New Nonlinear Machine Learning Algorithms With Theoretical Analysis

by

Guodong Liu

BS, Shanghai Jiao Tong University, 2011

Submitted to the Graduate Faculty of

the Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2023

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Guodong Liu

It was defended on

June 27th 2023

and approved by

Liang Zhan, PhD, Associate Professor, Department of Electrical and Computer Engineering

Zhi-Hong Mao, PhD, Professor, Department of Electrical and Computer Engineering

Wei Gao, PhD, Associate Professor, Department of Electrical and Computer Engineering

Wei Chen, PhD, Associate Professor, The School of Medicine, Department of Pediatrics

Dissertation Director: Heng Huang, PhD, John A. Jurenko Endowed Professor,

Department of Electrical and Computer Engineering

New Nonlinear Machine Learning Algorithms With Theoretical Analysis

Guodong Liu, PhD

University of Pittsburgh, 2023

Recent advances in machine learning have spawned progress in various fields. In the context of machine learning data, nonlinearity is an intrinsic property. Therefore, a nonlinear model will facilitate the flexibility of representation and fit the data properly. However, increasing flexibility usually means the higher complexity and less interpretability. Thus, there is a niche for designing feasible nonlinear machine learning models to handle the forementioned challenges.

As a part of this work, a new method, called as sparse shrunk additive models (SSAM) is proposed. This model explores the structure information among features for high-dimensional nonparametric regression with the allowance of the flexible interactions among features. It bridges the sparse kernel regression and sparse feature selection. Theoretical results on the convergence rate and sparsity characteristics are established by the novel analysis techniques with integral operator and concentration estimate.

Most of the nonlinear models usually involve tuning multiple (up to thousands) hyperparameters, which plays a pivotal role in model generalization. Another part of this work is a new hyperparameter optimization method with zeroth-order hyper-gradients (HOZOG). We proved the feasibility analysis of using HOZOG to achieve hyperparameter optimization under the condition of Lipschitz continuity. The extensive experiments verify the analysis.

For large-scale data, there remain computational challenges in implementing various algorithms. To address this issue, we propose a new regularized modal regression model with robust sampling strategy. Unlike conventional sampling for large-scale least squares, our sampling probabilities are dependent on the robust loss function for learning the conditional mode. We provide theoretical analysis to support the proposed model: the approximation bound is established by error analysis with Rademacher complexity, and the robustness characterization is provided based on the finite sample breakdown point analysis. Experiments on both synthetic and real-world data show promising performance of the proposed estimator.

Table of Contents

List of Tables

List of Figures

Preface

I would like to express my heartfelt gratitude to my Ph.D. advisor, Professor Heng Huang, for his exceptional guidance throughout my eight-year journey. Working under his supervision has been a privilege, and I am immensely grateful for the opportunity to conduct research with him.

Professor Heng Huang's unwavering dedication and passion for research have been truly inspiring. Over the years, he has provided me with visionary advice and unwavering support. I have learned valuable lessons from him on problem discovery, problem definition, and problem-solving. Starting and pursuing my research career under Professor Heng Huang's supervision has been a stroke of great fortune.

I would also like to extend my gratitude to the members of my Ph.D. committee: Professor Zhi-Hong Mao, Professor Wei Gao, Professor Liang Zhan, and Professor Wei Chen. Their valuable advice and guidance on my research direction have been greatly appreciated. I am honored to have received their insightful instructions and I am grateful for the time and assistance they have provided.

I would like to thank every member of the Pitt Data Science Lab for their contributions. Special thanks to Prof.Hong Chen, Dr.Feiping Nie, Prof.Bin Gu, for their guidance, insightful discussions, and collaboration during my Ph.D. studies. I am grateful to my friends and lab mates, including An, De, Zhouyuan, Haoteng, Jie, Xiaoqian, Hongchang, Kamran, Yanfu, Runxue, Wenhan, and Shangqian. Being a part of such a wonderful research group has been an honor, and it has been a pleasure working together with all of you.

Lastly, I would like to extend special thanks to my family, particularly my parents, grandparents, and wife and daughter. Their love, support, and upbringing have been instrumental in my achievements. I am grateful for their understanding and unwavering love throughout these years. It is because of their encouragement that I have been able to overcome obstacles and stay on the right path. Their presence has been my guiding light, and I am truly grateful for their constant support.

1.0 Introduction

1.1 Background

Recent advances in machine learning have spawned progress in various fields such as medical diagnosis, information extraction and financial forecasting. These applications are generating vast amounts of data. These data can be highly complicated and heterogeneous. What we would like to do is to learn from data: to extract important patterns and trends and understand "what does data say".

1.1.1 Linear Models

In Machine learning community, liner model must be the most common used and well-developed tool to understand the world of data. It assumes that the underlying function of data presentation is linear in the input. Liner models have been studied since the very early stage (before the computer stage) of statistics, and they still play an important role in today's computer era [41]. They are simple and able to provide an adequate and interpretable explanation of how the input actually effect the output. However, linear models could perform competitive only when the scale of data is small and with a low signal noise ratio or sparse assumption [14].

Under the setting of nowadays machine learning application, the amount of data collected in a wide array of scientific domains is dramatically increasing in both size and complexity, and the relationship between the input and output are highly possible to be not linear. These complicated data dose not satisfy the forementioned assumptions anymore. While nonlinear models can facilitate the flexibility of representation and fit the data properly [82]. Following the well-known "no free lunch" principle [101], increased flexibility usually means higher complexity and less interpretability. Thus, there is a niche for designing feasible nonlinear machine learning models to handle these challenges.

1.1.2 Sparse Additive Models

Additive models [74, 44, 73, 109, 108, 13, 59] provide a feasible extension of linear models, making them more flexible while still remaining the treasured property of interpretability. Beyond that, the familiar tools for inference in linear models are also available for additive models. Among additive models, sparse additive models [109] have shown good performance in combining variable selection with regression and classification tasks due to their additive hypothesis function spaces and sparse regularization. In most of these sparse models, the interactions between features are often ignored or just discussed under prior structure information, there is a blank of how to find the interaction among correlated features automatically, and This is one main focus of this work.

On the other hand, since more and more complicated nonlinear models are prosed, and these new machine learning algorithms usually involve tuning multiple (from one to thousands) hyperparameters. These hyperparameters usually play a pivotal role in terms of model generalizability.

1.1.3 Hyperparameters Optimization

Classic hyperparameters optimization techniques such as grid search [37, 58] and random search [7] have a very restricted application in modern hyperparameter optimization tasks, because they only can manage a very small number of hyperparameters and cannot guarantee to converge to local/global minima.

modern hyperparameter tuning tasks, black-box optimization [24] and gradient-based algorithms [30] are currently the dominant approaches due to the advantages in terms of effectiveness, efficiency, scalability, simplicity and flexibility. How to design a new hyperparameter optimization technique [38] inheriting all benefits from both approaches is still an open problem.

1.1.4 Modal Regression

Modal regression [80, 17, 53] has attracted much attention in statistical machine learning research, because the resulting estimator is more efficient and robust than ordinary least square-based estimation in the case of outliers or heavy-tail error distribution. Unlike conventional regression for learning conditional mean or median, modal regression focuses on estimating the conditional mode of a response Y given input $X = x$ [107, 106]. The mode can better reveal numerical characteristic of a statistical distribution or data set, which is usually missed by the traditional mean for data with outliers or the skewed noise distribution [15].

1.2 Contribution

We summarize our contribution as follows:

- A sparse shrunk additive algorithm is proposed to improve the feature selection ability of nonlinear models. It is a uniform framework to bridge sparse feature selection, sparse sample selection, and feature interaction structure learning tasks. SSAM can be implemented efficiently and its effectiveness is supported by the empirical studies.
- Generalization bound on the excess risk is provided for SSAM under mild conditions, which implies the fast convergence rate can be achieved. Additionally, the necessary and sufficient condition is derived to characterize the sparsity of SSAM.
- A zeroth-order gradient algorithm to solve the problem of hyperparameters optimization is proposed. This is the first method having the benefits of Effectiveness, efficiency, scalability, simplicity and flexibility. We provide an upper bound to the Lipschitz constant of the A-based constrained optimization problem which theoretically guarantees.
- A fast modal regression with robust sampling method is proposed and we establish its asymptotic and robust analysis on function estimation. The current results fills the gap of modal regression for large-scale data computation and extends the gradient-based sampling from the conditional mean regression to the mode setting.

1.3 Thesis Organization

The rest of the thesis is organized as follows. In Chapter 2, we propose a sparse shrunk additive algorithm to look into the relations between data variables. Generalization bound on the excess risk is provided for SSAM under mild conditions, which implies the fast convergence rate can be achieved. In Chapter 3, we propose a zeroth-order gradient algorithm to solve the problem of hyperparameters optimizationan and an upper bound to the Lipschitz constant of the A-based constrained optimization problem which theoretically guarantees. In Chapter 4, we propose a fast modal regression with robust sampling, and establish its asymptotic and robust analysis on function estimation.

Finally, we conclude the thesis in Chapter 5.

2.0 Sparse Shrunk Additive Models

Most existing feature selection methods in literature are linear models, so that the nonlinear relations between features and response variables are not considered. Meanwhile, in these feature selection models, the interactions between features are often ignored or just discussed under prior structure information. To address these challenging issues, we consider the problem of sparse additive models for high-dimensional nonparametric regression with the allowance of the flexible interactions between features. A new method, called as sparse shrunk additive models (SSAM), is proposed to explore the structure information among features. This method bridges sparse kernel regression and sparse feature selection. Theoretical results on the convergence rate and sparsity characteristics of SSAM are established by the novel analysis techniques with integral operator and concentration estimate. In particular, our algorithm and theoretical analysis only require the component functions to be continuous and bounded, which are not necessary to be in reproducing kernel Hilbert spaces. Experiments on both synthetic and real-world data demonstrate the effectiveness of the proposed approach.

2.1 Introduction

Sparse feature selection has attracted much attention in machine learning community for learning tasks with high-dimensional data, especially useful in bioinformatics related applications. Linear models with ℓ_1-norm regularization, such as Lasso [93] and Dantzig selector[11], have been well studied for their theoretical properties and extensively used for feature selection applications. However, in many applications, the linear assumption could be too restricted to select the optimal features, because the relations between features and response variables could be nonlinear. Because of the difficulties in both computational algorithm and learning theory analysis, only few of existing feature selection methods in literature focus on the nonlinear feature selection.

To enhance the ability of feature selection models with considering nonlinear relationship between features and response variables, several sparse learning based additive models were proposed for regression [74, 44, 73, 109, 108, 13] and classification [110, 12], which are extensions of original additive models [40]. Note that, in these additive models, each component function is a univariate smooth function [74, 44, 73, 109, 110] or is defined on grouped features with prior structure information [12, 108]. Although these sparse additive models can conduct nonlinear feature selection, all of them do not explore the important feature interaction without prior structure information. Recently, the shrunk additive least square approximation (SALSA) [49] method was introduced to utilizing the feature interactions, but without feature selection mechanism.

On the other hand, the sparse sample selection arises from learning tasks with large-scale data. The generalized Lasso was proposed in [76] to handle the regression problem with addressing sample sparsity, and its learning theory has been studied in [85]. Recently, Nyström approximation has been used for selecting important samples (landmark points) in kernel methods, which show that the predictor can be derived efficiently from data dependent hypothesis spaces associated with subsamples [52, 1, 77]. While some fast algorithms have been developed for sparse kernel regression, none of them is capable of the feature selection and provides the interpretability of prediction.

To address the above challenges, in this paper, we propose a novel *sparse shrunk additive model* (SSAM) for jointly selecting features and samples with learning the feature interactions and mining the structure information among features. Different to previous models, our new method will simultaneously conduct sparse feature selection, sparse sample selection, and feature interactions learning. Our SSAM can utilize the component functions from general continuous and bounded function space [91, 14] and can be implemented efficiently via the optimization technique in [65].

More important, to better understand the learning theory properties of SSAM, we investigate its convergence rate and sparsity. The proposed SSAM involves the shrunk structure on features and the ℓ_1-norm regularization on data dependent hypothesis spaces. While these features provide the superior flexibility and adaptivity of SSAM, there are new technical difficulties to characterize its theory properties. To address the new difficulties, we

introduce a novel decomposition on the excess generalization error, and develop the recent approximation techniques with integral operator and concentration estimates with empirical covering numbers. Our main contributions in this paper include:

- A sparse shrunk additive algorithm is proposed to improve the feature selection ability of nonlinear models. It is a uniform framework to bridge sparse feature selection, sparse sample selection, and feature interaction structure learning tasks. SSAM can be implemented efficiently and its effectiveness is supported by the empirical studies.
- Generalization bound on the excess risk is provided for SSAM under mild conditions, which implies the fast convergence rate can be achieved. Additionally, the necessary and sufficient condition is derived to characterize the sparsity of SSAM.

2.2 Sparse Shrunk Additive Models

Let $\mathcal{X} \subset \mathbb{R}^n$ be an explanatory feature space and let $\mathcal{Y} \subset [-1, 1]$ be the response set. Let $\mathbf{z} := \{z_i\}_{i=1}^m = \{(x_i, y_i)\}_{i=1}^m$ be independent copies of a random sample (x, y) following an unknown intrinsic distribution ρ on $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Denote the marginal distribution of ρ on $\mathcal{X}$ as $\rho_{\mathcal{X}}$ and denote the conditional distribution for given $x \in \mathcal{X}$ as $\rho(\cdot|x)$. Given $\mathbf{z}$, the main goal of regression learning is to infer a functional relation between the input $x \in \mathcal{X}$ and the corresponding output $y \in \mathcal{Y}$. Usually, the expected risk associated with least squares loss is used to evaluate the prediction performance, which is denoted by

$$\mathcal{E}(f) = \int_{\mathcal{Z}} (y - f(x))^2 d\rho(x, y).$$

In theory, the minimizer of $\mathcal{E}(f)$ over all measurable functions is the regression function

$$f_\rho(x) = \int_{\mathcal{Y}} y d\rho(y|x).$$

2.2.1 Sparse Additive Models

Additive models [40] aim to find the predictor in the special hypothesis space $\mathcal{F} = \{f : f(X) = \sum_{j=1}^{n} f_j(X_j), X = (X_1, ..., X_n) \in \mathcal{X}\}$. Here, each $f_j \in \mathcal{F}_j$ is one-dimensional smooth function, and its typical examples include the spline function and the Gaussian function. The optimization framework of standard additive model is

$$\min_{f_j} \frac{1}{m} \sum_{i=1}^{m} (y_i - \sum_{j=1}^{n} f_j(x_{ij}))^2. \tag{2-1}$$

Theoretical analysis on (2-1) shows the good performance of additive model relies on the condition that the number of features n is not large relative to the sample size m.

The algorithm of sparse additive models (SpAM) [74] is proposed to address the feature selection in the high dimensional setting, which can be formulated as the following regularized framework

$$\min_{f_j} \Big\{ \frac{1}{m} \sum_{i=1}^{m} (y_i - \sum_{j=1}^{n} f_j(x_{ij}))^2 + \lambda \sum_{j=1}^{n} \|f_j\| \Big\}, \tag{2-2}$$

where $\lambda > 0$ is a regularization parameter and $\sum_{j=1}^{n} \|f_j\|$ behaves liken an ℓ_1 ball across different components to encourage functional sparsity [74, 108]. The SpAM (2-2) can be solved efficiently in terms of the back-fitting algorithm [41], and has been extended to the group sparse additive regression [44, 73, 108].

2.2.2 Shrunk Additive Models

Although SpAM (2-2) has nice properties, it ignores the interactions between features. Recently, a novel method, called *shrunk additive least squares approximation* (SALSA), is proposed in [49] and has shown satisfactory prediction performance.

For any given $1 \leq k \leq n$ and $\{1, 2, ..., n\}$, we denote $d = \binom{n}{k}$ as the number of index subsets with k elements . It is easy to see that $d = n$ as $k = 1$ and $d = \frac{n(n-1)}{2}$ as $k = 2$. Let $x^{(j)} \in \mathbb{R}^k$ be a subset of x with k features and denote its corresponding space as $\mathcal{X}^{(j)}$.

Denote $\mathcal{H}_{K^{(j)}}$ as a reproducing kernel Hilbert space (RKHS) [3, 82, 83] associated with a symmetric and positive definite kernel $K^{(j)} : \mathcal{X}^{(j)} \times \mathcal{X}^{(j)} \to \mathbb{R}, j \in \{1, ..., d\}$. The SALSA is dependent on the hypothesis space with additive kernels, which is defined by:

$$\mathcal{H} = \Big\{ \sum_{j=1}^{d} f^{(j)} : f^{(j)} \in \mathcal{H}_{K^{(j)}}, j = 1, 2, ..., d \Big\}.$$

Indeed, $(\mathcal{H}, \|\cdot\|_K)$ also is an RKHS for $K = \sum_{j=1}^{d} K^{(j)}$, where $\|f\|_K^2 = \inf\{\sum_{j=1}^{d} \|f^{(j)}\|_{K^{(j)}}^2 : f = \sum_{j=1}^{d} f^{(j)}\}$ [73, 16, 109].

Given training samples $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^{m}$, the SALSA in [49] can be formulated as the following optimization problem:

$$\begin{aligned} \tilde{f}_{\mathbf{z}} \;=\; & \underset{f=\sum_{j=1}^{d} f^{(j)} \in \mathcal{H}}{\arg\min} \Big\{ \frac{1}{n} \sum_{i=1}^{m} \Big(y_i - \sum_{j=1}^{d} f^{(j)}(x_i^{(j)}) \Big)^2 \\ & + \eta \sum_{j=1}^{d} \|f^{(j)}\|_{K^{(j)}}^2 \Big\}, \end{aligned} \tag{2-3}$$

where $\eta > 0$ is a regularization parameter.

Remark 6 in [49] tells us that the predictor of SALSA can be expressed as:

$$\tilde{f}_{\mathbf{z}} = \sum_{j=1}^{d} \tilde{f}_{\mathbf{z}}^{(j)} = \sum_{j=1}^{d} \sum_{i=1}^{m} w_i K^{(j)}(x_i^{(j)}, \cdot), w_i \in \mathbb{R}\,.$$

It also has been demonstrated that SALSA in (2-3) can be considered as kernel ridge regression with shrunk features and additive kernels [49]. Despite nice theoretical and empirical analysis, SALSA does not address the sparsity of shrunk features. For high dimensional data, the sparsity on shrunk features usually is benefit to explore the structure information among features, which will improve the interpretability of learning model.

2.2.3 New Sparse Shrunk Additive Models

To improve the sparsity of SALSA, we propose a new algorithm, named as *sparse shrunk additve models* (SSAM). Some sparse methods (*e.g.*, Lasso [93] and kernelized Lasso [76]) can be considered as the special cases of our new model. It is interesting that SSAM also is a natural but nontrivial extension of sparse regularized regression in data dependent hypothesis spaces [85, 91, 29].

For any given training samples $\mathbf{z}$, we introduce the following data dependent hypothesis space:

$$\mathcal{H}_{\mathbf{z}} = \{f : f(x) = \sum_{j=1}^{d} f^{(j)}(x^{(j)}), f^{(j)} \in \mathcal{H}_{\mathbf{z}}^{(j)}\}, \tag{2-4}$$

where $\mathcal{H}_{\mathbf{z}}^{(j)} = \{f^{(j)} = \sum_{i=1}^{m} \alpha_i^{(j)} K^{(j)}(x_i^{(j)}, \cdot) : \alpha_i^{(j)} \in \mathbb{R}\}$ and $K^{(j)} : \mathcal{X}^{(j)} \times \mathcal{X}^{(j)} \to \mathbb{R}$ be a continuous function satisfying $\|K^{(j)}\|_\infty < +\infty$. Without loss of generality, this paper assumes $\|K^{(j)}\|_\infty \leq 1$ for each $1 \leq j \leq d$.

