q}^*_{D_j^{tr},\mu}+1\right)^4\mathcal{Q}^4_{\Xi_n,\mu}\right]\right)^{1/2}.
\end{aligned}$$

Using (59), (60), and Lemma 24 with $u=4$ and $v=4$, we get from Hölder's inequality and $\mu \geq (8C_1^*(\log(1+\kappa)+2))^2\max_{j=1,\ldots,m}\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|}$ that

$$E\left[\left(\mathcal{Q}_{D_j^{tr},\mu}\mathcal{Q}^*_{D_j^{tr},\mu}+1\right)^2\mathcal{Q}^{4r}_{\Xi_n,\mu}\right] \leq 2^{2r+1}(1+5(2+\kappa)(4+\kappa)).$$

Moreover, it follows from (59) with $u=4r$ and $v=4$ that

$$E\left[\mathcal{Q}^{4r}_{D_j^{tr},\mu}\right] \leq 5(2+\kappa)^{2r}.$$

Using (58) with $v=4$ and (59) with $u=8r-4$ and $v=4$, we obtain from $\lambda \geq (8C_1^*(\log(1+\kappa)+2))^2\max_{j=1,\ldots,m}\frac{\log^4|D_j^{tr}|}{|D_j^{tr}|}$ that

$$\begin{aligned}
& \left(E\left[\lambda^{-2}\mathcal{Q}^8_{D_j^{tr},\lambda}\mathcal{P}^4_{D_j^{tr},\lambda}+\lambda^{4r-2}\mathcal{Q}^{8r-4}_{D_j^{tr},\lambda}\|h\|_\rho^4\right]\right)^{1/2} \\
\leq\ & \lambda^{-1}(C_0^*)^{1/2}\left(\left(\frac{1}{|D_j^{tr}|\sqrt{\lambda}}+\sqrt{\frac{\mathcal{N}(\lambda)}{|D_j^{tr}|}}\right)^2+|D_j^{tr}|^{-1}\right)+\lambda^{2r-1}\|h\|_\rho^2\sqrt{5}(2+\kappa)^{2r-1}.
\end{aligned}$$

Furthermore, Lemma 22 with $u=4$ and Lemma 24 yield

$$\begin{aligned}
& \left(E\left[\mathcal{Q}^4_{D_j^{tr},\mu}+\left(\mathcal{Q}_{D_j^{tr},\mu}\mathcal{Q}^*_{D_j^{tr},\mu}+1\right)^4\mathcal{Q}^4_{\Xi_n,\mu}\right]\right)^{1/2} \\
\leq\ & \left(E\left[\mathcal{Q}^4_{D_j^{tr},\mu}\right]\right)^{1/2}+4\sqrt{2}\left(1+\left(E\left[\mathcal{Q}^8_{D_j^{tr},\mu}\right]\right)^{1/2}\left(E\left[\left(\mathcal{Q}^*_{D_j^{tr},\mu}\right)^8\right]\right)^{1/2}\right)^{1/2} \\
\leq\ & \sqrt{5}(2+\kappa)+4\sqrt{2}+4\sqrt{2}4\sqrt{10}(2+\kappa)(4+\kappa).
\end{aligned}$$

Combining all the above estimates, we have from $\mu \geq (8C_1^*(\log(1{+}\kappa){+}2))^2 \max_{j=1,\dots,m} \frac{\log^4 |D_j^{tr}|}{|D_j^{tr}|}$ that

$$
\begin{aligned}
& E\left[\left\|f^{global}_{D^{tr},\lambda,\mu} - \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} f_{D_j,\lambda}\right\|_\rho^2\right] \leq \mu^{2r}\|h_\rho\|_\rho^2 \left(2^{2r+3}(1+5(2+\kappa)(4+\kappa)) + 20(2+\kappa)^{2r}\right) \\
+ \quad & 16\mu\left(\sqrt{5}(2+\kappa) + 4\sqrt{2} + 4\sqrt{2}4\sqrt{10}(2+\kappa)(4+\kappa)\right) \\
\times \quad & \left(\lambda^{-1}\left(C_0^*\right)^{1/2} \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}\left(\left(\frac{1}{|D_j^{tr}|\sqrt{\lambda}} + \sqrt{\frac{\mathcal{N}(\lambda)}{|D_j^{tr}|}}\right)^2 + |D_j^{tr}|^{-1}\right) + \lambda^{2r-1}\|h\|_\rho^2 \sqrt{5}(2+\kappa)^{2r-1}\right) \\
\leq \quad & C_4\mu^{2r} + C_4\mu\left(\lambda^{2r-1} + \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}\left(\left(\frac{1}{|D_j^{tr}|\lambda} + \sqrt{\frac{\mathcal{N}(\lambda)}{\lambda|D_j^{tr}|}}\right)^2 + |D_j^{tr}|^{-1}\right)\right),
\end{aligned}
$$

where C_4 is a constant depending only on κ, M, r, C_0^*, and $\|h_\rho\|_\rho$. This completes the proof of Proposition 23. ■

Finally, we prove Theorem 3.

Proof of Theorem 3. Let $\bar{\lambda} := C_1|D|^{-\frac{1}{2r+s}}$. We have from $|D_j| \geq |D|^{\frac{1}{2r+s}} \log^4 |D|$ and Assumption 3 that

$$
\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|}\left(\left(\frac{1}{|D_j^{tr}|\bar{\lambda}} + \sqrt{\frac{\mathcal{N}(\bar{\lambda})}{\bar{\lambda}|D_j^{tr}|}}\right)^2 + |D_j^{tr}|^{-1}\right) \leq \tilde{C}_2 m|D|^{-\frac{2r-1}{2r+s}},
$$

where $\tilde{C}_2$ is a constant depending only on C_0 and C_1. Then it follows from Proposition 23 that, under (18), there holds

$$
E\left[\left\|f^{global}_{D^{tr},\bar{\lambda},\mu} - \sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} f_{D_j,\lambda}\right\|_\rho^2\right] \leq \tilde{C}_3\left(\mu^{2r} + \mu|D|^{-\frac{2r-1}{2r+s}}\right),
$$

where $\tilde{C}_3$ is a constant depending only on C_4, C_1, and C_0. Furthermore, it follows from Corollary 2 that, under (18), there holds

$$
E\left[\left\|\sum_{j=1}^m \frac{|D_j^{tr}|}{|D^{tr}|} f_{D_j,\bar{\lambda}} - f_\rho\right\|_\rho^2\right] \leq C_2|D|^{-\frac{2r}{2r+s}}.
$$

Plugging the above two estimates into Proposition 12 and noting $\mu \leq |D|^{-\frac{1}{2r+s}}$, we have

$$
E\left[\left\|\overline{f}^{Ada}_{D,\vec{\lambda}^*} - f_\rho\right\|_\rho^2\right] \leq C_5\left(m\frac{\log|\Lambda|}{|D|} + |D|^{-\frac{2r}{2r+s}}\right),
$$

where C_5 is a constant depending only on $\tilde{C}_2$, C_2, and C_1'. This completes the proof of Theorem 3. ■