The predictor of SSAM can be expressed as

$$f_{\mathbf{z}} = \sum_{j=1}^{d} f_{\mathbf{z}}^{(j)} = \sum_{j=1}^{d} \sum_{t=1}^{m} \hat{\alpha}_t^{(j)} K^{(j)}(x_t^{(j)}, \cdot),$$

where, for $1 \leq t \leq m$ and $1 \leq j \leq d$,

$$\begin{aligned} \{\hat{\alpha}_t^{(j)}\} = \arg\min_{\alpha_t^{(j)} \in \mathbb{R}, t, j} \Big\{ & \lambda \sum_{j=1}^{d} \sum_{t=1}^{m} |\alpha_t^{(j)}| \\ & + \frac{1}{m} \sum_{i=1}^{m} \big(y_i - \sum_{j=1}^{d} \sum_{t=1}^{m} \alpha_t^{(j)} K^{(j)}(x_t^{(j)}, x_i^{(j)})\big)^2 \Big\}. \end{aligned} \tag{2-5}$$

Let $\alpha^{(j)} = (\alpha_1^{(j)}, ..., \alpha_m^{(j)})^T \in \mathbb{R}^m$ and $\mathbf{K}_i^{(j)} = (K^{(j)}(x_1^{(j)}, x_i^{(j)}), ..., K^{(j)}(x_m^{(j)}, x_i^{(j)}))^T \in \mathbb{R}^m$. Denote $\mathbf{K}_i = ((\mathbf{K}_i^{(1)})^T, ..., (\mathbf{K}_i^{(d)})^T)^T \in \mathbb{R}^{md}$ and $\alpha = ((\alpha^{(1)})^T, ..., (\alpha^{(d)})^T)^T \in \mathbb{R}^{md}$, we can see $\sum_{j=1}^{d} (\mathbf{K}_i^{(j)})^T \alpha^{(j)} = \mathbf{K}_i^T \alpha$. Moreover, by denoting $\mathbf{Y} = (y_1, y_2, ..., y_m)^T \in \mathbb{R}^m$ and $\mathbf{K} = (\mathbf{K}_1, ..., \mathbf{K}_m)^T \in \mathbb{R}^{m \times md}$, we have

$$\hat{\alpha} = \arg\min_{\alpha \in \mathbb{R}^{md}} \Big\{ \frac{1}{m} \|\mathbf{Y} - \mathbf{K}\alpha\|_2^2 + \lambda \|\alpha\|_1 \Big\}. \tag{2-6}$$

Moreover, for $j \in \{1, ..., d\}$ and $q \in \{1, 2\}$, define

$$\begin{aligned} \|f^{(j)}\|_{\ell_q}^q &= \inf \Big\{ m^{q-1} \sum_{t=1}^{m} |\alpha_t^{(j)}|^q : \\ & \qquad f^{(j)} = \sum_{t=1}^{m} \alpha_t^{(j)} K^{(j)}(x_t^{(j)}, \cdot) \Big\} \end{aligned}$$

and $\|f\|_{\ell_q}^q := \sum_{j=1}^d \|f^{(j)}\|_{\ell_q}^q$ for $f = \sum_{j=1}^d f^{(j)}$. Then, we can formulate SSAM from the viewpoint of function approximation as below

$$f_{\mathbf{z}} = \arg\min_{f \in \mathcal{H}_{\mathbf{z}}} \Big\{ \frac{1}{m} \sum_{i=1}^{m} (y_i - f(x_i))^2 + \lambda \|f\|_{\ell_1} \Big\}. \tag{2-7}$$

Except the additive structure on $\mathcal{H}_{\mathbf{z}}$, (2-7) is consistent with the sparse kernel machine in data dependent hypothesis spaces [76, 85].

SSAM can be transformed to other methods by explicit selections on $k, K^{(j)}$. When $k = 1$ and $K^{(j)}(x^{(j)}, \tilde{x}^{(j)}) = x^{(j)}$, our model is equivalent to Lasso [93]. When $k = n$ and $K^{(j)}(x^{(j)}, \tilde{x}^{(j)}) = K(x, \tilde{x})$, SSAM can be considered the kernelized Lasso [76].

Different from SALSA [49], our SSAM is based on general kernel, which is not necessary to be a Mercer kernel. Moreover, our SSAM not only can handle regression prediction by using the interactions between features, but also can explore the structure of shrunk features for model selection. The previous SALSA only works for prediction task.

2.2.4 Comparisons With the Related Methods

Now we provide some comparisons for SSAM in (2-5) with the related regularized methods, including *Kernel ridge regression* (KRR), *Least absolute shrinkage and selection operator* (Lasso) [93], *Kernelized Lasso* (KLasso) [76, 92], *Additive model with kernel regularization* (KAM) [16], *Sparse additive models* (SpAM) [74], *Component selection and smoothing operator* (COSSO) [56], and *Shrunk additive least squares approximation* (SALSA) [49]. A brief summary is presented in Table 1 to show the algorithmic properties including the component function, the regularizer on each component, sample/feature sparsity, feature interaction, and the number of additive components.

Table 1: Properties of Kernel Methods and Additive Models

property	KRR	KLasso	Lasso	KAM	SpAM	COSSO	SALSA	SSAM
Component function	RKHS	continuous	linear	RKHS	Hilbert	Spline	RKHS	continuous
Regularization	K-norm	1-norm	1-norm	K-norm	2,1-norm	2,1-norm	K-norm	1-norm
Sparsity (sample)	$\times$	$\surd$	$\times$	$\times$	$\times$	$\times$	$\times$	$\surd$
Sparsity (feature)	$\times$	$\times$	$\surd$	$\times$	$\surd$	$\surd$	$\times$	$\surd$
Feature Interaction	$--$	$--$	$\times$	$\times$	$\times$	$\surd$	$\surd$	$\surd$
Component number	1	1	n	n	n	$\sum_{k=1}^{d}\binom{n}{k}$	$\binom{n}{k}*$	$\binom{n}{k}*$

K-norm:=Kernel norm. *The number can be reduced largely by incorporating prior information of features.

From Table 1, we know that SSAM bridges sparse kernel regression and sparse additive models. In theory, SSAM not only can exploit the interactions among features for prediction, but also handle the selections on features and samples simultaneously. In particular, the selection of shrunk features can be used to characterize the structure among features, which is essentially different from the grouped features under prior knowledge [108]. By introducing the shrunk features, the proposed SSAM encourages the group features to be selected simultaneously, while the previous sparse additive models [64, 44] usually select feature individually.

Indeed, as shown in [4] ,the nonparametric group Lasso can be seen as a variable selection method in a generalized additive model, and can also be seen as equivalent to learning a convex combination of kernel, a framework referred to multiple kernel learning (MKL). The link between the group Lasso and MKL is established in [4] based on the works in [5, 34]. However, there are key deferences between our SSAM and MKL (or group Lasso in [4]):

1) Hypothesis space (continuous and bounded function space VS RKHS). The proposed SSAM only requires the component functions to be continuous and bounded, which are not necessary to be in reproducing kernel Hilbert spaces (RKHS). That is to say, we consider the generalized kernel-based hypothesis space [85, 91, 14], which is not necessary to be associated with positive definite kernel used in [4].

2) Regularization (1-norm with data-dependent hypothesis space VS Hilbert norm with data-independent RKHS). We use the 1-norm on coefficients, which is different from the Hilbert norm used in the nonparametric group Lasso [4]. From the function approximation point of view, we find the prediction function from data dependent hypothesis spaces [85, 91, 14, 29] with sparsity restriction on samples and features simultaneously (via 1-norm). However, the nonparametric group Lasso [4] is associated with data independent RKHS and only addresses the feature sparsity. In addition, the kernel Lasso [76] only focuses on the sample sparsity since it does not consider the input variable decomposition.

3) Learning theory (Error bound based on integral operator approximation and concentration estimate with empirical covering numbers VS Consistency based on covariance operator analysis). According to 1) and 2), the theory analysis for MKL (e.g. [5, 34]) or group Lasso [4] doesn't hold true for our approach under mild restriction on component function. As studied

in [85, 91, 14, 29], the learning theory analysis is much more difficult for data-dependent hypothesis space with generalized kernel. In this paper, we overcame the difficulty of theoretical analysis by developing and integrating the integral operator approximation [87, 91, 84] and the concentration estimation with empirical covering numbers [102, 85].

2.3 Theoretical Analysis

We begin this section with some necessary definitions and assumptions used in our analysis. Let $L^2_{\rho_{\mathcal{X}^{(j)}}}$ be a square-integrable function space on $\mathcal{X}^{(j)}$ with distribution $\rho_{\mathcal{X}^{(j)}}$. For each $j \in \{1, 2, ..., d\}$ and $f \in L^2_{\rho_{\mathcal{X}^{(j)}}}$, define the integral operator $L_{K^{(j)}} : L^2_{\rho_{\mathcal{X}^{(j)}}} \to L^2_{\rho_{\mathcal{X}^{(j)}}}$ as

$$L_{K^{(j)}}(f)(x^{(j)}) = \int_{\mathcal{X}^{(j)}} K^{(j)}(x^{(j)}, u) f(u) d\rho_{\mathcal{X}^{(j)}}(u).$$

Define $\tilde{K}^{(j)}(x^{(j)}, \tilde{x}^{(j)}) = \int K^{(j)}(x^{(j)}, u) K^{(j)}(\tilde{x}^{(j)}, u) d\rho_{\mathcal{X}^{(j)}}(u)$. It has been verified in [91] that $\tilde{K}^{(j)}$ is a Mercer kernel and $L_{\tilde{K}^{(j)}} = L_{K^{(j)}} L^T_{K^{(j)}} : L^2_{\rho_{\mathcal{X}^{(j)}}} \to L^2_{\rho_{\mathcal{X}^{(j)}}}$ is a self-adjoint positive operator with decreasing eigenvalues $\{\lambda^{(j)}_t\}_{t=1}^{\infty}$ and eigenfunctions $\{\psi^{(j)}_t\}_{t=1}^{\infty}$, where $\{\psi^{(j)}_t\}_{t=1}^{\infty}$ form an orthonormal basis of $L^2_{\rho_{\mathcal{X}^{(j)}}}$. For given $r > 0$, define the r-th power $L^r_{\tilde{K}^{(j)}}$ of $L_{\tilde{K}^{(j)}}$ by

$$L^r_{\tilde{K}^{(j)}}(\sum_t c_{j,t} \psi^{(j)}_t) = \sum_t c_{j,t} (\lambda^{(j)}_t)^r \psi^{(j)}_t, \forall (c_{j,t})_{t \in \mathbb{N}} \in \ell_2.$$

Assumption 1. *Assume that $f_\rho = \sum_{j=1}^d f^{(j)}_\rho$, where for each $j \in \{1, 2, ..., d\}$, $f^{(j)}_\rho : \mathcal{X}^{(j)} \to \mathbb{R}$ is a function of the form $f^{(j)}_\rho = L^r_{\tilde{K}^{(j)}}(g^{(j)}_\rho)$ with some $r > 0$ and $g^{(j)}_\rho \in L^2_{\rho_{\mathcal{X}^{(j)}}}$.*

This regularity condition on f_ρ has been studied for coefficient-based regularized regression with general kernel [91, 84]. For the additive model with Mercer kernel, similar assumption has been introduced in [16].

We also need the Lipschitz continuous condition on each kernel $K^{(j)}$. The restrictive condition has been studied extensively in learning theory of kernel methods, *e.g.*, [85, 84]. In particular, the Gaussian kernel satisfies this condition.

Assumption 2. *For each $j \in \{1, 2, ..., d\}$, the kernel function $K^{(j)} : \mathcal{X}^{(j)} \times \mathcal{X}^{(j)} \to \mathbb{R}$ is $\mathcal{C}^s$ with some $s > 0$ satisfying*

$$\|K^{(j)}(u, v) - K^{(j)}(u, v')\| \leq c_s \|v - v'\|_2^s, \forall u, v, v' \in \mathcal{X}^{(j)}$$

for some positive constant c_s.

From the definition of f_ρ and $\mathcal{Y} \in [-1, 1]$, we know that $|f_\rho(x)| \leq 1$ for any $x \in \mathcal{X}$. Thus, we can utilize the following projection operator to get tight error estimate which is a standard technique in error analysis [18, 90].

Definition 1. *The projection operator π is defined on the space of measurable functions $f : \mathcal{X} \to \mathbb{R}$ as $\pi(f)(x) = \max\big\{-1, \min\{f(x), 1\}\big\}$.*

Denote

$$p = \begin{cases} 2k/(k+2s), & \text{if } s \in (0, 1]; \\ 2k/(k+2), & \text{if } s \in (1, 1 + k/2]; \\ k/s, & \text{if } s \in (1 + k/2, \infty). \end{cases} \tag{2-8}$$

Our first theoretical result is the upper bound of $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)$.

Theorem 1. *Let Assumptions 1 and 2 be true. For any $0 < \delta < 1$, with confidence $1 - \delta$, there exists positive constant $\tilde{c}_1$ independent of m, δ such that:*

(1) If $r \in (0, \frac{1}{2})$ in Assumption 1, setting $\lambda = m^{-\theta_1}$ with $\theta_1 \in (0, \frac{2}{2+p})$,

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq \tilde{c}_1 \log(8/\delta) m^{-\gamma_1},$$

where $\gamma_1 = \min\big\{2r\theta_1, \frac{1-\theta_1+2r\theta_1}{2}, \frac{2}{2+p} - (2-2r)\theta_1, \frac{2(1-p\theta_1)}{2+p}\big\}$.

(2) If $r \geq \frac{1}{2}$ in Assumption 1, taking $\lambda = m^{-\theta_2}$ with some $\theta_2 \in (0, \frac{2}{2+p})$,

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq \tilde{c}_1 \log(8/\delta) m^{-\gamma_2},$$

where $\gamma_2 = \min\Big\{\theta_2, \frac{1}{2}, \frac{2}{2+p} - \theta_2\Big\}$.

Theorem 4 provides the upper bound of generalization error to SSAM with Lipshitz continuous kernel. For $r \in (0, \frac{1}{2})$, as $s \to \infty$, we have $\gamma_1 \to \min\{2r\theta_1, \frac{1}{2} + (r - \frac{1}{2})\theta, 1 - 2\theta_1 + 2r\theta_1\}$. Moreover, when $r \to \frac{1}{2}$ and $\theta_1 \to \frac{1}{2}$, the convergence rate $O(m^{-\frac{1}{2}})$ can be reached.

For $r \geq \frac{1}{2}$, taking $\theta_2 = \frac{1}{2+p}$, we get the convergence rate $O(m^{-\frac{1}{2+p}})$.

The following result is about a special case when $f_\rho^{(j)} \in \mathcal{H}^{(j)}$.

Theorem 2. *Assume that $f_\rho^{(j)} \in \mathcal{H}^{(j)}$ for each $1 \leq j \leq d$. Take $\lambda = m^{-\frac{2}{2+3p}}$ in (2-5). For any $0 < \delta < 1$, with confidence $1 - \delta$ we have*

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq \tilde{c}_2 \log(1/\delta) m^{-\frac{2}{2+3p}},$$

where $\tilde{c}_2$ is a positive constant independent of m, δ, and p is defined in (2-8).

Under the strong condition on f_ρ, the convergence rate can be arbitrary close to $O(m^{-1})$ as $s \to \infty$.

Now we summarize the comparisons on the related convergence analysis of additive models with feature interactions.

- For SALSA in [49], the convergence rate with polynomial decay is also obtained under mild condition on f_ρ. Different from our work, the previous analysis is limited to the Mercer kernel and the error is expressed with the expectation version.
- For the generalized SpAM in [95], theoretical analysis demonstrates its effectiveness to estimate the underlying component functions, which provides stronger guarantees than generalization bound. However, the condition on $\mathcal{H}^{(j)}$ is much restrictive than SSAM.
- For the fixed design setting, the COSSO estimator in [56] has a convergence rate $O(m^{-\frac{\tilde{s}}{2\tilde{s}+1}})$, where $\tilde{s}$ is the order of smoothness of the components in Sobolev space. It can be seen from Theorem 5 that the faster learning rate of SSAM can be reached as $K \in \mathcal{C}^\infty$.

In the future, it is natural to extend the current result from uniform boundedness to unbounded sampling by the analysis techniques in [90, 99, 39].

Besides the generalization ability, SSAM also advocates the sparsity on features and samples by employing the ℓ_1 regularization. The sparsity of SSAM can be characterized as below.

Theorem 3. *For $t \in \{1, 2, ..., m\}$ and $j \in \{1, 2, ..., d\}$, $\hat{\alpha}_t^{(j)} = 0$ if and only if*

$$\Big|\frac{1}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)})\Big| < \frac{\lambda}{2}.$$

Theorem 3 provides a necessary and sufficient condition for the zero pattern of $\hat{\alpha}$. In terms of the discussions in [85], Theorem 3 also implies the probabilistic confidence bound to ensure the sparsity of $\hat{\alpha}_t^{(j)}$ in (2-5). In particular, for the fixed design setting, the sparsity recovery may be achieved by adding some conditions [29, 104]. We leave it for future study.

2.4 Proof

The proofs of Theorems 4 and 5 involve a integration of techniques for error analysis with integral operator approximation [87, 91, 84, 67] and the empirical process theory for analyzing kernel methods [70, 102, 16]. The proof of Theorem 3 follows the analysis technique for sparse characterization [85, 92].

2.4.1 Key Error Decomposition

The key to bound $\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)$ is a novel error decomposition, where some intermediate functions are constructed as the stepping stone functions. Then, we bound the decomposed terms respectively in terms of operator approximation and concentration equalities for empirical processes.

From Proposition 1 in [84], we know that $L_{K^{(j)}}^T = U L_{\tilde{K}^{(j)}}^{\frac{1}{2}}$ and $L_{K^{(j)}} = L_{\tilde{K}^{(j)}}^{\frac{1}{2}} U^T$ for each $j \in \{1, 2, ..., d\}$, where U is a partial isometry on $L^2_{\rho_{\mathcal{X}^{(j)}}}$ with $U^T U$ being the orthogonal prediction onto the RKHS $\mathcal{H}_{\tilde{K}^{(j)}}$.

For any $j \in \{1, 2, ..., d\}$, define the intermediate function $f_\lambda^{(j)}$ by

$$f_\lambda^{(j)} = \arg\min_{f \in L^2_{\rho_{\mathcal{X}^{(j)}}}} \Big\{ \|L_{K^{(j)}} f^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} + \lambda \|U^T f^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \Big\}. \tag{2-9}$$

Denote $f_\lambda = \sum_{j=1}^d f_\lambda^{(j)}$ and $g_\lambda = \sum_{j=1}^d g_\lambda^{(j)}$ with $g_\lambda^{(j)} = L_{K^{(j)}} f_\lambda^{(j)}$.

Define the empirical version of g_λ as

$$\hat{g}_\lambda(x) = \frac{1}{m}\sum_{i=1}^m \sum_{j=1}^d f_\lambda^{(j)}(x_i^{(j)}) K^{(j)}(x_i^{(j)}, x^{(j)}), x \in \mathcal{X}. \tag{2-10}$$

Now we give the following error decomposition.

Proposition 1. *For $f_{\mathbf{z}}, \hat{g}_\lambda$ defined in (2-5) and (2-10), respectively, there holds*

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \le E_1 + E_2 + E_3,$$

where

$$\begin{aligned} E_1 &= \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) + \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}(\hat{g}_\lambda), \\ E_2 &= \mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda) + \lambda \|\hat{g}_\lambda\|_{\ell_1} \end{aligned}$$

and

$$E_3 = \mathcal{E}(g_\lambda) - \mathcal{E}(f_\rho).$$

Proof. According the definition of $f_{\mathbf{z}}$, we have

$$\begin{aligned} & \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \\ \le\ & \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) + \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}(f_\rho) + \lambda \|\hat{g}_\lambda\|_{\ell_1} \\ & + \Big\{ \mathcal{E}_{\mathbf{z}}(f_{\mathbf{z}}) + \lambda \|f_{\mathbf{z}}\|_{\ell_1} - (\mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) + \lambda \|\hat{g}_\lambda\|_{\ell_1}) \Big\} \\ \le\ & \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) + \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}(f_\rho) \\ & + \lambda \|\hat{g}_\lambda\|_{\ell_1}. \end{aligned} \tag{2-11}$$

Note that

$$\begin{aligned} \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}(f_\rho) &= (\mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda)) + \mathcal{E}(g_\lambda) - \mathcal{E}(f_\rho) \\ & \quad + \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}(\hat{g}_\lambda). \end{aligned} \tag{2-12}$$

Combining both (2-11) and (2-12), we get the desired decomposition.

□

The error term E_1 measures the divergence between the empirical risk and the corresponding expected risk, which usually is called sample error in learning theory. In terms of recent theoretical progress for learning with data dependent hypothesis spaces [85, 84, 29], we can bound sample error E_1 via concentration inequality associated with empirical covering numbers [102, 16].

The error term E_2 reflects the drift risk for learning with hypothesis spaces $\mathcal{H}_{\mathbf{z}}$ and $\mathcal{H}$, and hence is called as the hypothesis error. By relating $\mathcal{E}(\hat{g}_\lambda)-\mathcal{E}(g_\lambda)$ with $\sum_{j=1}^d \|\hat{g}_\lambda^{(j)}-g_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}$, we can estimate this hypothesis error through the inequality in Hilbert space [70, 87].

2.4.2 Estimate of Approximation Error E_3

The error term E_3 is called the approximation error, which describes the approximation ability of regularized scheme. Following the approximation analysis with integral operator in [87, 84, 67], we derive the upper bound of E_2 based on the properties of $L_{\tilde{K}^{(j)}}, 1 \leq j \leq d$.

The following lemma is used in our analysis, which is proved in Proposition 2 in [84].

Lemma 1. *From the definition of $f_\lambda^{(j)}$ and $g_\lambda^{(j)} = L_{K^{(j)}} f_\lambda^{(j)}$, $j \in \{1, 2, ..., d\}$, there are*

$$f_\lambda^{(j)} = U(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L_{\tilde{K}^{(j)}}^{\frac{1}{2}} f_\rho^{(j)}$$

and

$$\|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} = \|U^T f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.$$

Lemma 2. *Under Assumption 1, there holds*

$$\begin{aligned} & \|L_{\tilde{K}^{(j)}} f_\lambda^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} + \lambda \|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ \leq \; & \lambda^{\min\{1,2r\}} \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} (2 + \|L_{\tilde{K}^{(j)}}^{r-\frac{1}{2}}\|^2 + \|L_{\tilde{K}^{(j)}}^{r-1}\|^2). \end{aligned}$$

Proof. Recall that $\{\lambda_i^{(j)}, \psi_i^{(j)}\}_{i\geq 1}$ are the normalized eigenpairs of the integral operator $L_{\tilde{K}^{(j)}}$ and $\{\psi\}_{i\geq 1}$ form an orthogonal basis of $L^2_{\rho_{\mathcal{X}^{(j)}}}$. Let $g_\rho^{(j)} = L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)} = \sum_{t=1}^{\infty} a_t \psi_t^{(j)}$. Then $\|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} = \sum_{t=1}^{\infty} (a_t^{(j)})^2 < \infty$.

If Assumption 1 holds for some $r \in (0, \frac{1}{2})$, then from Lemma 9 we have

$$\begin{aligned}
& \|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} = \|U^T f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \|U^T U(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L^{\frac{1}{2}}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L^{\frac{1}{2}}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L^{\frac{1}{2}+r}_{\tilde{K}^{(j)}} L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}
\end{aligned}$$

Moreover,

$$\begin{aligned}
& \lambda \|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \lambda \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L^{\frac{1}{2}+r}_{\tilde{K}^{(j)}} L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \lambda \Big\| \sum_{t\geq 1} \frac{(\lambda_t^{(j)})^{\frac{1}{2}+r}}{\lambda_t^{(j)} + \lambda} a_t^{(j)} \psi_t^{(j)} \Big\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \lambda \sum_{t\geq 1} \frac{\lambda_t^{(j)}}{\lambda_t^{(j)} + \lambda} \cdot \frac{\lambda_t^{2r}}{\lambda_t + \lambda} (a_t^{(j)})^2 \\
\leq \; & \lambda^{2r} \sum_{t\geq 1} \frac{\lambda_t^{(j)}}{\lambda_t^{(j)} + \lambda} (a_t^{(j)})^2 \\
\leq \; & \lambda^{2r} \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}},
\end{aligned} \tag{2-13}$$

where the first inequality follows from Lemma 1 in [67] and the second inequality is obtained based on the definition of $g_\rho^{(j)}$.

If Assumption 1 is true for some $r \geq \frac{1}{2}$,

$$\begin{aligned}
& \lambda \|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
= \; & \lambda \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L_{\tilde{K}^{(j)}} L^{r-\frac{1}{2}}_{\tilde{K}^{(j)}} L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
\leq \; & \lambda \|L^{r-\frac{1}{2}}_{\tilde{K}^{(j)}}\|^2 \cdot \|L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
\leq \; & \lambda \|L^{r-\frac{1}{2}}_{\tilde{K}^{(j)}}\|^2 \cdot \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.
\end{aligned} \tag{2-14}$$

Now turn to bound $\|g_\lambda^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}$. From Lemma 9, we can deduce that

$$\begin{aligned} g_\lambda^{(j)} &= L_{K^{(j)}} f_\lambda^{(j)} = L_{\tilde{K}^{(j)}}(\lambda I + L_{\tilde{K}^{(j)}})^{-1} f_\rho^{(j)} \\ &= (\lambda I + L_{\tilde{K}^{(j)}})^{-1} L_{\tilde{K}^{(j)}} f_\rho^{(j)} \end{aligned}$$

and

$$\|g_\lambda^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} = \lambda^2 \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.$$

For $r \in (0,1)$, we have

$$\begin{aligned} & \lambda \|g_\lambda^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ = \;& \lambda^2 \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L^r_{\tilde{K}^{(j)}} L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ \leq \;& \lambda^2 \sum_{t\geq 1} (a_t^{(j)})^2 \Big(\frac{(\lambda_t^{(j)})^r}{\lambda_t^{(j)} + \lambda} \Big)^2 \\ \leq \;& \lambda^{2r} \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}. \end{aligned} \tag{2-15}$$

For $r \geq 1$, we get

$$\begin{aligned} & \lambda \|g_\lambda^{(j)} - f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ = \;& \lambda^2 \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L_{\tilde{K}^{(j)}} L^{r-1}_{\tilde{K}^{(j)}} L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ \leq \;& \lambda^2 \|L^{r-1}_{\tilde{K}^{(j)}}\|^2 \|L^{-r}_{\tilde{K}^{(j)}} f_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ \leq \;& \lambda^2 \|L^{r-1}_{\tilde{K}^{(j)}}\|^2 \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}. \end{aligned} \tag{2-16}$$

Combining (2-13)-(2-16), we get the desired result. □

Lemma 3. *For $j \in \{1, 2, ..., d\}$ and $g_\lambda^{(j)} = L_{K^{(j)}} f_\lambda^{(j)}$ with $f_\lambda^{(j)}$ defined in Section 4, there hold*

$$\begin{aligned} \|f_\lambda^{(j)}\|_\infty \;\leq\; & \sqrt{2 + \|L^{r-\frac{1}{2}}_{\tilde{K}^{(j)}}\|^2 + \|L^{r-1}_{\tilde{K}^{(j)}}\|^2} \\ & \cdot \lambda^{\min\{-\frac{1}{2}, r-1\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \end{aligned}$$

and

$$\begin{aligned} \|f_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \;\leq\; & \sqrt{2 + \|L^{r-\frac{1}{2}}_{\tilde{K}^{(j)}}\|^2 + \|L^{r-1}_{\tilde{K}^{(j)}}\|^2} \\ & \cdot \lambda^{\min\{0, r-\frac{1}{2}\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}. \end{aligned}$$

Proof. Note that

$$\begin{aligned} f_\lambda^{(j)} &= U(\lambda I + L_{\tilde{K}^{(j)}})^{-1} L_{\tilde{K}^{(j)}}^{\frac{1}{2}} f_\rho^{(j)} \\ &= L_{K^{(j)}}^T (\lambda I + L_{\tilde{K}^{(j)}})^{-1} f_\rho^{(j)} \end{aligned}$$

and

$$\|g_\lambda^{(j)} - f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} = \lambda \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.$$

Therefore,

$$\begin{aligned} \|f_\lambda^{(j)}\|_\infty &\leq \|(\lambda I + L_{\tilde{K}^{(j)}})^{-1} f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ &= \lambda^{-1} \|L_{K^{(j)}} f_\lambda^{(j)} - f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ &\leq \sqrt{2 + \|L_{\tilde{K}^{(j)}}^{r-\frac{1}{2}}\|^2 + \|L_{\tilde{K}^{(j)}}^{r-1}\|^2} \\ &\quad \cdot \lambda^{\min\{-\frac{1}{2}, r-1\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}. \end{aligned}$$

The second statement follows directly form the result of Lemma 10. □

Proposition 2. For $g_\lambda = \sum_{j=1}^d g_\lambda^{(j)} = \sum_{j=1}^d L_{\tilde{K}^{(j)}} f_\lambda^{(j)}$, there holds

$$\begin{aligned} E_3 &\leq \lambda^{\min\{1,2r\}} \Big(2 + \|L_{\tilde{K}^{(j)}}^{r-\frac{1}{2}}\|^2 + \|L_{\tilde{K}^{(j)}}^{r-1}\|^2\Big) \\ &\quad \cdot \Big(\sum_{j=1}^d \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}\Big)^2. \end{aligned}$$

Proof. Based on Cauchy-Schwarz inequality, we can observe that

$$\begin{aligned} \sqrt{E_3} &= \Big(\int_{\mathcal{Z}} (g_\lambda(x) - f_\rho(x))^2 d\rho(x,y)\Big)^{\frac{1}{2}} \\ &= \Big(\int_{\mathcal{Z}} \big(\sum_{j=1}^d (g_\lambda^{(j)}(x^{(j)}) - f_\rho^{(j)}(x^{(j)})\big)^2 d\rho(x,y)\Big)^{\frac{1}{2}} \\ &\leq \sum_{j=1}^d \|L_{\tilde{K}^{(j)}} f_\lambda^{(j)} - f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \end{aligned} \tag{2-17}$$

Lemma 10 tells us that $\forall j \in \{1, 2, ..., d\}$

$$\begin{aligned} &\|L_{\tilde{K}^{(j)}} f_\lambda^{(j)} - f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ \leq\ &\sqrt{2 + \|L_{\tilde{K}^{(j)}}^{r-\frac{1}{2}}\|^2 + \|L_{\tilde{K}^{(j)}}^{r-1}\|^2} \lambda^{\min\{\frac{1}{2}, r\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}. \end{aligned}$$

Combining this estimate with (4-37), we get the desired upper bound on E_3. □

2.4.3 Estimate of Hypothesis Error E_2

The hypothesis error reflects the divergence between $\hat{g}_\lambda$ and g_λ on the expected risk and regularization. The following inequality from [70, 87] is used to bound the divergence.

Lemma 4. *Let $\mathcal{H}$ be a Hilbert space. For an independent random variable ξ on $\mathcal{Z}$ with values in $\mathcal{H}$, assume that $\|\xi\|_{\mathcal{H}} \leq M < \infty$ almost surely. For any given independent identical distributed samples $\{z_i\}_{i=1}^m \subset \mathcal{Z}$ and any $\delta \in (0,1)$, there holds*

$$\begin{aligned} & \Big\| \frac{1}{m} \sum_{i=1}^{m} \xi(z_i) - E\xi \Big\|_{\mathcal{H}} \\ \leq \quad & \frac{2M \log(2/\delta)}{m} + \sqrt{\frac{2E\|\xi\|_{\mathcal{H}}^2 \log(2/\delta)}{m}} \end{aligned}$$

with confidence at least $1 - \delta/2$.

Proposition 3. For any $0 < \delta < 1$, with confidence $1 - \delta$, we have

$$\begin{aligned} & \mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda) \\ \leq \quad & 16\sqrt{c}\lambda^{\min\{0, r-\frac{1}{2}\}} \Big(\frac{\log(2/\delta)}{m} + \sqrt{\frac{\log(2/\delta)}{m}} \Big) \\ & \cdot \Big(\sum_{j=1}^{d} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} + (\sum_{j=1}^{d} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}})^2 \Big) \end{aligned}$$

and

$$E_2 \leq c_2 \Big(\lambda^{\min\{0, r-\frac{1}{2}\}} \sqrt{\frac{\log(2/\delta)}{m}} + \lambda^{\min\{\frac{1}{2}, r\}} \Big),$$

where $c = 2 + \|L_{\tilde{K}^{(j)}}^{r-\frac{1}{2}}\|^2 + \|L_{\tilde{K}^{(j)}}^{r-1}\|^2$ and c_2 is a positive constant independent of m, δ.

Proof. From Cauchy-Schwarz inequality, we can see that

$$\begin{aligned}
& \mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda) \\
\leq\ & \Big(\int_{\mathcal{Z}} (2y - \hat{g}_\lambda(x) - g_\lambda(x))^2 d\rho(x,y) \Big)^{\frac{1}{2}} \\
& \cdot \Big(\int_{\mathcal{Z}} (\hat{g}_\lambda(x) - g_\lambda(x))^2 d\rho(x,y) \Big)^{\frac{1}{2}} \\
\leq\ & \Big(8 + 2 \int_{\mathcal{Z}} (\hat{g}_\lambda(x) - g_\lambda(x))^2 d\rho(x,y) \Big)^{\frac{1}{2}} \\
& \cdot \Big(\int_{\mathcal{Z}} (\hat{g}_\lambda(x) - g_\lambda(x))^2 d\rho(x,y) \Big)^{\frac{1}{2}} \\
\leq\ & \Big(\sqrt{8} + \sqrt{2} \sum_{j=1}^{d} \|\hat{g}_\lambda^{(j)} - g_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \Big) \\
& \cdot \sum_{j=1}^{d} \|\hat{g}_\lambda^{(j)} - g_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.
\end{aligned} \tag{2-18}$$

Denote $\xi^{(j)} = f_\lambda^{(j)}(x^{(j)}) K(x^{(j)}, u)$ for any $j \in \{1, 2, ..., d\}$. Then, from Lemma 3, we can deduce that

$$\|\xi^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \leq \|f_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \leq \sqrt{c} \lambda^{\min\{0, r-\frac{1}{2}\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}$$

and

$$E\|\xi^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \leq \|f_\lambda^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \leq c \lambda^{\min\{0, 2r-1\}} \|g_\rho^{(j)}\|^2_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.$$

Moreover, for any $j \in \{1, ..., d\}$ and $u \in \mathcal{X}^{(j)}$,

$$\begin{aligned}
& \|\hat{g}_\lambda^{(j)} - g_\lambda^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
=\ & \Big\| \frac{1}{m} \sum_{i=1}^{m} \xi_i^{(j)} - E\xi^{(j)} \Big\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\
\leq\ & \frac{2\sqrt{c} \lambda^{\min\{0, r-\frac{1}{2}\}} \|g^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \log(2/\delta)}{m} \\
& + \lambda^{\min\{0, r-\frac{1}{2}\}} \|g_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \sqrt{\frac{2c \log(2/\delta)}{m}},
\end{aligned} \tag{2-19}$$

where the last inequality is derived from Lemma 4. Then, we obtain the first statement by combining the estimates (2-18) and (2-19).

Now consider the upper bound of $\lambda\|\hat{g}_\lambda\|_{\ell_1}$. From the definition of $\hat{g}_\lambda$, we have

$$\begin{aligned}\lambda\|\hat{g}_\lambda\|_{\ell_1} &\leq \lambda\sum_{j=1}^{d}\|f_\lambda^{(j)}\|_\infty \leq \sum_{j=1}^{d}\|L_{K^{(j)}}f_\lambda^{(j)} - f_\rho^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}} \\ &\leq \sqrt{c}\lambda^{\min\{0,r-\frac{1}{2}\}}\sum_{j=1}^{d}\|g^{(j)}\|_{L^2_{\rho_{\mathcal{X}^{(j)}}}}.\end{aligned}$$

Combining this estimate with the first statement, we derive the desired upper bound of E_2. □

2.4.4 Estimate of Sample Error E_1

In this paper, the sample error is estimated by the analysis technique associated with the empirical covering numbers. The empirical covering numbers with ℓ_2-metric is denoted by $\mathcal{N}_2(\mathcal{F},\varepsilon)$ and its detail definition can be founded in [96, 85].

Definition 2. *For a function set $\mathcal{F}$ and $\mathbf{u} = (u_i)_{i=1}^k \in \mathcal{X}$, the metric $d_{2,\mathbf{u}}$ is defined by*

$$d_{2,\mathbf{u}}(f,g) = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(f(u_i)-g(u_i))^2}, \forall f,g\in\mathcal{F}.$$

For every $\varepsilon > 0$, the empirical covering number is defined as $\mathcal{N}_2(\mathcal{F},\varepsilon) = \sup_{k\in\mathbb{N}}\sup_{\mathbf{u}\in\mathcal{X}^k}\mathcal{N}_{2,\mathbf{u}}(\mathcal{F},\varepsilon)$, where

$$\begin{aligned}\mathcal{N}_{2,\mathbf{u}}(\mathcal{F},\varepsilon) = \inf\Big\{&l\in\mathbb{N}: \exists\{f_i\}_{i=1}^{l} \text{ such that} \\ &\mathcal{F}\subset\cup_{i=1}^{l}\{f\in\mathcal{F}: d_{2,\mathbf{u}}(f,f_i)\leq\varepsilon\}\Big\}.\end{aligned}$$

The following concentration inequality is established in [102].

Lemma 5. *Let $\mathcal{F}$ be a measurable function set on $\mathcal{Z}$. Assume that, for any $f \in \mathcal{F}$, $\|f\|_\infty \le B$ and $E(f^2) \le cEf$ for some positive constants B, c. If for some $a > 0$ and $s \in (0,2)$, $\log \mathcal{N}_2(\mathcal{F}, \varepsilon) \le a\varepsilon^{-p}$ for any $\varepsilon > 0$, then there exists a constant c_p' such that for any $\delta \in (0,1)$,*

$$\begin{aligned} & \Big| Ef - \frac{1}{m}\sum_{i=1}^m f(z_i) \Big| \\ \le\ & \frac{1}{2}Ef + c_p' \max\{c^{\frac{2-p}{2+p}}, B^{\frac{2-p}{2+p}}\}(\frac{a}{m})^{\frac{2}{2+p}} \\ & + \frac{(2c+18B)\log(1/\delta)}{m} \end{aligned}$$

with confidence at least $1 - 2\delta$.

For any $R > 0$, denote

$$\begin{aligned} \mathcal{B}_R^{(j)} \ =\ & \Big\{ f^{(j)} = \sum_{i=1}^m \alpha_i^{(j)} K^{(j)}(u_i^{(j)}, \cdot) \in \mathcal{H}^{(j)} : \\ & \|f^{(j)}\|_{\ell_1} \le R \Big\} \end{aligned}$$

and

$$\mathcal{B}_R = \Big\{ f = \sum_{j=1}^d f^{(j)} : \|f\|_{\ell_1} \le R \Big\},$$

where

$$\|f\|_{\ell_1} = \inf \Big\{ \sum_{j=1}^d \|f^{(j)}\|_{\ell_1} : f = \sum_{j=1}^d f^{(j)}, f^{(j)} \in \mathcal{H}^{(j)} \Big\}.$$

Now we state the estimate on the empirical covering numbers of $\mathcal{B}_1$. Similar analysis can be found in [16] for $\mathcal{B}_1$ in reproducing kernel Hilbert spaces.

Lemma 6. *For any $j \in \{1, 2, ..., d\}$, assume that $K^{(j)} \in C^s$ for some $s > 0$. Then,*

$$\log \mathcal{N}_2(\mathcal{B}_1, \varepsilon) \le d^{1+p} c_p \varepsilon^{-p},$$

where p is defined in Section 3 and c_p is a constant independent of ε.

Proof. For every $j \in \{1, 2, ..., d\}$ and $\mathbf{x}^{(j)} \in (\mathcal{X}^{(j)})^S$, there exists a set $\{f_i^{(j)}\}_{i=1}^{N_j}$ with $N_i = \mathcal{N}_2(\mathcal{B}_1^{(j)}, \varepsilon)$ such that

$$\begin{aligned}\forall f^{(j)} \in \mathcal{B}_1^{(j)}, \quad \exists \quad & i_j \in \{1, 2, ..., N_j\}, s.t., \\ & d_{2,\mathbf{x}^{(j)}}(f^{(j)} - f_{i_j}^{(j)}) \leq \varepsilon.\end{aligned}$$

For $f = \sum_{j=1}^d f^{(j)} \in \mathcal{B}_1$, we know $f^{(j)} \in \mathcal{B}_1^{(j)}$. For every $\mathbf{x} = (x_\ell)_{\ell=1}^S \in \mathcal{X}^S$, we have $\mathbf{x}^{(j)} = (x_\ell^{(j)})_{\ell=1}^S \in (\mathcal{X}^{(j)})^S, j \in \{1, 2, ..., d\}$. Let $\tilde{f} = \sum_{j=1}^d f_{i_j}^{(j)}$. Then

$$\begin{aligned} & d_{2,\mathbf{x}}(f, \tilde{f}) \\ = & \Big\{\frac{1}{S}\sum_{\ell=1}^S (f(x_\ell) - \tilde{f}(x_\ell))^2\Big\}^{\frac{1}{2}} \\ = & \Big\{\frac{1}{S}\sum_{\ell=1}^S (\sum_{j=1}^d f^{(j)}(x_\ell^{(j)}) - \sum_{j=1}^d \tilde{f}_{i_j}^{(j)}(x_\ell^{(j)}))^2\Big\}^{\frac{1}{2}} \\ \leq & \sum_{j=1}^d \Big\{\frac{1}{S}\sum_{l=1}^S (f^{(j)}(x_\ell^{(j)}) - \tilde{f}_{i_j}^{(j)}(x_\ell^{(j)}))^2\Big\}^{\frac{1}{2}} \\ \leq & \sum_{j=1}^d d_{2,\mathbf{x}^{(j)}}(f^{(j)} - f_{i_j}^{(j)}) \\ \leq & d\varepsilon. \end{aligned}$$

Therefore,

$$\log \mathcal{N}_2(\mathcal{B}_1, d\varepsilon) \leq \sum_{j=1}^d \log \mathcal{N}_2(\mathcal{B}_1^{(j)}, d\varepsilon).$$

According to Theorem 2 in [85] (also see Lemmas 2 and 3 in [84]) and considering $\|f^{(j)}\|_{\ell_1} \leq \sqrt{m\|f^{(j)}\|_{\ell_2}^2}$, we further get

$$\log \mathcal{N}_2(\mathcal{B}_1, d\varepsilon) \leq dc_p\varepsilon^{-p}.$$

Setting $\tilde{\varepsilon} = d\varepsilon$, we get the desired result. □

Proposition 4. Under Assumptions 1 and 2, for any $\delta \in (0,1)$, there holds

$$\begin{aligned} E_1 \;\leq\; & \frac{1}{2}(\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)) + \frac{1}{2}(\mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda)) \\ & + E_3 + C_1 \log(2/\delta)\big(\lambda^{-\frac{2p}{2+p}} m^{-\frac{2}{2+p}} \\ & + \lambda^{\min\{-1,2r-2\}} m^{-\frac{2}{2+p}} + m^{-1}\big) \end{aligned}$$

with confidence $1-\delta$, where C_1 is a positive constant independent m, λ, δ, and p is defined in Section 3.

Proof. The sample error E_1 can be decomposed as

$$E_{11} = \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_\rho))$$

and

$$E_{12} = \mathcal{E}_{\mathbf{z}}(\hat{g}_\lambda) - \mathcal{E}_{\mathbf{z}}(f_\rho) - (\mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(f_\rho)).$$

In the sequel, we will bound E_{11} and E_{12} respectively.

Denote

$$\mathcal{G}_R = \{g(z) = (y - \pi(f)(x))^2 - (y - f_\rho(x))^2 : f \in \mathcal{B}_R\}.$$

For any $g \in \mathcal{G}_R$, we can deduce that $|g(z)| \leq 8$ and $Eg^2 \leq 16Eg$. Let $g_1, g_2 \in \mathcal{G}_R$ associated with f_1, f_2 respectively. It can be seen that

$$\begin{aligned} |g_1(z) - g_2(z)| \;&\leq\; 4|\pi(f_1)(x) - \pi(f_2)(x)| \\ &\leq\; 4|f_1(x) - f_2(x)|. \end{aligned}$$

This means

$$\begin{aligned} \log \mathcal{N}_2(\mathcal{G}_R, \varepsilon) \;&\leq\; \log \mathcal{N}_2(\mathcal{B}_R, \frac{\varepsilon}{4}) \leq \log \mathcal{N}_2(\mathcal{B}_1, \frac{\varepsilon}{4R}) \\ &\leq\; c_p d^{1+p} (4R)^p \varepsilon^{-p}, \end{aligned}$$

where the last inequality follows form Lemma 6.

Applying Lemma 5 to $\mathcal{G}_R$, we have with confidence $1-\frac{\delta}{2}$

$$Eg - \frac{1}{m}\sum_{i=1}^{m} g(z_i) \le \frac{1}{2}(\mathcal{E}(\pi(f)) - \mathcal{E}(f_\rho))$$
$$+\tilde{c}_1(R^{\frac{2p}{2+p}} m^{-\frac{2}{2+p}} + m^{-1}\log(2/\delta)), \forall g \in \mathcal{G}_R,$$

where $\tilde{c}_1$ is a constant independent of m, δ.

From the definition of $f_{\mathbf{z}}$ in Section 2, we know $f_{\mathbf{z}} \in \mathcal{B}_R$ with $R = \lambda^{-1}$. Then

$$\begin{aligned} E_{11} &\le \frac{1}{2}(\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho)) \\ &\quad +\tilde{c}_1(\lambda^{-\frac{2p}{2+p}} m^{-\frac{2}{2+p}} + m^{-1}\log(1/\delta)) \end{aligned} \tag{2-20}$$

with confidence $1-\frac{\delta}{2}$.

Now we turn to bound E_{12}. Denote

$$\hat{\mathcal{G}} = \Big\{\hat{g} = \sum_{j=1}^{d} \hat{g}_\lambda^{(j)} : \hat{g}_\lambda^{(j)} = \frac{1}{m}\sum_{i=1}^{m} f_\lambda^{(j)}(v_i^{(j)}) K(v_i^{(j)}, \cdot)\Big\}$$

and

$$\hat{\mathcal{H}} = \Big\{h : h(z) = (y - \hat{g}(x))^2 - (y - f_\rho(x))^2, \hat{g} \in \hat{\mathcal{G}}\Big\}.$$

We can verify that

$$\begin{aligned} \|h\|_\infty &= \sup |2y - \hat{g}(x) - f_\rho(x)| \cdot |\hat{g}(x) - f_\rho(x)| \\ &\le (3 + \sum_{j=1}^{d} \|f_\lambda^{(j)}\|_\infty)^2 \end{aligned}$$

and

$$Eh^2 \le (3 + \sum_{j=1}^{d} \|f_\lambda^{(j)}\|_\infty)^2 Eh.$$

For any given $\hat{g}_1, \hat{g}_2 \in \hat{\mathcal{H}}$, the corresponding $h_1, h_2 \in \hat{\mathcal{H}}$ satisfy

$$|h_1(z) - h_2(z)| \le 2(1 + \sum_{j=1}^{d} \|f_\lambda^{(j)}\|_\infty)|\hat{g}_1(x) - \hat{g}_2(x)|.$$

Then, from Lemma 6 and $\hat{g} \in \mathcal{B}_R$ with $R = \sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty$, we have

$$\begin{aligned}
& \log \mathcal{N}_2(\hat{\mathcal{H}}, \varepsilon) \\
\leq & \log \mathcal{N}_2\Big(\hat{\mathcal{G}}, \frac{\varepsilon}{2(1+\sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty)}\Big) \\
\leq & \log \mathcal{N}_2\Big(\mathcal{B}_1, \frac{\varepsilon}{2\sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty (1+\sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty)}\Big) \\
\leq & c_p d^{1+p} 2^p (\sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty + \sum_{j=1}^d \|f_\lambda^{(j)}\|_\infty^2)^p \varepsilon^{-p}.
\end{aligned}$$

Applying Lemma 5 to $\hat{\mathcal{H}}$, with confidence $1 - \frac{\delta}{2}$ we have

$$\begin{aligned}
E_{12} = & \sum_{i=1}^m h(z_i) - Eh \leq \frac{1}{2}(\mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(f_\rho)) \\
& + \tilde{c}_2 \|f_\lambda\|_\infty^2 (m^{-\frac{2}{2+p}} + m^{-1}\log(2/\delta)) \\
\leq & \frac{1}{2}(\mathcal{E}(\hat{g}_\lambda) - \mathcal{E}(g_\lambda) + E_3) \\
& + d\tilde{c}_2' \lambda^{\min\{-1, 2r-2\}} (m^{-\frac{2}{2+p}} + m^{-1}\log(2/\delta)),
\end{aligned}$$

where the last inequality follows from Lemma 3 and $\tilde{c}_2, \tilde{c}_2'$ are some positive constants.

Combining this with the estimates of E_{11} in (2-20), we get the upper bound on E_1. □

2.4.5 Proof of Theorem 1

Proof. Combining Propositions 1-4, we have with confidence $1 - 4\delta$

$$\begin{aligned}
& \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \\
\leq & C\log(2/\delta)(\lambda^{\min\{1,2r\}} + \lambda^{\min\{0, r-\frac{1}{2}\}} m^{-\frac{1}{2}} \\
& + \lambda^{\min\{-1, 2r-2\}} m^{-\frac{2}{2+p}} + \lambda^{-\frac{2p}{2+p}} m^{-\frac{2}{2+p}}).
\end{aligned}$$

When $r \in (0, \frac{1}{2})$, by setting $\lambda = m^{-\theta_1}$ with $0 < \theta_1 < \min\{\frac{1}{p}, \frac{1}{(2+p)(1-r)}\}$, we get with confidence $1 - 4\delta$

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq 4C\log(2/\delta) m^{-\gamma_1},$$

where $\gamma_1 = \min\big\{2r\theta_1, \frac{1}{2} + (r - \frac{1}{2})\theta_1, \frac{2}{2+p} - (2-2r)\theta_1, \frac{2}{2+p} - \frac{2p\theta_1}{2+p}\big\}$.

When $r \geq \frac{1}{2}$, taking $\lambda = m^{-\theta_2}$ with some $0 < \theta_2 < \min\{\frac{1}{p}, \frac{2}{2+p}\}$, we have with confidence $1 - 4\delta$

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq 4C \log(2/\delta) m^{-\gamma_2},$$

where

$$\gamma_2 = \min\Big\{\theta_2, \frac{1}{2}, \frac{2}{2+p} - \theta_2, \frac{2}{2+p} - \frac{2p\theta_2}{2+p}\Big\},$$

This completes the proof. □

2.4.6 Proof of Theorem 2

Theorem 2 is dependent on much stronger conditions on f_ρ than Theorem 1. The proof can be obtained directly by the estimate of E_{11} in Proposition 4.

Proof. Since $f_\rho^{(j)} \in \mathcal{H}^{(j)}$ for each $j \in \{1, 2, ..., d\}$, we know that $f_\rho \in \mathcal{H}$. Then,

$$\begin{aligned}
& \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \\
\leq\ & \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) \\
& + \{\mathcal{E}_{\mathbf{z}}(f_{\mathbf{z}}) + \lambda\|f_{\mathbf{z}}\|_{\ell_1} - (\mathcal{E}_{\mathbf{z}}(f_\rho) + \lambda\|f_\rho\|_{\ell_1})\} \\
\leq\ & \mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) - (\mathcal{E}_{\mathbf{z}}(\pi(f_{\mathbf{z}})) - \mathcal{E}_{\mathbf{z}}(f_\rho)) + \lambda\|f_\rho\|_{\ell_1} \\
=\ & E_{11} + \lambda\|f_\rho\|_{\ell_1}.
\end{aligned}$$

From the estimate of E_{11} in (2-20), with confidence $1 - \delta$ we have

$$\mathcal{E}(\pi(f_{\mathbf{z}})) - \mathcal{E}(f_\rho) \leq \bar{c} \log(1/\delta)(\lambda^{-\frac{2p}{2+p}} m^{-\frac{2}{2+p}} + \lambda),$$

where $\bar{c}$ is a positive constant independent of m, λ.

Taking λ such that $\lambda^{-\frac{2p}{2+p}} m^{-\frac{2}{2+p}} = \lambda$, we get the desired result. □

2.4.7 Proof of Theorem 3

Proof. Denote $\alpha = (\alpha_t^{(j)})_{t,j} \in \mathbb{R}^{md}$, where $t \in \{1, 2, ..., m\}$ and $j \in \{1, 2, ..., d\}$. Define

$$\begin{aligned} G(\alpha) &= \frac{1}{m}\sum_{i=1}^{m}\big(y_i - \sum_{j=1}^{d}\sum_{t=1}^{m}\alpha_t^{(j)}K^{(j)}(x_t^{(j)}, x_i^{(j)})\big)^2 \\ &\quad +\lambda\sum_{j=1}^{d}\sum_{t=1}^{m}|\alpha_t^{(j)}|. \end{aligned}$$

Recall that $f_{\mathbf{z}} = \sum_{j=1}^{d}\sum_{t=1}^{m}\hat{\alpha}_t^{(j)}K^{(j)}(x_t^{(j)}, \cdot)$ and $\hat{\alpha} = (\hat{\alpha}_t^{(j)})_{t,j}$ is the maximizer of $G(\alpha)$. Let $I_+ = \{(t,j) : \hat{\alpha}_t^{(j)} > 0\}$, $I_- = \{(t,j) : \hat{\alpha}_t^{(j)} < 0\}$, and $I_0 = \{(t,j) : \hat{\alpha}_t^{(j)} = 0\}$.

For $(t,j) \in I_+$, we get

$$\begin{aligned} &\frac{\partial G(\alpha)}{\partial \alpha_t^{(j)}}\Big|_{\alpha=\hat{\alpha}} \\ =\ & -\frac{2}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)}) + \lambda \\ =\ & 0 \end{aligned}$$

This means

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)}) = \frac{\lambda}{2}.$$

Similarly, for $(t,j) \in I_-$, there exists

$$\frac{1}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)}) = -\frac{\lambda}{2}.$$

For $(t,j) \in I_0$, there holds

$$\begin{aligned} -\frac{2}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)}) - \lambda &\le \frac{\partial G(\alpha)}{\partial \alpha_t^{(j)}}\Big|_{\alpha=\hat{\alpha}} \\ &\le -\frac{2}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)}) + \lambda. \end{aligned}$$

This means, for any $(t,j) \in I_0$,

$$\Big|\frac{1}{m}\sum_{i=1}^{m}(y_i - f_{\mathbf{z}}(x_i))K^{(j)}(x_t^{(j)}, x_i^{(j)})\Big| < \frac{\lambda}{2}.$$

This completes the proof. □

2.5 Experimental Results

This section shows the empirical evaluation of SSAM. We first introduce the experimental setups following [49], and then validate SSAM's ability for feature selection and regression prediction.

We consider SSAM for pairwise interaction setting, and set $k = 2, d = \binom{n}{2}$. Similar with [49], each kernel on $\mathcal{X}^{(j)}$ is generated from Gaussian kernel. For example, when $x_s^{(j)} = (x_{s1}, x_{s2})$ and $x_t^{(j)} = (x_{t1}, x_{t2})$, the shrunk kernel $K^{(j)}(x_s^{(j)}, x_t^{(j)}) = \exp\{-\frac{(x_{s1}-x_{t1})^2}{2\mu_1^2}\} \cdot \exp\{-\frac{(x_{s2}-x_{t21})^2}{2\mu_2^2}\}$, where $\mu_i = 4.5\sigma_i m^{-\frac{1}{10}}$ and σ_i is the standard deviation on i-th coordination. The regularization parameter λ is chosen via five-fold cross validation with respect to the mean square error (MSE).

We implement our SSAM method via accelerated proximal gradient methods [65] to get the coefficient vector $\hat{\alpha}$. For sparse representation and feature selection, we compute $\sum_{t=1}^{m} \hat{\alpha}_t^{(j)}$ on the j-th pairwise features, and then select the informative shrunk features. For synthetic data, we compare our model with COSSO [56] to validate our motivation for feature selection. For real-word benchmark data, we compare MSE of SSAM with SALSA [49], COSSO [56], SpAM [74], and Lasso [93].

2.5.1 Experiments With Synthetic Data

Following the ideas in [56, 108], we use two different types of data to evaluate the model selection ability of SSAM. The first type of synthetic data has at most one informative pairwise features and the second one has at least two pairwise features. Since SALSA does not concern the selection of shrunk features, we compare the performance SSAM with COSSO [56]. As shown in Table 1, COSSO is based on component functions on both single and pairwise input features.

Generate synthetic data: We generate the n-dimensional input $x_i = (x_{i1}, x_{i2}, ..., x_{in})^T$ with $x_{ij} = \frac{W_{ij}+\eta U_i}{1+\eta}$ and $n = 10$, where W and U are sampled from independent uniform distributions defined in $[-0.5, 0.5]$. Parameter η controls the magnitude of correlation. Inputs are independent if $\eta = 0$ and correlated if $\eta = 1$.

Table 2: Precision@τ for Feature Selection

(a) Synthetic Data I

f^*	(m,n,η)	τ	SSAM	COSSO
a	(100,10,0)	4	3.88	3.69
		5	3.92	3.81
		6	3.93	3.85
	(100,10,1)	4	3.37	2.58
		5	3.68	2.80
		6	3.82	2.91
b	(100,10,0)	1	0.97	1
		2	0.97	1
		3	0.97	1
	(100,10,1)	1	0.95	0.62
		2	0.95	0.65
		3	0.98	0.68
c	(100,10,0)	4	3.94	0.63
		5	3.97	0.68
		6	3.97	0.75
	(100,10,1)	4	3.69	0.84
		5	3.87	0.91
		6	3.92	0.94

(b) Synthetic Data II

f^*	(m,n,η)	τ	SSAM	COSSO
e	(100,10,0)	2	1.05	0.73
		3	1.13	0.90
		4	1.20	0.90
	(100,10,1)	2	1.04	0.13
		3	1.10	0.16
		4	1.12	0.20
f	(100,10,0)	2	0.72	0.88
		3	0.93	1
		4	1.23	1
	(100,10,1)	2	1.90	0.94
		3	1.94	0.94
		4	1.95	0.97
g	(100,10,0)	3	2.94	2.98
		4	2.94	2.98
		5	2.94	3
	(100,10,1)	3	2.85	2.14
		4	2.85	2.40
		5	2.85	2.49

Example set I: We apply SSAM with 100 training samples on three underlying functions (a. simple additive model, b. simple pairwise interaction model, c. multi-ways interaction

model). For $x_t = (x_{t1}, x_{t2}, ..., x_{tn})^T$,

$$\begin{aligned} a.\ f^*(x_t) &= x_{t1} + x_{t2} + x_{t3} + exp(-x_{t4}) \\ b.\ f^*(x_t) &= (2x_{t1} - 1)(2x_{t2} - 1) \\ c.\ f^*(x_t) &= (2sin(x_{t1}) - 1)(2sin(x_{t2}) - 1) \\ &\quad \cdot(2sin(x_{t3}) - 1)(2sin(x_{t4}) - 1) \end{aligned}$$

Example set II: We also apply SSAM with 100 training samples on much complicated interaction models ($e.$ overlapped pairwise interaction, $f.$ independent pairwise interaction, $g.$ circle related pairwise interaction):

$$\begin{aligned} e.\ f^*(x_t) &= (2sin(x_{t1})-1)(2sin(x_{t2})-1) \\ &\quad +sin(x_{t1})sin(x_{t3}), \\ f.\ f^*(x_t) &= 2exp(x_{t1}+x_{t2}+0.2)+2exp^{-1}(x_{t3}+x_{t4}), \\ g.\ f^*(x_t) &= (2x_{t1}-1)(2x_{t2}-1)+(2x_{t2}-1)(2x_{t3}-1) \\ &\quad +(2x_{t1}-1)(2x_{t3}-1). \end{aligned}$$

The final output is $y = f^*(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.25)$. For each example, we make feature selection according to the values of $\sum_{t=1}^{100} \hat{\alpha}_t^{(j)}$ for $j \in \{1, ..., 45\}$. The Precision@τ is used to measure the performance of feature selection, which describes the number of truly informative features in the top-τ selected results. Tables 2(a) and 2(b) provide the average results on Precision@τ after repeating 100 times. In most cases, SSAM performs better than COSSO for feature selection. Especially, SSAM behaves more stable than COSSO in complicated models (*e.g.* c, g) and dependent features.

Table 3: Average MSE on Real Data.

	SSAM	SALSA	COSSO	SpAM	Lasso
Insulin	**1.0146**	1.0206	1.1379	1.2035	1.1103
Skillcraft	**0.5432**	0.5470	0.5551	0.90545	0.6650
Airfoil	**0.4866**	0.5176	0.5178	0.9623	0.5199
Forestfire	**0.3477**	0.3530	0.3753	0.9694	0.5193
Housing	0.3787	**0.2642**	1.3097	0.8165	0.4452
CCPP	0.0694	0.0678	0.9684	**0.0647**	0.0740
Music	0.6295	**0.6251**	0.7982	0.7683	0.6349
Telemonit	0.0689	**0.0347**	5.7192	0.8643	0.0863

2.5.2 Experiments With Real-world Benchmark Data

We compare the prediction performance of SSAM with the most related additive models, where eight data sets are used under the same experimental setups in [49]. The data sets from UCI repository (*http://archive.ics.uci.edu/ml*) and [94], which include *Insulin* ($n = 50, m = 256$), *Skillcraft* ($n = 18, m = 1700$), *Airfoil* ($n = 40, m = 750$), *Forestfire* ($n = 10, m = 211$), *Housing* ($n = 12, m = 256$), *CCPP* ($n = 59, m = 2000$), *Music* ($n = 90, m = 1000$), *Telemonit* ($n = 19, m = 1000$). As shown in Table 3, on all eight benchmark datasets, our SSAM has best results on four of them, second best results on three of the rest, and third best result on the rest one. Experimental results show that our SSAM has comparable performance with SALSA, even if only pairwise interaction features are used.As shown in [49], SALSA has shown competitive performance with many nonparametric models and parametric models (but SALSA cannot do feature selection). Therefore, SSAM is effective for regression prediction besides its capacity for sparse feature selection.

Table 4: Precision@τ for Feature Selection

f^*	(m, n, η)	τ	SSAM
a	(20000,10,0)	4	3.95
		5	4.00
		6	4.00
	(20000,10,1)	4	3.90
		5	3.90
		6	4.00
c	(20000,10,0)	4	3.90
		5	4.00
		6	4.00
	(20000,10,1)	4	3.70
		5	4.00
		6	4.00
g	(20000,10,0)	3	2.90
		4	2.95
		5	3.00
	(20000,10,1)	3	2.85
		4	2.85
		5	3.00

2.5.3 More Experimental Results

According to the reviewer comments of scalability, we did new experiments on simulated data for the high dimensional setting (20,000 samples and other settings remain the same). The average results (with 20 repeats) in Table 4 demonstrate that SSAM scales well in high dimensional setting.

One reviewer suggested us to compare with more methods beside COSSO. We added new

Table 5: Average MSE on Real Data.

	SSAM	RMR
Insulin	**1.0146**	1.0198
Skillcraft	**0.5432**	0.6486
Airfoil	**0.4866**	0.5314
Forestfire	**0.3477**	0.3765
Housing	**0.3787**	0.4375
CCPP	0.0694	**0.0667**
Music	0.6295	**0.6210**
Telemonit	**0.0689**	0.0824

comparison results on simulated data with SpAM and the other new method RMR [100] in Table 6 7. The new results also verify the effectiveness of the proposed method.

In addition, we added new experimental results on real data with RMR in Table 5, This results also show the proposed method is better.

2.5.4 Conclusion

In this paper, we proposed a uniform scheme for nonlinear feature and sample selections under additive models. Learning theory analysis has been provided to demonstrate the convergence and sparsity properties of SSAM, where involves novel analysis technique with integral operator and concentration estimation. Experiments on both synthetic and real-world datasets support the effectiveness of our new model.

Table 6: Precision@τ for Feature Selection

f^*	(m, n, η)	τ	SSAM	SpAM	RMR
a	(100,10,0)	4	3.88	3.85	3.81
		5	3.92	3.90	3.95
		6	3.93	3.97	3.97
	(100,10,1)	4	3.37	3.50	2.64
		5	3.68	3.61	3.02
		6	3.82	4.00	3.06
b	(100,10,0)	1	0.97	0.94	1.00
		2	0.97	1.00	2.00
		3	0.97	1.00	2.00
	(100,10,1)	1	0.95	0.94	0.91
		2	0.95	1.00	0.91
		3	0.98	1.00	0.98
c	(100,10,0)	4	3.94	3.93	3.55
		5	3.97	3.96	3.71
		6	3.97	3.98	3.81
	(100,10,1)	4	3.69	3.54	2.82
		5	3.87	3.83	3.20
		6	3.92	3.40	3.46

Synthetic Data I.

Table 7: Precision@τ for Feature Selection

f^*	(m, n, η)	τ	SSAM	SpAM	RMR
e	(100,10,0)	2	1.05	1.00	1.67
		3	1.13	1.17	1.96
		4	1.20	1.19	2.13
	(100,10,1)	2	1.04	1.00	1.26
		3	1.10	1.13	1.64
		4	1.12	1.30	2.97
f	(100,10,0)	2	0.72	0.89	1.83
		3	0.93	1.00	2.37
		4	1.23	1.00	2.97
	(100,10,1)	2	1.90	2.00	1.13
		3	1.94	2.00	1.66
		4	1.95	2.00	1.93
g	(100,10,0)	3	2.94	2.90	3.00
		4	2.94	2.98	3.00
		5	2.94	3.00	3.00
	(100,10,1)	3	2.85	2.80	2.50
		4	2.85	2.82	2.72
		5	2.85	3.00	2.84

Synthetic Data II.

3.0 Optimizing Large-Scale Hyperparameters via Automated Learning Algorithm

Modern machine learning algorithms usually involve tuning multiple (from one to thousands) hyperparameters which play a pivotal role in terms of model generalizability. Black-box optimization and gradient-based algorithms are two dominant approaches to hyperparameter optimization while they have totally distinct advantages. How to design a new hyperparameter optimization technique inheriting all benefits from both approaches is still an open problem. To address this challenging problem, in this paper, we propose a new hyperparameter optimization method with zeroth-order hyper-gradients (HOZOG). Specifically, we first exactly formulate hyperparameter optimization as an $\mathcal{A}$-based constrained optimization problem, where $\mathcal{A}$ is a black-box optimization algorithm (such as deep neural network). Then, we use the average zeroth-order hyper-gradients to update hyperparameters. We provide the feasibility analysis of using HOZOG to achieve hyperparameter optimization. Finally, the experimental results on three representative hyperparameter (the size is from 1 to 1250) optimization tasks demonstrate the benefits of HOZOG in terms of *simplicity, scalability, flexibility, effectiveness and efficiency* compared with the state-of-the-art hyperparameter optimization methods.

3.1 Introduction

Modern machine learning algorithms usually involve tuning multiple hyperparameters whose size could be from one to thousands. For example, support vector machines [97] have the regularization parameter and kernel hyperparameter, deep neural networks [51] have the optimization hyperparameters (e.g., learning rate schedules and momentum) and regularization hyperparameters (e.g., weight decay and dropout rates). The performance of the most prominent algorithms strongly depends on the appropriate setting of these hyperparameters.

Traditional hyperparameter tuning is a bi-level optimization problem as follows.

$$\min_{\lambda \in \mathbb{R}^p} f(\lambda) = E(w(\lambda), \lambda), \quad s.t. \;\; w(\lambda) \in \arg\min_{w \in \mathbb{R}^d} L(w, \lambda) \tag{3-21}$$

where $w \in \mathbb{R}^d$ are the model parameters, $\lambda \in \mathbb{R}^p$ are the hyperparameters, the outer objective E [1] represents a proxy of the generalization error w.r.t. the hyperparameters, the inner objective L represents traditional learning problems (such as regularized empirical risk minimization problems), and $w(\lambda)$ are the optimal model parameters of the inner objective L for the fixed hyperparameters λ. Note that the size of hyperparameters is normally much smaller than the one of model parameters (*i.e.*, $p \ll d$). Choosing appropriate values of hyperparameters is extremely computationally challenging due to the nested structure involved in the optimization problem. However, at the same time both researchers and practitioners desire the hyperparameter optimization methods as *effective*, *efficient*, *scalable*, *simple* and *flexible*[2] as possible.

Classic techniques such as grid search [37] and random search [7] have a very restricted application in modern hyperparameter optimization tasks, because they only can manage a very small number of hyperparameters and cannot guarantee to converge to local/global minima. For modern hyperparameter tuning tasks, black-box optimization [88, 24] and gradient-based algorithms [62, 31, 30] are currently the dominant approaches due to the advantages in terms of *effectiveness, efficiency, scalability, simplicity and flexibility* which are abbreviated as E2S2F in this paper. We provide a brief review of representative black-box optimization and gradient-based hyperparameter optimization algorithms in §3.2.1, and a detailed comparison of them in terms of the above properties in Table 8.

Table 8 clearly shows that black-box optimization and gradient-based approaches have totally distinct advantages, *i.e.*, black-box optimization approach is simple, flexible and salable in term of model parameters, while gradient-based approach is effective, efficient and scalable in term of hyperparmeters. Each property of E2S2F is an important criterion to a successful hyperparameter optimization method. To the best of our knowledge, there is still

[1] The choice of objective function E depends on the specified tasks. For example, accuracy, AUC or F1 can be used for binary classification problem. Square error loss or absolute error loss can be used as the objective of E for regression problems on validation samples.

[2] "effective": good generalization performance. "efficient": running fast. "scalable": scalable in terms of the sizes of hyperparameters and model parameters. "simple": easy to be implemented. "flexible": flexible to various learning algorithms.

Table 8: Representative Black-box Optimization and Gradient-Based Hyperparameter Optimization Algorithms.

Algorithm	**Type**	**Properties**					
		Effective	**Efficient**	**Scalable-H**	**Simple**	**Flexible**	**Scalable-P**
GPBO [88]	BB	♣	♣	✗	✓	✓	✓
BOHB [24]	BB	♣	♣	✗	✓	✓	✓
HOAG [69]	G	✓	✓	✓	✗	✗	✗
RMD [62]	G	✓	✓	✓	✗	✗	✗
RFHO [30, 31]	G	✓	✓	✓	✗	✗	✗
HOZOG	BB+G	✓	✓	✓	✓	✓	✓

no algorithm satisfying all the five properties simultaneously. Designing a hyperparameter optimization method having the benefits of both approaches is still an open problem.

To address this challenging problem, in this paper, we propose a new hyperparameter optimization method with zeroth-order hyper-gradients (HOZOG). Specifically, we first exactly formulate hyperparameter optimization as an $\mathcal{A}$-based constrained optimization problem, where $\mathcal{A}$ is a black-box optimization algorithm (such as the deep neural network). Then, we use the average zeroth-order hyper-gradients to update hyperparameters. We provide the feasibility analysis of using HOZOG to achieve hyperparameter optimization. Finally, the experimental results of various hyperparameter (the size is from 1 to 1250) optimization problems demonstrate the benefits of HOZOG in terms of E2S2F compared with the state-of-the-art hyperparameter optimization methods.

3.2 Hyperparameter Optimization Based on Zeroth-Order Hyper-Gradients

In this section, we first give a brief review of black-box optimization and gradient-based algorithms, and then provide our HOZOG algorithm. Finally, we provide the feasibility analysis of HOZOG.

3.2.1 Brief Review of Black-Box Optimization and Gradient-based Algorithms

3.2.1.1 Black-box Optimization Algorithms

Black-box optimization algorithms view the bilevel optimization problem f as a black-box function. Existing black-box optimization methods [88, 24] mainly employ Bayesian optimization [10] to solve (3-21). Black-box optimization approach has good simplicity and flexibility. However, a lot of references have pointed out that it can only handle hyperparmeters from a few to several dozens [24] while the number of hyperparmeters in real hyperparameter optimization problems would range from hundreds to thousands. Thus, black-box optimization approach has weak scalability in term of the size of of hyperparmeters.

3.2.1.2 Gradient-based Algorithms

The existing gradient-based algorithms can be divided into two parts (*i.e.*, inexact gradients and exact gradients). The approach of inexact gradients first solves the inner problem approximately, and then estimates the gradient of (3-21) based on the approximate solution by the approach of implicit differentiation [69]. Because the implicit differentiation involves Hessian matrices of sizes of $d \times d$ and $d \times p$ where $p \ll d$, they have poor scalability. The approach of exact gradients[3] treats the inner level problem as a dynamic system, and use chain rule [79] to compute the gradient. Because the chain rule highly depends on specific learning algorithms, this approach has poor flexibility and simplicity. Computing the gradients involves Hessian matrices of sizes of $p \times p$ and $d \times p$. Thus, the approach of exact gradients has better scalability than the approach of inexact gradients because normally we have $p \ll d$.

3.2.1.3 Enlightenment

As introduced in [66, 36], zeroth-order gradient (also known as finite difference approximation [19]) technique is a black-box optimization method which estimates the gradient

[3]Although the inner-problem is usually solved approximately *e.g.* by taking a finite number of steps of gradient descent, we still call this kind of methods as exact gradients throughout this paper to avoid using too complex terminology.

only by two function evaluations. Thus, zeroth-order gradient technique belongs both to black-box optimization and gradient-based optimization. We hope that the hyperparameter optimization method bases on zeroth-order hyper-gradients[4] can inherit all benefits as described in Table 8.

3.2.2 HOZOG Algorithm

► **Principle:** Instead of directly computing the hyper-gradient as in [69, 62, 30, 31], we use two function evaluations (*i.e.*, the zeroth-order hyper-gradient technique [66, 36]) to estimate the hyper-gradient, and update hyperparameters with hyper-gradients which derives our HOZOG algorithm.

Before presenting HOZOG algorithm in detail, we first clarify what problem we are solving exactly.

► **What problem we are solving exactly?** As mentioned in (3-21), the inner level problem in the traditional hyperparameter tuning is finding the model parameters that minimize the inner objective L, (*i.e.*, $w(\lambda) \in \arg\min_{w\in\mathbb{R}^d} L(w,\lambda)$). However, in the real-world hyperparameter tuning problems, we are usually trying to find an approximate minimum solution of L by an optimization algorithm if the inner level problem L in convex. If the inner level problem L in non-convex, we usually try to find an approximate local solution or a stationary point. Thus, we replace the inner level problem by $w(\lambda) = \mathcal{A}(\lambda)$ where $\mathcal{A}$ is an optimization algorithm which approximately solves the inner objective L. Further, we replace the bi-level optimization problem (3-21) by the following $\mathcal{A}$-based constrained optimization problem (3-22).

$$\min_{\lambda\in\mathbb{R}^p} f(\lambda) = E(w(\lambda),\lambda), \quad s.t. \;\; w(\lambda) = \mathcal{A}(\lambda) \tag{3-22}$$

where $w(\lambda)$ are the values returned by the optimization algorithm $\mathcal{A}$.

• *Hyperparameters*: Hyperparameters can be divided into two types, *i.e.*, problem-based hyperparameters and algorithm-based hyperparameters.

[4]We call the gradient *w.r.t.* hyperparameter as hyper-gradient in this paper.

1. Problem-based hyperparameters: The problem-based hyperparameters are the hyperparameters involved in *learning problems* such as the regularization parameter and the architectural hyperparameters in deep neural networks.
2. Algorithm-based hyperparameters: These are the hyperparameters involved in *optimization algorithms* such as the learning rate, momentum and dropout rates.

The traditional bi-level optimization problem (3-21) can only formulate the problem-based hyperparameters. However, our $\mathcal{A}$-based constrained optimization problem (3-22) can formulate both types of hyperparameters.

▶ **Algorithm:** To solve the $\mathcal{A}$-based constrained optimization problem (3-22), we propose HOZOG algorithm in Algorithm 1, where the "for" loop is referred to as "meta-iteration". We describe the two key operations of Algorithm 1 (*i.e.*, estimating the function value and average zeroth-order hyper-gradient) in detail as follows.

• *Estimating the function value*: We treat the optimization algorithm $\mathcal{A}$ as a black-box oracle. Given hyperparameters λ, the black-box oracle $\mathcal{A}$ returns model parameters $w(\lambda)$. Based on the pair of λ and $w(\lambda)$, the function value can be estimated as $E(w(\lambda), \lambda)$.

• *Computing the average zeroth-order hyper-gradient*: Zeroth-order hyper-gradient can be computed as $\bar{\nabla} f(\lambda) = \frac{p}{\mu} \left(f(\lambda + \mu u) - f(\lambda)\right) u$ based on the two function evaluations $f(\lambda + \mu u)$ and $f(\lambda)$, where $u \sim N(0, I_p)$ is a random direction drawn from a uniform distribution over a unit sphere, and μ is an approximate parameter. $\bar{\nabla} f(\lambda)$ has a large variance due to single direction u. To reduce the variance, we use the average zeroth-order hyper-gradient (3-23) by sampling a set of directions $\{u_i\}_{i=1}^{q}$.

$$\hat{\nabla} f(\lambda) = \frac{p}{\mu q} \sum_{i=1}^{q} \left(f(\lambda + \mu u_i) - f(\lambda)\right) u_i \tag{3-23}$$

Based on the average zeroth-order hyper-gradient $\hat{\nabla} f(\lambda)$, we update the hyperparameters as follows.

$$\lambda \leftarrow \lambda - \gamma \hat{\nabla} f(\lambda) \tag{3-24}$$

Note that $\hat{\nabla} f(\lambda)$ is a biased approximation to the true gradient $\nabla f(\lambda)$. Its bias can be reduced by decreasing the value of μ. However, in a practical system, μ could not be too

Algorithm 1 Hyperparameter Optimization Method With Zeroth-order Hyper-gradients (HOZOG)

Input: Learning rate γ, approximate parameter μ, size of directions q and black-box inner solver $\mathcal{A}$.

1: Initialize $\lambda_0 \in \mathbb{R}^p$.

2: **for** $t = 0, 1, 2, \ldots, T-1$ **do**

3: Generate $u = [u_1, \ldots, u_q]$, where $u_i \sim N(0; I_p)$.

4: Compute the average zeroth-order hyper-gradient $\hat{\nabla} f(\lambda_t) = \frac{p}{\mu q} \sum_{i=1}^{q} \left(f(\lambda_t + \mu u_i) - f(\lambda_t)\right) u_i$, where $f(\lambda_t)$ is estimated based on the solution returned by the black-box inner solver $\mathcal{A}$.

5: Update $\lambda_{t+1} \leftarrow \lambda_t - \gamma \hat{\nabla} f(\lambda_t)$.

6: **end for**

Output: λ_T.

small, because in that case the function difference could be dominated by the system noise (or error) and fails to represent the function differential [55].

• *Parallel acceleration.* Because the average zeroth-order hyper-gradient involves $q + 1$ function evaluations as shown in (3-23), we can use GPU or multiple cores to compute the $q + 1$ function evaluations in parallel to accelerate the computation of average zeroth-order hyper-gradients.

3.2.3 Feasibility Analysis

▶ **Challenge:** In treating the optimization algorithm $\mathcal{A}(\lambda)$ as a black-box oracle that maps λ to w, the most important problem is whether the mapping function $\mathcal{A}(\lambda)$ is *continuous* which is the basis of using the zeroth-order hyper-gradient technique to optimize (3-22).

• **Continuity:** Before discussing the continuity of the $\mathcal{A}$-based constrained optimization problem $f(\lambda)$, we first give the definitions of iterative algorithm and continuous function in Definitions 3 and 4 respectively.

Definition 3 (Iterative algorithm)**.** *Assume the optimization algorithm $\mathcal{A}(\lambda)$ can be formulated as a nested function as $\mathcal{A}(\lambda) = w_T$ and $w_t = \Phi_t(w_{t-1}, \lambda)$ for $t = 1, \ldots, T$, where*

T is the number of iterations, w_0 is an initial solution, and, for every $t \in \{1, \ldots, T\}$, $\Phi_t : (\mathbb{R}^d \times \mathbb{R}^p) \to \mathbb{R}^d$ is a mapping function that represents the operation performed by the t-th step of the optimization algorithm. We call the optimization algorithm $\mathcal{A}(\lambda)$ as an iterative algorithm.

Definition 4 (Continuous function)**.** *For all $\lambda \in \mathbb{R}^p$, if the limit of $f(\lambda + \delta)$ as $\delta \in \mathbb{R}^p$ approaches $\boldsymbol{0}$ exists and is equal to $f(\lambda)$, we call the function $f(\lambda)$ is continuous everywhere.*

Based on Definitions 3 and 4, we give Theorem 4 to show that the $\mathcal{A}$-based constrained optimization problem $f(\lambda)$ is continuous under mild assumptions. The proof is provided in Appendix.

Theorem 4. *If the hyperparameters λ are continuous and the mapping functions $\Phi_t(w_{t-1}, \lambda)$ (for every $t \in \{1, \ldots, T\}$) are continuous, the mapping function $\mathcal{A}(\lambda)$ is continuous, and the outer objective E is continuous, we have that the $\mathcal{A}$-based constrained optimization problem $f(\lambda)$ is continuous* w.r.t. *λ.*

We provide several popular types of optimization algorithms to show that almost existing iterative algorithms are continuous mapping functions which would make $f(\lambda)$ continious.

1. **Gradient descent algorithms**: If $\mathcal{A}$ is a gradient descent algorithm (such as SGD [32], SVRG [75, 2], SAGA [20], SPIDER [25]), the updating rules can be formulated as $w \leftarrow w - \gamma' v$, where v is a stochastic or deterministic gradient estimated by the current w, and γ' is the learning rate. To accelerate the training of deep neural networks, multiple adaptive variants of SGD (e.g., Adagrad, RMSProp and Adam [33]) have emerged.
2. **Proximal gradient descent algorithms**: If $\mathcal{A}$ is a proximal gradient descent algorithm [111, 103, 35], the updating rules should be the form of $w \leftarrow \text{Prox}(w - \gamma' v)$, where Prox is a proximal operator (such as the soft-thresholding operator for Lasso [93]) which is normally continuous [9, 113].

It is easy to verify that the mapping functions $\mathcal{A}(\lambda)$ corresponding to these iterative algorithms are continuous according to Theorem 4.

For a continuous function $f(\lambda)$, there exists a Lipschitz constant L (see Definition 5) which upper bounds $\frac{|f(\lambda_1) - f(\lambda_2)|}{\|\lambda_1 - \lambda_2\|}$, $\forall \lambda_1, \lambda_2 \in \mathbb{R}^p$. Unfortunately, exactly calculating the Lips-

chitz constant of $f(\lambda)$ is NP-hard problem [98]. We provide an upper bound[5] to the Lipschitz constant of $f(\lambda)$ in Theorem 5.

Definition 5 (Lipschitz continuous constant)**.** *For a continuous function $f(\lambda)$, there exists a constant L such that, $\forall \lambda_1, \lambda_2 \in \mathbb{R}^p$, we have $\|f(\lambda_1) - f(\lambda_2)\| \leq L\|\lambda_1 - \lambda_2\|$. The smallest L for which the inequality is true is called the Lipschitz constant of $f(\lambda)$.*

Theorem 5. *Given the continuous mapping functions $\Phi_t(w_{t-1}, \lambda)$ where $t \in \{1, \ldots, T\}$), $A_t = \frac{\partial \Phi_t(w_{t-1},\lambda)}{\partial w_{t-1}}$, $B_t = \frac{\partial \Phi_t(w_{t-1},\lambda)}{\partial \lambda}$. Given the continuous objective function $E(w_T, \lambda)$, $A_{T+1} = \frac{\partial E(w_T,\lambda)}{\partial w_T}$ and $B_{T+1} = \frac{\partial E(w_T,\lambda)}{\partial \lambda}$. Let $L_{A_t} = \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|A_{t+1}\|_2$, $L_{B_t} = \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|B_t\|_2$. Let $L(f)$ denote the Lipschitz constant of the continuous function $f(\lambda)$, we can upper bound $L(f)$ by $\sum_{t=1}^{T+1} L_{B_t} L_{A_{t+1}} \ldots L_{A_{T+1}}$.*

► **Conclusion:** Because the $\mathcal{A}$-based constrained optimization problem $f(\lambda)$ is continuous, we can use the zeroth-order hyper-gradient technique to optimize $f(\lambda)$ [66]. [66] provided the convergence guarantee of zeroth-order hyper-gradient method when $f(\lambda)$ is Lipschitz continuous as defined in Definition 5.

3.3 Experiments

We conduct the hyperparameter optimization experiments on three representative learning problems (*i.e.*, l_2-regularized logistic regression, deep neural networks (DNN) and data hyper-cleaning), whose sizes of hyperparameters are from 1 to 1250. We also test the parameter sensitivity analysis of HOZOG under different settings of parameters q, μ and γ, which are included in Appendix due to the page limit. All the experiments are conducted on a Linux system equipped with four NVIDIA Tesla P40 graphic cards.

• **Compared algorithms:** We compare our HOZOG with the representative hyperparameter optimization approaches such as random search (RS) [7], RFHO with forward (FOR) or

[5] Although the upper bound is related to T, our simulation results show that it does not grow exponentially with T because L_{A_t} or L_{B_t} is not larger than one at most times.

Table 9: The Parameter Settings of HOZOG in the Experiments.

Experiment		**# HP**	**Dataset**	q	μ	γ
l_2-regularized logistic regression		1	News20	1	0.01	0.05
			Covtype	1	0.01	0.03
			Real-sim	1	0.01	0.005
Deep Neural Networks	2-layer CNN	100	CIFAR-10	1	1	1
	VGG-16	20		3	1	1
	ResNet-152	10		3	1	1
Data hyper-cleaning		500/1250	Mnist	5	1	1

reverse (REV) gradients [30] [6], HOAG [69][7], GPBO [88] [8] and BOHB [24] [9]. Most of them are the representative black-box optimization and gradient-based hyperparameter optimization algorithms as presented in Table 8. We implement our HOZOG in Python[10].

• **Evaluation criteria:** We compare different algorithms with three criteria, *i.e.*, $\|\nabla f(\lambda)\|_2$, suboptimality and test error, where "suboptimality" denotes $f(\lambda) - f(\lambda^\diamond)$ and $f(\lambda^\diamond)$ is the minimum value of $f(\lambda)$ for all λ which have been explored, and test error is the average loss on the testing set. Note the hyper-gradients $\nabla f(\lambda)$ for all method except for FOR and REV are computed by Eq. (3-23).

• **Datasets:** The datasets used in experiments are News20, Covtype, Real-sim, CIFAR-10 and Mnist datasets from LIBSVM repository, which is available at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. Especially, for News20 and Mnist two multi-class datasets, we transform them to binary classification problems by randomly partitioning the data into two groups.

• **Parameters of HOZOG:** The values of parameters q, μ and γ in HOZOG are given in

[6]The code of RFHO is is available at `https://github.com/lucfra/RFHO`.

[7]The code of HOAG is available at `https://github.com/fabianp/hoag`.

[8]The code of GPBO is available at `http://github.com/fmfn/BayesianOptimization/`.

[9]The code of BOHB is available at `https://github.com/automl/HpBandSter`. Note that BOHB is an improved version of Hyperband [54]. Thus, we do not compare HOZOG with Hyperband.

[10]We will release the code of HOZOG and the experiments after the paper is accepted.

Table 9. Especially, q plays an important role to HOZOG because it determines the accuracy and the running time of estimating the gradients. We empirically observe that $q \leq 5$ has a good balance between the two objectives.

3.3.1 l_2-Regularized Logistic Regression

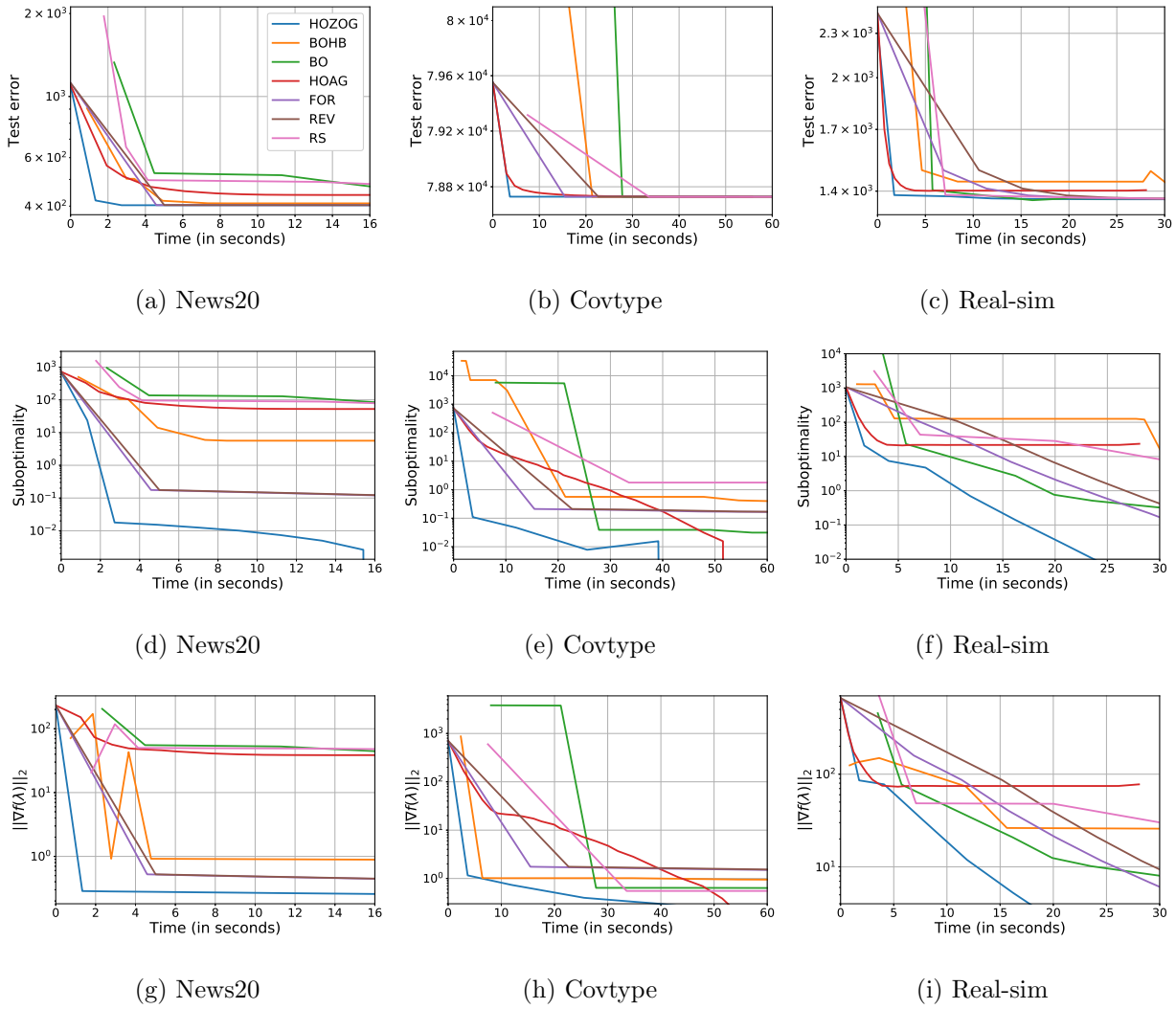
Experimental setup: We consider to estimate the regularization parameter in the l_2-regularized logistic regression model. We split one data set into three subsets (*i.e.*, the train set $\mathcal{D}_{tr}$, validation set $\mathcal{D}_{val}$ and test set $\mathcal{D}_t$) with a ratio of 2:1:1. We use the logistic loss $l(t) = \log(1 + e^{-t})$ as the loss function. The hyperparameter optimization problem for l_2-regularized logistic regression is formulated as follows.

$$\arg\min_{\lambda \in [-10,10]} \sum_{i \in \mathcal{D}_{val}} l(y_i \langle x_i, w(\lambda) \rangle), \qquad s.t. \quad w(\lambda) \in \arg\min_{w \in \mathbb{R}^d} \sum_{i \in \mathcal{D}_{tr}} l(y_i \langle x_i, w(\lambda) \rangle) + e^{\lambda} \|w\|^2 \quad (3\text{-}25)$$

The solver used for solving the inner objective is L-BFGS[11] [57] for HOAG and Adam [50] for the others.

Results and discussions: Figure 1 presents the convergence results of suboptimality, $\|\nabla f(\lambda)\|_2$ and test error *vs.* the running time for different methods. Note that we take same initial values of λ and w for all gradient-based methods, while the black-box methods naturally start from different points. Because HOAG works with tolerances and warm start strategy, HOAG has a fast convergence at the early stage but a slow convergence at the late stage as shown in Figures 1(d)-1(f). We observe that HOZOG runs faster than other gradient-based methods. This is because that FOR and REV need much time to compute hyper-gradients. Figures 1(d)-1(f) provide $\|\nabla f(\lambda)\|_2$ of different methods as functions of running time. We can see that the black-box methods (*i.e.*, BOHB and GPBO) spend much time on exploring because $\|\nabla f(\lambda)\|_2$ of these methods didn't strictly go down in the early stage. Overall, all the results show that HOZOG has a faster convergence than other methods.

[11] The implementation is available at `https://github.com/fabianp/hoag`.

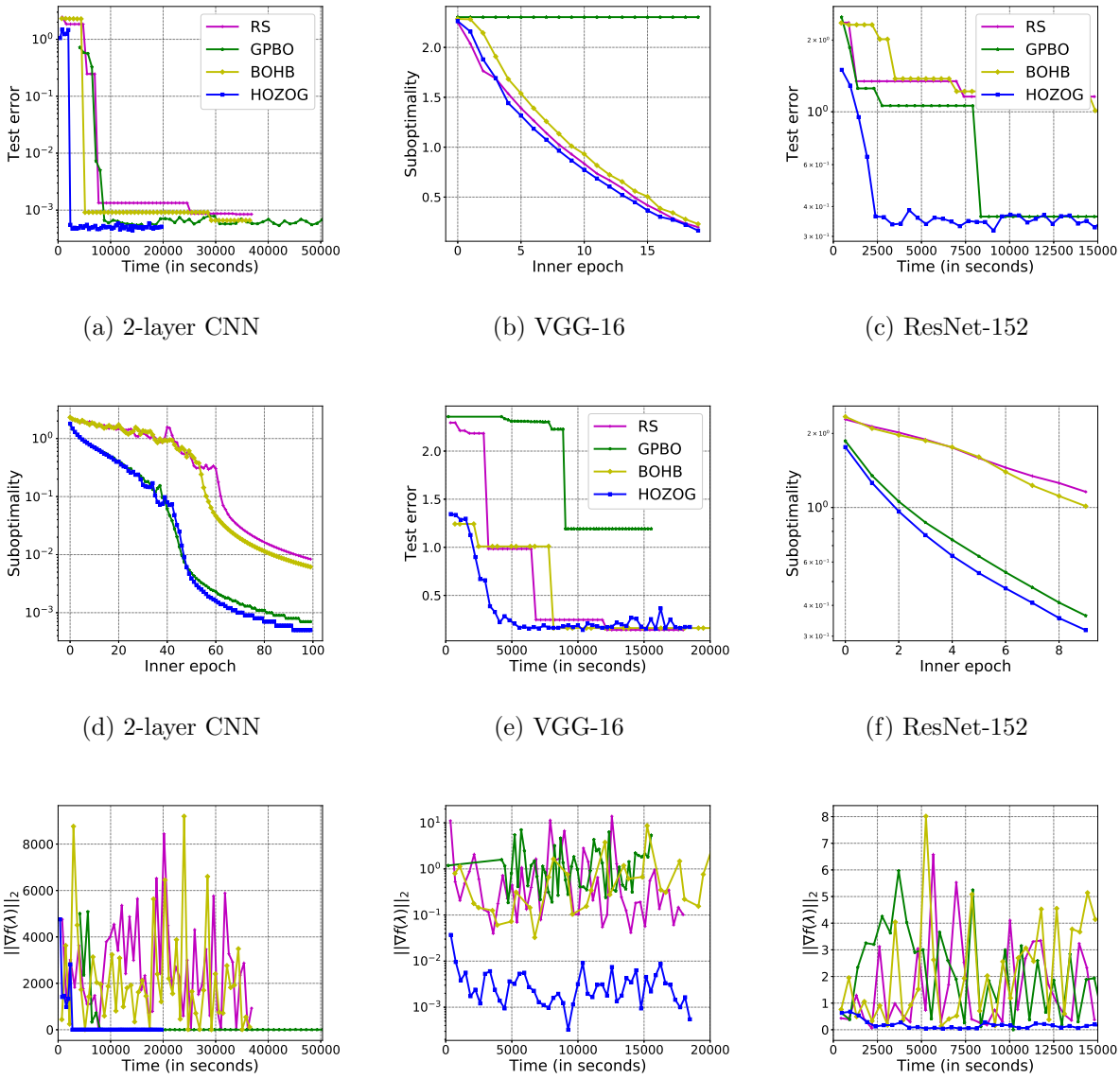


(a) News20 (b) Covtype (c) Real-sim

(d) News20 (e) Covtype (f) Real-sim

(g) News20 (h) Covtype (i) Real-sim

Figure 1: Comparison of Different Hyperparameter Optimization Algorithms for l_2-Regularized Logistic Regression. (a)-(c): Test Error. (d)-(f): Suboptimality. (g)-(i): $\|\nabla f(\lambda)\|_2$.



(a) 2-layer CNN (b) VGG-16 (c) ResNet-152

(d) 2-layer CNN (e) VGG-16 (f) ResNet-152

(g) 2-layer CNN (h) VGG-16 (i) ResNet-152

Figure 2: Comparison of Different Hyperparameter Optimization Algorithms for 2-layer CNN, VGG-16 and ResNet-152. (a)-(c): Test Error. (d)-(f): Suboptimality. (g)-(i): $\|\nabla f(\lambda)\|_2$.

3.3.2 Deep Neural Networks

Experimental setup: We validate the advantages of HOZOG on optimizing learning rates of DNN which is much more complicated in both structure and training compared to l_2-regularized logistic regression.

Specifically, the training of modern DNN is usually an intriguing process, involving multiple heuristic hyperparameter schedules, *e.g.* learning rate with exponential weight decay. Instead of intuitive settings, we propose to apply epoch-wise learning rates and jointly optimize these hyperparameters. The experiments are conducted on CIFAR-10 dataset with 50,000 samples. To demonstrate the scalability of HOZOG, three deep neural networks with various structure are used, including (1) two layers DNN (2-layer CNN) with convolutional, max pooling, and normalizing layers; (2) VGG-16 [86], (3) ResNet-152 [42]. The initialization of inner problem is randomized for different meta-iterations to avoid the potential dependence on the quirks of particular settings. In detail, for all experiments we apply 50 meta-iterations and optimize inner problems using stochastic gradient descent, with batch size of 256. On CNN, 100 epochs for inner problem are used, which indicates 100 hyperparameters are involved. On VGG-16, the original model takes 224×224 images as inputs, and we adjust the size of the first fully-connected layer from 7×7 convolution to 1×1 to fit CIFAR-10 inputs. Here 20 epochs for inner are used. On ResNet-152, similar processing is exploited and the inner epoch is 10.

Results and discussions: The results are summarized in Figure 2. The experimental results show that the learning rates computed by HOZOG achieve the lowest test error and the fastest descending speed compared to baselines on all tasks. Moreover, the proposed method requires much less time to attain the best hyperparameters, and tends to have smaller variances in gradients. It is noteworthy that, some state-of-the-art hyperparameter optimization approaches (including HOAG, REV and FOR) are missing in this setting, due to the algorithms of REV and FOR are limited to smooth functions and the implementation of HOAG is limited to the hyperparameter optimization problems with a small number of hyperparamters. However, these difficulties are avoided by our HOZOG, which also demonstrates the flexibility of HOZOG. Moreover, as a brutal search method, the performance of RS is very unstable, which can be identified from the hyper-gradients. For BO and BOHB,

the instability also exists, potentially due to the highly complexity of the network structure. Another noteworthy problem with respect to BO and BOHB is the computational overhead in sampling, which make the meta-iteration extremely time consuming, compared to other methods.

We observe that the difficulty of this problem mainly comes from model complexity, instead of hyper-parameter numbers. For CNN with 100 hyper-parameters, HOZOG shows advantages in both time and suboptimality, although baselines can also efficiently find a reasonable solution. For VGG-16 and ResNet-152, we notice that though the size of hyperparameters is reduced, it takes baselines longer time to find acceptable results. Instead, HOZOG still shows fast convergence empirically. This observation indicates that HOZOG is potentially more suitable for hyperparameter optimization in large DNN.

3.3.3 Data Hyper-Cleaning

Experimental setup: We evaluate HOZOG on tuning the hyperparameters of data hyper-cleaning task. Compared with the preceding problems, the data cleaning task is more challenging, since it has more hyperparameters (hundreds or even thousands).

Assuming that we have a label noise dataset, with only limited clean data provided. The data hyper-cleaning task is to allocate a hyperparameter weight λ_i to a certain data point or a group of data points to counteract the influence of noisy samples. We split a certain data set into three subsets: $\mathcal{D}_{tr}$ of N_{tr} training samples, $\mathcal{D}_{val}$ of N_{val} validation samples and a test set $\mathcal{D}_t$ containing the N_t samples. We set random labels to $\lceil 0.5 * N_{tr} \rceil$ training examples, and select a random subset $\mathcal{D}_f$ from $\mathcal{D}_{tr}$.

Similar to [30], we considered a plain softmax regression model with parameters W (weights) and b (bias). The error of a model (W, b) on an example (x, y) was evaluated by using the cross-entropy $l(W, b, (x, y))$ both in the training objective function, L, and in the validation one, E. We added in L an hyperparameter vector $\lambda \in \mathbb{R}^{N_h}$ that weights each group of examples in the training phase through sigmoid function, *i.e.* $L(W, b) = \frac{1}{N_{tr}} \sum_{g \in \mathcal{G}} \sum_{i \in g} \text{sigmoid}(\lambda_g) l(W, b, (x_i, y_i))$, where $\mathcal{G}$ contain N_h groups random select from $\mathcal{D}_{tr}$. Thus, we have the hyperparameter optimization problem as follows.

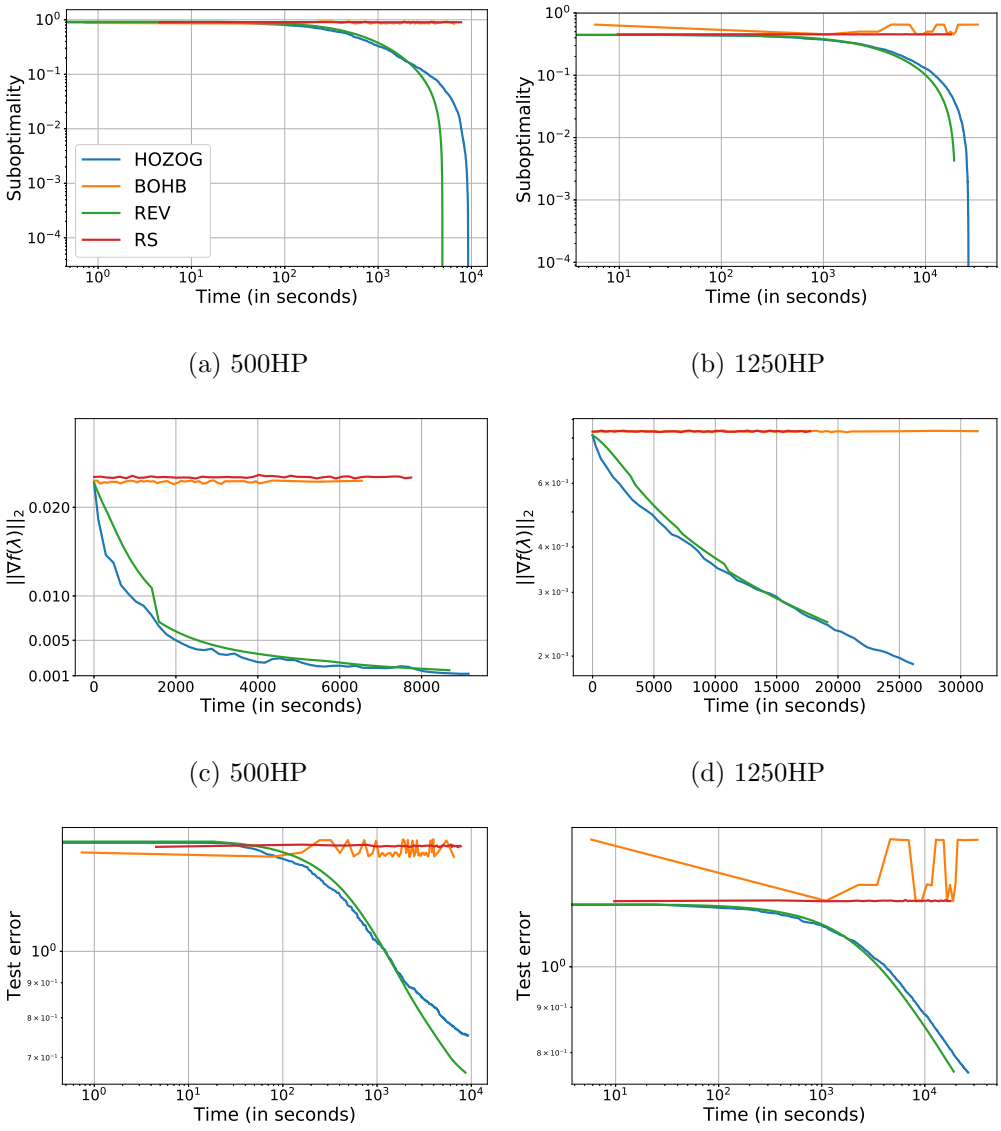
$$\arg\min_{\lambda \in \mathbb{R}^{N_h}} E(W(\lambda), b(\lambda)), \quad s.t. \;\; [W(\lambda), b(\lambda)] \approx \arg\min_{W,b} L(W, b) \tag{3-26}$$

We instance two subset dataset for the MNIST dataset, with $N_{tr} = 5000$, $N_{val} = 5000$, $N_t = 10000$, $N_h = 1250$ and $N_{tr} = 1000$, $N_{val} = 1000$, $N_t = 4000$, $N_h = 500$. We use a standard gradient descent method for the inner problem with fixed learning rate 0.05 and 4000 iteration. RS is used as baseline method, and BOHB and REV are used as comparison.

Results and discussions: Figure 3 presents the results of HOZOG, BOHB, REV and RS for data hyper-cleaning. Note that the methods of GPBO, FOR and HOAG are missing here, because the hyperparameter size is beyond the capability of their implementations. The results show that HOZOG can beat RS and BOHB easily, while not perform completely as good as REV in the long run. This is because REV is an exact gradient method whose convergence rate is faster than the one of zeroth-order gradient method (*i.e.*, HOZOG) by a constant whose value is depending on p [66]. However, computing the exact gradients in REV is costly. Specifically, REV takes about 40 seconds to finish the computation of one hyper-gradient under the setting of 1250 hyperparameters, which is only about 24 seconds for HOZOG. This is the reason why our method converges faster than REV in the early stage of training. Importantly, the application scenarios of REV are limited to smooth functions, *e.g.*, not suitable for the experimental settings of convolutional neural networks and deeper neural networks. However, our HOZOG can be utilized to a broader class of functions (*i.e.*, continuous functions).

3.3.4 Discussion: Importance of HOZOG

The experimental results show that the black-box optimization methods have a weak performance for the high-dimensional hyperparameter optimization problems which is also verified in a large number of existing references [10, 88], while they have the advantages of *simplicity* and *flexibility*. On the other hand, the existing gradient-based methods [30, 31] need experienced researchers to provide a customized program against the optimization algorithm and sometime it would fail, while they have the advantages of *scalability* and *efficiency*. HOZOG inherits all the benefits from both approaches in that, the gradients are computed in a black-box manner, while the hyperparameter search is accomplished via gradient descent. Especially, for high-dimensional hyperparameter optimization problems



(a) 500HP (b) 1250HP

(c) 500HP (d) 1250HP

(e) 500HP (f) 1250HP

Figure 3: Comparison of Different Hyperparameter Optimization Algorithms for Data Hyper-Cleaning. (a)-(b): Suboptimality. (c)-(d): $\|\nabla f(\lambda)\|_2$. (e)-(f): Test Error.

which have no customized RFHO algorithm, HOZOG currently is the only choice for this kind of problems to the best of our knowledge.

3.4 Proof

3.4.1 Proof of Theorem 4

Before proving Theorem 4, we first give Lemma 7.

Lemma 7. *Let f and g be a continuous function of $\mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}$, and let $w_0 \in \mathbb{R}^d$ and $\lambda_0 \in \mathbb{R}^p$. Assume that f and g are continuous at the points w_0 and λ_0, and let a be a real number. If $h = f(g(w, \lambda), \lambda)$, then h is continuous at w_0 and λ_0.*

Proof. Given $\delta' \in \mathbb{R}^d$ and $\delta \in \mathbb{R}^p$, according to the definition of continuous function in Definition 4, we have that

$$\begin{aligned} & \lim_{\delta' \to \mathbf{0}, \delta \to \mathbf{0}} h(w_0 + \delta', \lambda_0 + \delta) \\ = & \lim_{\delta' \to \mathbf{0}, \delta \to \mathbf{0}} f\left(g\left(w_0 + \delta', \lambda_0 + \delta\right), \lambda_0 + \delta\right) \\ = & f\left(\lim_{\delta' \to \mathbf{0}, \delta \to \mathbf{0}} g\left(w_0 + \delta', \lambda_0 + \delta\right), \lambda_0\right) \\ = & f(g(w_0, \lambda_0), \lambda_0) \end{aligned} \tag{3-27}$$

where the second equality uses the definition of continuous function in Definition 4. This completes the proof. □

If the hyperparameters λ are continuous and the mapping functions $\Phi_t(w, \lambda)$ (for every $t \in \{1, \ldots, T\}$) are continuous, the mapping function $\mathcal{A}(\lambda)$ is continuous, and the outer objective E is continuous, we have that the $\mathcal{A}$-based constrained optimization problem $f(\lambda)$ is continuous *w.r.t.* λ.

Proof. As defined in Definition 3, the mapping function is actually the function

$$\mathcal{A}(\lambda) = w_T = \Phi_T(\Phi_{T-1}(\ldots(\Phi_1(w_0, \lambda), \lambda), \ldots, \lambda) \tag{3-28}$$

Because each mapping function $\Phi_t(w, \lambda)$ is continuous *w.r.t.* w and λ, we can recursively use Lemma 7 to have that the mapping function $\mathcal{A}$ is continuous *w.r.t.* λ.

Because $f(\lambda) = E(w_T, \lambda)$ and the function $E(w, \lambda)$ is continuous *w.r.t.* w and λ, we have that the function $f(\lambda)$ is continuous *w.r.t.* λ according to Lemma 7. This completes the proof. □

3.4.2 Proof of Theorem 5

Before proving Theorem 5, we first give Lemma 8 which is provided in [26].

Lemma 8 ([26]). *If $f(\lambda) : \mathbb{R}^p \to \mathbb{R}$ is a Lipschitz continuous function. Then, its Lipschitz constant $L(f)$ is*

$$L(f) = \sup_{\lambda \in \mathbb{R}^p} \|\partial_\lambda f(f(\lambda))\|_2 \tag{3-29}$$

Given the continuous mapping functions $\Phi_t(w_{t-1}, \lambda)$ where $t \in \{1, \ldots, T\}$), $A_t = \frac{\partial \Phi_t(w_{t-1}, \lambda)}{\partial w_{t-1}}$, $B_t = \frac{\partial \Phi_t(w_{t-1}, \lambda)}{\partial \lambda}$. Given the continuous objective function $E(w_T, \lambda)$, $A_{T+1} = \frac{\partial E(w_T, \lambda)}{\partial w_T}$ and $B_{T+1} = \frac{\partial E(w_T, \lambda)}{\partial \lambda}$. Let $L_{A_t} = \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|A_{t+1}\|_2$, $L_{B_t} = \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|B_t\|_2$. Let $L(f)$ denote the Lipschitz constant of the continuous function $f(\lambda)$, we can upper bound $L(f)$ by $\sum_{t=1}^{T+1} L_{B_t} L_{A_{t+1}} \ldots L_{A_{T+1}}$.

Proof. Firstly, according to the chain rule [78], we give the computation of $\partial_\lambda f(f(\lambda))$ as follows.

$$\begin{aligned}
\partial_\lambda f(\lambda) &= \frac{\partial E(w_T, \lambda)}{\partial w_T} \frac{\partial w_T}{\partial \lambda} + \frac{\partial E(w_T, \lambda)}{\partial \lambda} \qquad (3\text{-}30)\\
&= A_{T+1} \frac{\partial w_T}{\partial \lambda} + B_{T+1} \\
&= A_{T+1} \left(\frac{\partial \Phi_T(w_{T-1}, \lambda)}{\partial w_{T-1}} \frac{\partial w_{T-1}}{\partial \lambda} + \frac{\partial \Phi_T(w_{T-1}, \lambda)}{\partial \lambda} \right) + B_{T+1} \\
&= A_{T+1} \left(A_T \frac{\partial w_{T-1}}{\partial \lambda} + B_T \right) + B_{T+1} \\
&= A_{T+1} A_T \frac{\partial w_{T-1}}{\partial \lambda} + A_{T+1} B_T + B_{T+1} \\
&= \sum_{t=1}^{T+1} B_t A_{t+1} \ldots A_{T+1}
\end{aligned}$$

Secondly, according to Lemma 8, we have that

$$
\begin{aligned}
& L(f) = \sup_{\lambda \in \mathbb{R}^p} \|\partial_\lambda f(\lambda)\|_2 \qquad (3\text{-}31)\\
= & \sup_{\lambda \in \mathbb{R}^p} \|\partial_\lambda f(\lambda)\|_2 \\
= & \sup_{\lambda \in \mathbb{R}^p} \left\| \sum_{t=1}^{T+1} B_t A_{t+1} \dots A_{T+1} \right\|_2 \\
\leq & \sum_{t=1}^{T+1} \sup_{\lambda \in \mathbb{R}^p} \|B_t A_{t+1} \dots A_{T+1}\|_2 \\
\leq & \sum_{t=1}^{T+1} \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|B_t\|_2 \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|A_{t+1}\|_2 \dots \sup_{\lambda \in \mathbb{R}^p, w \in \mathbb{R}^d} \|A_{T+1}\|_2 \\
\leq & \sum_{t=1}^{T+1} L_{B_t} L_{A_{t+1}} \dots L_{A_{T+1}}
\end{aligned}
$$

This completes the proof. □

3.5 Conclusion

Effectiveness, efficiency, scalability, simplicity and flexibility (*i.e.*, E2S2F) are important evaluation criteria for hyperparameter optimization methods. In this paper, we proposed a new hyperparameter optimization paradigm with zeroth-order hyper-gradients (HOZOG) which is the first method having all these benefits to the best of our knowledge. We proved the feasibility of using HOZOG to achieve hyperparameter optimization under the condition of Lipschitz continuity. The experimental results on three representative hyperparameter (the size is from 1 to 1250) optimization tasks not only verify the result in the feasibility analysis, but also demonstrate the benefits of HOZOG in terms of E2S2F, compared with the state-of-the-art hyperparameter optimization methods.

4.0 Fast Modal Regression With Robust Sampling

Modal regression has shown promising prediction ability and robustness to data with outliers or heavy-tailed noise. However, for large-scale data, there is still computational challenge for the implementation of modal regression. To address this issue, in this paper, we propose a new regularized modal regression model with robust sampling strategy. Unlike conventional sampling for large-scale least squares, our sampling probabilities are dependent on the robust loss function for learning the conditional mode. We provide theoretical analysis to support the proposed model: the approximation bound is established by error analysis with Rademacher complexity, and the robustness characterization is provided based on the finite sample breakdown point analysis. The experiments are conducted on both synthetic and real-word data sets and the empirical results demonstrate the promising performance of resulting estimator.

4.1 Introduction

Modal regression [80, 17, 53] has attracted much attention in statistical machine learning research, because the resulting estimator is more efficient and robust than ordinary least square-based estimation in the case of outliers or heavy-tail error distribution. Unlike conventional regression for learning conditional mean or median, modal regression focuses on estimating the conditional mode of a response Y given input $X = x$ [107, 106]. The mode can better reveal numerical characteristic of a statistical distribution or data set, which is usually missed by the traditional mean for data with outliers or the skewed noise distribution [15].

There are some theoretical studies for modal regression based on maximizing a conditional density or a jointed density, see, *e.g.*, [107, 106, 15, 81, 27]. Typical works include the parametric estimation in [106] and nonparametric estimate method in [15]. In [106], an expectation-maximization (EM) algorithm is proposed for modal linear regression and its

asymptotic properties are investigated. In [15], a local modal regression is proposed based on kernel density estimation and theoretical analysis is provided to characterize its asymptotic error bounds as well as techniques for constructing confidence sets and prediction sets. Recently, from the viewpoint of statistical learning, a novel modal regression algorithm is formulated in [27] under the empirical risk minimization (ERM) framework. In particular, theoretical foundations of the EMR-based modal regression are established in [27] including the approximation ability, robustness characterization, and relationship with the maximum correntropy criterion [60, 43, 28]. Besides the above theoretical analysis, there exist some application-oriented studies for modal regression in [53, 63, 23]. In particular, some empirical comparisons have been given in [63] for nonparametric forecasting problems via the conditional mean regression, the conditional median regression, and the conditional mode regression.

These studies push the progress of modal regression along the directions of both theoretical understanding and real-word applications. However, in large-scale data setting, the existing modal regression methods face the computational difficulty. This poses two important questions: *Can we reduce the computational burden of modal regression by developing sampling strategies used in large-scale least square regression? Can we establish statistical guarantees for the corresponding fast modal regression?* To answer the first question, we design a new robust sampling strategy with similar motivation in fast linear least-squares [21, 22, 61, 72], and then propose a regularized modal regression based on structural risk minimization. Since modal regression has not the solution representation as linear least-squares, we develop a robust solution-independent sampling to select important samples. This strategy not only can improve the computational feasibility and efficiency of modal regression, but also enhance the robustness of the gradient-based sampling in [112]. To reply the second question, we establish the approximation rate estimate to the conditional mode function and robustness analysis for the proposed method.

The main contribution of this paper is to propose a fast modal regression with robust sampling, and establish its asymptotic and robust analysis on function estimation. The current results fills the gap of modal regression for large-scale data computation and extends the gradient-based sampling from the conditional mean regression to the mode setting.

The rest of this paper is organized as follows. In section 4.2, we recall the background of modal regression from statistical learning aspect and formulate our sampling algorithm. Asymptotic theory and empirical evaluations of the proposed method are provided in Sections 4.4 and 4.6 respectively. Section 4.7 closes the paper with a brief conclusion.

4.2 Modal Regression With Robust Sampling

4.2.1 Modal Regression

Let $\mathcal{X} \subset \mathbb{R}^p$ be an input space and $\mathcal{Y} \subset \mathbb{R}$ be the corresponding output set. Assume that independent identical distributed (i.i.d) observations $\mathbf{z} = \{(x_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ are generated by

$$Y = f^*(X) + \varepsilon, \tag{4-32}$$

where $\text{mode}(\varepsilon|X = x) = \arg\max_t p_{\varepsilon|X}(t|X = x) = 0$ for any $x \in \mathcal{X}$ and $p_{\varepsilon|X}$ is the conditional density of ε conditioned on X. It is easy to see that

$$f^*(x) = \text{mode}(Y|X = x) = \arg\max_t p_{Y|X}(t|X = x), \forall x \in \mathcal{X},$$

where $p_{Y|X}$ is the conditional density of Y for given X. Denote ρ on $\mathcal{X} \times \mathcal{Y}$ as the intrinsic distribution for data generated from (4-32), and denote $\rho_{\mathcal{X}}$ as its marginal distribution on $\mathcal{X}$. For modal regression, the learning performance of a measurable prediction function $f : \mathcal{X} \to \mathcal{Y}$ can be measured by the modal regression criterion [27], which is defined as

$$\mathcal{R}(f) = \int_{\mathcal{X}} p_{Y|X}(f(x)|X = x) d\rho_{\mathcal{X}}(x). \tag{4-33}$$

It can be verified that the maximizer of $\mathcal{R}(f)$ over all measurable function is the target mode f^*. Hence, modal regression algorithms aim to construct a estimator $f : \mathcal{X} \to \mathbb{R}$ such that $\mathcal{R}(f)$ as large as possible. However, we can not get the estimator through $\mathcal{R}(f)$ directly, since the marginal distribution $\rho_{\mathcal{X}}$ and conditional density $p_{Y|X}$ are unknown in real-word applications. Fortunately, $\mathcal{R}(f)$ is equivalent to the density of variable $\varepsilon_f(x, y) = y - f(x)$

at the point 0 (see Theorem 5.1 in [27]). Then, based on kernel density estimation, we can introduce a empirical modal regression criterion

$$\mathcal{R}_{\mathbf{z}}^{\sigma}(f) = \frac{1}{n\sigma}\sum_{i=1}^{n}\phi(\frac{y_i - f(x_i)}{\sigma}),$$

where ϕ is a kernel representing function symmetric about 0 with width σ. As summarized in [27], many kernel functions can be used for density estimation, e.g., Gaussian kernel, Epanechnikov kernel, quadratic kernel, triweight kernel, and sigmoid function. With respect to $\mathcal{R}_{\mathbf{z}}^{\sigma}(f)$, we denote its expectation version as

$$\mathcal{R}^{\sigma}(f) = \frac{1}{\sigma}\int_{\mathcal{X}\times\mathcal{Y}}\phi(\frac{y - f(x)}{\sigma})d\rho(x,y).$$

It is worth noticing that the quantitative relationship between $\mathcal{R}(f)$ and $\mathcal{R}^{\sigma}(f)$ has been established in [27].

Different from previous studies for modal regression, we consider the learning setting with large scale data, where $n \gg p$. For the large-scale least squares, there are extensive discussions for subsampling strategies to improve their computational feasibility, e.g., the leverage-based sampling [22, 61], the column subset sampling [105], and the gradient-based sampling [112]. Since modal regression has not similar solution expression as linear least squares, we extend the gradient-based sampling [112] to the robust modal regression.

4.2.2 Fast Sampling Modal Regression

Integrating the sampling strategy in [112] and linear modal regression [107, 106], we perform the sampling modal regression through the following three steps:

Step 1. We get a subset S_0 with m_0 samples from $\mathbf{z}$ by uniform sampling and derive a pilot predictor $f_0(x) = w_0^T x$ with

$$w_0 = \arg\min_{w\in\mathbb{R}^p}\Big\{\frac{1}{m_0}\sum_{(x,y)\in S_0}(y - w^T x)^2\Big\}. \tag{4-34}$$

Step 2. Setting $\sigma = (\frac{1}{n}\sum_{i=1}^{n}(y_i - w_0^T x_i)^2)^{\frac{1}{2}}$. We get the right derivation $g_i = \frac{\partial_+\phi(\frac{y_i - w_0^T x_i}{\sigma})}{\partial w_0}$ for each $z_i = (x_i, y_i) \in \mathbf{z}$. The subset $S = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^{m} \subset \mathbf{z}$ is collected by sampling probabilities $\{\frac{\|g_i\|_2}{\sum\|g_i\|_2}\}_{i=1}^{n}$.

Step 3. The renew predictor $f_S = w_S^T x$, where w_S is obtained by the following regularized scheme

$$w_S = \arg\max_{w\in\mathbb{R}^p} \Big\{ \frac{1}{m}\sum_{i=1}^{m} \phi(\frac{\tilde{y}_i - w^T\tilde{x}_i}{\sigma}) - \lambda\sigma\|w\|_2^2 \Big\}. \tag{4-35}$$

Here, $\lambda > 0$ is a regularized parameter and usually chosen by cross-validation in applications.

Following the conjugated function theory [8] and half-quadratic optimization [68, 43], we know that the regularized scheme in (4-35) can be transformed as a optimization problem of iterative weighted least squares. Due to space limitation, we provide the optimization steps in the supplementary materials.

When the iterative time is T, the computation complexities for the first and the third steps are $O(m_0p^2)$ and $O(mp^2T)$, respectively. Considering the complexity $O(np)$ for the second step, we deduce that the total computation complexity is $O(\max\{np, m_0p^2, mp^2T\})$ for the above sampling modal regression.

In particular, for Gaussian kernel-based density estimation, we have

$$g_i = -2x_i(y_i - w_0^T x_i)\exp\Big\{ -\frac{(y_i - w_0^T x_i)^2}{\sigma^2}\Big\}$$

and

$$w_S = \arg\max_{w\in\mathbb{R}^p} \Big\{ \frac{1}{m}\sum_{i=1}^{m} \exp\Big\{ -\frac{(\tilde{y}_i - w^T\tilde{x}_i)^2}{\sigma^2}\Big\} - \lambda\sigma\|w\|_2^2 \Big\}.$$

Thus, the proposed sampling modal regression can be considered as sampling algorithm under maximum correntropy criterion [71, 60, 28]. That is to say our model contains the sampling correntropy regression as its special case.

4.3 Computing Algorithm

We apply half-quadratic (HQ) optimization [68] to solve the regularized modal regression problem. Given a convex problem $\min_s u(s)$, it is equivalent to optimize the following half-quadratic reformulation:

$$\min_{s,t} Q(s,t) + v(t), \forall s, t \in \mathbb{R}$$

The quadratic function $Q(s,t)$ and dual potential function $v : \mathbb{R} \to \mathbb{R}$ satisfiy:

$$u(s) = \min_t Q(s,t) + v(t), \forall s \in \mathbb{R}.$$

where v can be determined via convex conjugacy approach:

According to [8], $\forall f(a)$ is closed and convex, $\exists$ convex function $g(b)$, such that $f(a) = \max_b(ab - g(b))$, where g is the conjugate of f, *i.e.*, $g = f^\star and\, f = g^\star$. Also, the following statement is estabilshed:

$$\arg\max_b(ab - g(b)) = f'(a)$$

In [43],have been provided the optimization steps for Gaussian kernel setting. This optimization strategy could be employed to general kernels density estimation,as well. For example, a Epanechnikov kernel $\phi(e) = \frac{3}{4}(1 - e^2)I_{[|e|\leq 1]}$. Let us define a function f, which is convex and closed:

$$f(a) = \begin{cases} \frac{3}{4}(1-a), & 0 \leq a \leq 1 \\ 0, & a \geq 1. \end{cases}$$

As mentioned above, there must exist a convex function g such that $f(a) = \max_b(ab - g(b))$ and $\phi(e) = f(e^2) = \max_b(e^2 b - g(b))$. Therefore, our regularized modal regression can be rewritten as

$$\max_{w\in\mathbb{R}^p, b\in\mathbb{R}^n} \Big\{ \sum_{i=1}^n \Big(b_i (\frac{y_i - w^T x_i}{\sigma})^2 - g(b_i) \Big) - n\sigma\lambda \sum_{j=1}^p w_j^2 \Big\}. \tag{4-36}$$

The above minimization problem can be solved by alternating optimization algorithm.

4.4 Approximation and Robustness Analysis

This section provides our main theoretical results on the upper bound of $\mathcal{R}(f^*) - \mathcal{R}(f_S)$ and robustness characterization of f_S. Detail proofs are provided in the supplementary material.

4.4.1 Approximation Bound

Our approximation bound depends on the following conditions on f^*, ϕ, and $p_{\varepsilon|x}$. These assumptions have been used or discussed in [27].

Assumption 3. *Assume that $f^*(x) = \text{mode}(Y|X = x)$ is a linear function of x, that is $f^*(x) = w_*^T x$ for some $w_* \in \mathbb{R}^p$.*

Assumption 4. *Suppose that $\|x\|_2 \leq a < \infty$ for any $x \in \mathcal{X}$, and the representing function ϕ satisfies the following conditions:*

(1) $\forall u \in \mathbb{R}, \phi(u) \leq \phi(0) < \infty$, and ϕ is Lipschitz continuous with constant L_ϕ.

(2)$\int_{\mathbb{R}} \phi(u)du = 1$ and $\int_{\mathbb{R}} u^2\phi(u)du < \infty$.

The bounded input is a natural condition for linear regression [47, 48] and the restriction for ϕ holds true for many kernels for density estimation, e.g., Gaussian, Epanechnikov kernel, Quadratic kernel. To bridge $\mathcal{R}(f)$ and $\mathcal{R}^\sigma(f)$, we recall the following condition introduced in [27].

Assumption 5. *The conditional density $p_{\varepsilon|X}$ is second-order continuously differentiable and $\|p''_{\varepsilon|X}\|_\infty$ is bounded.*

Now, we state the upper bound on the excess modal error.

Theorem 6. *Let Assumptions 3-5 be true. Taking $\lambda\sigma^{\frac{5}{3}} = O(m^{-\frac{1}{3}})$ and $\sigma = O(m^{-\frac{1}{11}})$, we have*

$$\mathcal{R}(f^*) - \mathcal{R}(f_S) \leq C \log(4/\delta) m^{-\frac{2}{11}}$$

with confidence at least $1 - \delta$, where C is a positive constant independent of m, δ.

Theorem 6 demonstrates that the proposed method has convergence rate with polynomial decay. To best of our knowledge, this is the first learning theory analysis for sampling modal regression. Moreover, under Assumption 3 in [27], we can derive that $\|f_S - f^*\|^2_{L^2_{\rho_{\mathcal{X}}}} \to 0$ with approximation order $O(m^{-\frac{2}{11}})$. Now we give some comparisons for our result with the related analysis for modal regression [106, 27] and gradient-based sampling [112].

- The asymptotic analysis of modal regression has been established in [106] for EM algorithm and in [27] for empirical risk minimization method. Different from the previous works, our current result focuses on sampling regularized scheme for large scale data.
- The approximation ability of least-squares with gradient-based sampling has been presented in [112] under some eigenvalue condition for data-dependent matrix. Different from previous result measured by MSE criterion, our analysis is to characterize the approximation ability of estimator to the conditional mode function and independent of the eigenvalue condition. As a byproduct, our result also provides the convergence of sampling correntropy regression.

To proof Theorem 1, we first introduce a data-free intermediate function $f_\lambda = w_\lambda^T x$, where

$$w_\lambda = \arg\max_{w\in\mathbb{R}^p} \Big\{ \frac{1}{\sigma} \int_{\mathcal{X}\times\mathcal{Y}} \phi(\frac{y - w^T x}{\sigma}) d\rho(x,y) - \lambda \|w\|_2^2 \Big\}.$$

Then, we present the error decomposition on the excess error $\mathcal{R}(f^*) - \mathcal{R}(f_S)$ and bound the decomposed error terms based on analysis techniques in [27, 47, 48]. For the proof in the supplementary material, the Rademacher complexity [6] is used to measure the capacity of hypothesis function space.

4.4.2 Robustness Characterization

There are various notations for quantifying the algorithmic robustness, e.g., the influence function, the breakdown point, and the sensitivity curve [89, 27]. Here, we measure the robustness of f_S in (4-35) via its finite sample breakdown point [45, 46]. As illustrated in [46, 27], the finite sample breakdown can characterize the robustness of re-decreasing M-estimator and reflect the largest amount of contamination that an estimator can tolerate

before return arbitrary values. In particular, we do not require the boundedness of input to explore the robustness of f_S in terms of the finite sample breakdown point.

For given sampling set $S = \{(\tilde{x}_i, \tilde{y}_i)\}_{i=1}^m$, w_S is the responding estimator defined in (4-35). Let $S \cup S'$ be the corrupted training samples, where $S' = \{(\tilde{x}_{m+j}, \tilde{y}_{m+j})\}_{j=1}^k$ is arbitrary point set from $\mathcal{X} \times \mathcal{Y}$. Then, the finite sample contamination breakdown point of w_S is defined by

$$\epsilon(w_S) = \min_{1 \le k \le m} \Big\{ \frac{k}{m+k} : \sup_{S'} \|w_{S \cup S'}\|_2 = \infty \Big\},$$

where $w_{S \cup S'}$ denotes the estimator from (4-35) associated with $S \cup S'$.

Theorem 7. *Assume that $\phi(u) = \phi(-u)$ and $\lim_{t \to \infty} \phi(t) = 0$. Let*

$$M = \sum_{i=1}^m \frac{\phi(\frac{\tilde{y}_i - w_S^T \tilde{x}_i}{\sigma})}{\phi(0)} - \frac{\lambda \sigma}{\phi(0)} \|w_S\|_2^2.$$

Then the finite sample breakdown point of w_S in (4-35) is

$$\epsilon(w_S) = \frac{k^*}{m + k^*},$$

where $k^* \ge \lceil M \rceil$ and $\lceil M \rceil$ is the smallest integer not less than M.

Theorem 7 shows that the breakdown point of (4-35) depends on ϕ, σ, which is similar with the modal linear regression in [106] and empirical risk minimization in [27]. Our result extends the previous analysis in [106] to the regularized modal regression under structural risk minimization. Moreover, the steps 1 and 2 before (4-35) are useful to reduce the fraction of outliers in training samples, and hence improve the finite sample breakdown point. Therefore, the sampling procedures not only improve the computation feasibility for large scale data, but also strengthen the robustness of modal regression.

4.5 Proofs of Theorem 6 and Theorem 7

To establish the approximation estimation in Theorem 1, we introduce a data-free intermediate function $f_\lambda = w_\lambda^T x$, where

$$w_\lambda = \arg\max_{w\in\mathbb{R}^p} \Big\{ \frac{1}{\sigma} \int_{\mathcal{X}\times\mathcal{Y}} \phi(\frac{y - w^T x}{\sigma}) d\rho(x,y) - \lambda \|w\|_2^2 \Big\}.$$

The following decomposition on $\mathcal{R}(f^*) - \mathcal{R}(f_S)$ is key to our approximation analysis.

Proposition 2. *Under Assumptions 1-3, there holds*

$$\begin{aligned}
& \mathcal{R}(f^*) - \mathcal{R}(f_S) \\
\leq\ & \mathcal{R}^\sigma(f^*) - \mathcal{R}^\sigma(f_S) + \sigma^2 \|p''_{\varepsilon|X}\|_\infty \int_{\mathbb{R}} u^2 \phi(u) du \\
\leq\ & \mathcal{R}^\sigma(f_\lambda) - \mathcal{R}_S^\sigma(f_\lambda) + \mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S) \\
& + \sigma^2 \|p''_{\varepsilon|X}\|_\infty \int_{\mathbb{R}} u^2 \phi(u) du + \lambda \|w^*\|_2^2.
\end{aligned}$$

Proof. The first statement follows from Theorem 10 in [27]. Now, we consider the decomposition of $\mathcal{R}^\sigma(f^*) - \mathcal{R}^\sigma(f_S)$. We can deduce that

$$\begin{aligned}
& \mathcal{R}^\sigma(f^*) - \mathcal{R}^\sigma(f_S) \\
=\ & \mathcal{R}^\sigma(f^*) - \lambda \|w^*\|_2^2 - \mathcal{R}^\sigma(f_S) + \lambda \|w^*\|_2^2 \\
\leq\ & \mathcal{R}^\sigma(f_\lambda) - \lambda \|w_\lambda\|_2^2 - \mathcal{R}^\sigma(f_S) + \lambda \|w^*\|_2^2 \\
\leq\ & \mathcal{R}^\sigma(f_\lambda) - \mathcal{R}_S^\sigma(f_\lambda) \\
& + \Big\{ \mathcal{R}_S^\sigma(f_\lambda) - \lambda \|w_\lambda\|_2^2 - \big(\mathcal{R}_S^\sigma(f_S) - \lambda \|w_S\|_2^2 \big) \Big\} \\
& + \mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S) + \lambda \|w^*\|_2^2 \\
\leq\ & \mathcal{R}^\sigma(f_\lambda) - \mathcal{R}_S^\sigma(f_\lambda) + \mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S) + \lambda \|w^*\|_2^2,
\end{aligned}$$

where the first and second inequalities follow from the definitions of f_λ and f_S, respectively.

Combining the above decomposition with the first statement, we get the desired result. □

The error term $\mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S)$ can be bounded by the concentration estimate associated with Rademacher complexity. As a capacity measure of hypothesis function space, Rademacher complexity has been used extensively for error analysis of learning algorithms [6, 47, 48] .

Definition 6. *Let $\{x_i\}_{i=1}^m \in \mathcal{X}^m$ be independent samples selected according to μ and let $\mathcal{F}$ be a class of functions mapping from $\mathcal{X}$ to $\mathbb{R}$. Define the Rademacher complexity of $\mathcal{F}$ to be*

$$\mathcal{R}_m(\mathcal{F}) = E_\mu E_\epsilon\Big[\sup_{f\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m \epsilon_i f(x_i)\Big],$$

where $\{\epsilon_i\}_{i=1}^m$ are independent random variables uniformly chosen from $\{-1, 1\}$.

The uniform concentration inequality with Rademacher complexity has been provided in Theorem 8 [6]. In particular, some explicit versions are given in [47, 48] for linear function classes.

Lemma 9. *Assume that loss function $\psi(f, z)$ is L Lipschitz continuous with respect to f and $|\psi(f, z)| \leq c$ for any $z \in \mathcal{Z}$ and $f \in \mathcal{F}$. For any $\delta \in (0, 1)$, with confidence $1 - \delta$ there holds*

$$\frac{1}{m}\sum_{i=1}^m \psi(f, z_i) - E\psi(f, z) \leq 2L\mathcal{R}_m(\mathcal{F}) + c\sqrt{\frac{\ln(2/\delta)}{2m}}.$$

The error term $\mathcal{R}^\sigma(f_\lambda) - \mathcal{R}^\sigma_S(f_\lambda)$ can be estimated by the Bernstein inequality [**?** **?**].

Lemma 10. *Let ξ be a random variable on a probability space $\mathcal{Z}$ with mean $E\xi$ and variance ν. If $|\xi(z) - E\xi| \leq M_\xi$ for almost all $z \in \mathcal{Z}$, then with confidence $1 - \delta$*

$$E\xi - \frac{1}{m}\sum_{i=1}^m \xi(z_i) \leq \frac{2M_\xi \log(1/\delta)}{3m} + \sqrt{\frac{2\nu^2 \log(2/\delta)}{m}}.$$

Proposition 3. *Under Assumption 2, for any $\delta \in (0, 1)$, there holds*

$$\begin{aligned} & \mathcal{R}^\sigma(f_\lambda) - \mathcal{R}^\sigma_S(f_\lambda) + \mathcal{R}^\sigma_S(f_S) - \mathcal{R}^\sigma(f_S) \\ \leq\ & 2aL_\phi\sqrt{\frac{\phi(0)}{m\lambda\sigma^5}} + 4\phi(0)\sqrt{\frac{\ln(4/\delta)}{n\sigma^2}} \\ & + \frac{4\phi(0)\log(4/\delta)}{n\sigma} \end{aligned}$$

with confidence at least $1 - \delta$.

Proof. We first bound $\mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S)$. For any measurable functions f_1, f_2, and $(x, y) \in \mathcal{X} \times \mathcal{Y}$, there holds

$$\Big|\frac{1}{\sigma}\phi(\frac{y - f_1(x)}{\sigma}) - \frac{1}{\sigma}\phi(\frac{y - f_2(x)}{\sigma})\Big| \leq \sigma^{-2} L_\phi |f_1(x) - f_2(x)|.$$

This means $\sigma^{-1}\phi(\frac{y-f(x)}{\sigma})$ has Lipschitz constant $\sigma^{-2}L_\phi$. From Assumption 2, we know that $\sigma^{-1}\phi(\frac{y-f(x)}{\sigma}) \leq \sigma^{-1}\phi(0)$. Moreover, the definition f_S tells us that

$$\lambda \|w_S\|_2^2 \leq \mathcal{R}_S^\sigma(f_S) - \mathcal{R}_S^\sigma(0) \leq \sigma^{-1}\phi(0),$$

which implies $\|w_S\|_2 \leq \sqrt{\frac{\phi(0)}{\lambda\sigma}}$.

Denote

$$\mathcal{F} = \Big\{ f(x) = w^T x : \|w_S\|_2 \leq \sqrt{\frac{\phi(0)}{\lambda\sigma}}, \|x\|_2 \leq a \Big\}.$$

According to Theorem 3 in [47] (or Theorem 7 in [48]), we know that $\mathcal{R}_m(\mathcal{F}) \leq a\sqrt{\frac{\phi(0)}{m\sigma\lambda}}$. Applying Lemma 9 to $\psi(f, z) = \sigma^{-1}\phi(\frac{y-f(x)}{\sigma}), f \in \mathcal{F}$, we obtain that

$$\begin{aligned} \mathcal{R}_S^\sigma(f_S) - \mathcal{R}^\sigma(f_S) \quad &\leq \quad 2aL_\phi\sqrt{\frac{\phi(0)}{m\lambda\sigma^5}} + 2\phi(0)\sqrt{\frac{\ln(4/\delta)}{n\sigma^2}} \\ &\quad + \frac{4\phi(0)\log(4/\delta)}{n\sigma} \end{aligned} \tag{4-37}$$

with confidence at least $1 - \delta$.

Now we turn to bound $\mathcal{R}^\sigma(f_\lambda) - \mathcal{R}_S^\sigma(f_\lambda)$. Taking $\xi(x, y) = \sigma^{-1}\phi(\frac{y-f(x)}{\sigma})$, we know that $0 < \xi(x, y) \leq \sigma^{-1}\phi(0)$ and $|\xi - E\xi| \leq \sigma^{-1}\phi(0)$. By means of Bernstein inequality in Lemma 10, we have

$$\mathcal{R}^\sigma(f_\lambda) - \mathcal{R}_S^\sigma(f_\lambda) \leq \frac{4\phi(0)\log(2/\delta)}{3m\sigma} + \phi(0)\sqrt{\frac{2\ln(2/\delta)}{m}} \tag{4-38}$$

with confidence at least $1 - \delta$.

The desired upper bounds follows by combining (4-37) and (4-38). □

It is a position to present the proof of Theorem 1.

Proof of Theorem 6: Combining Propositions 2 and 3, we get with confidence at least $1 - \delta$

$$\mathcal{R}(f^*) - \mathcal{R}(f_S) \leq C\log(4/\delta)(\sigma^{-\frac{5}{2}}\lambda^{-\frac{1}{2}}m^{-\frac{1}{2}} + \sigma^2 + \lambda),$$

where C is a positive constant depending on $\phi(0), w^*, L_\phi$.

Setting $\lambda = \sigma^2 = \sigma^{-\frac{5}{2}}\lambda^{-\frac{1}{2}}m^{-\frac{1}{2}}$, we get $\sigma = m^{-\frac{1}{11}}$ and $\lambda = m^{-\frac{2}{11}}$. Then, with confidence at least $1-\delta$, we have

$$\mathcal{R}(f^*) - \mathcal{R}(f_S) \leq 3C\log(4/\delta)m^{-\frac{1}{11}}.$$

This completes this proof.

Our proof of Theorem 7 is inspired by the robustness analysis in [106].

Proof of Theorem 7: Observe that the regularized scheme of modal regression can be rewritten as the optimization problem

$$\max\Big\{\sum_{i=1}^{m}\frac{\phi(\frac{\tilde{y}_i - f(\tilde{x}_i)}{\sigma})}{\phi(0)} - \frac{\lambda\sigma\|w\|_2^2}{\phi(0)}\Big\}.$$

Denote $\phi^*(t) = \phi(t)/\phi(0)$. When $k < M$, there exists $k + m\zeta < M$ for some $\zeta > 0$. Let $\phi^*(t) \leq \zeta$ for $|t| \geq c$. Let w be any real vector such that $|y - w^T x| \geq c$ for any $(x, y) \in S$. Then,

$$\sum_{i=1}^{m+k}\phi^*(\tilde{y}_i - w_S^T\tilde{x}_i) - \frac{\lambda\sigma\|w_S\|_2^2}{\phi(0)} \geq M. \tag{4-39}$$

On the other hand, considering $\phi^*(t) \leq 1$ for any $t \in \mathbb{R}$, we have

$$\sum_{i=1}^{m+k}\phi^*(\tilde{y}_i - w^T\tilde{x}_i) - \frac{\lambda\sigma\|w\|_2^2}{\phi(0)} \leq k\zeta + m \leq M. \tag{4-40}$$

Combining inequalities (4-39) and (4-40), we have

$$\begin{aligned}&\sum_{i=1}^{m+k}\phi^*(\tilde{y}_i - w^T\tilde{x}_i) - \frac{\lambda\sigma\|w\|_2^2}{\phi(0)}\\ \leq\ &\sum_{i=1}^{m+k}\phi^*(\tilde{y}_i - w_S^T\tilde{x}_i) - \frac{\lambda\sigma\|w_S\|_2^2}{\phi(0)}.\end{aligned}$$

From the above relationship and the definition of $w_{S\cup S'}$, one knows that $w_{S\cup S'}$ must satisfy $|\tilde{y}_i - w_{S\cup S'}^T\tilde{x}_i| < c$ for at least one point in S. Therefore, $w_{S\cup S'}$ stays bounded no matter how S' varies, which means $k^* \geq \lceil M\rceil$.

4.6 Experimental Analysis

This section provides the empirical evaluation of our sampling method on both synthetic examples and real datasets. The following three sampling methods are used for comparison with our approaches:

- **Uniform Sampling (US)**. Uniformly select points from the total dataset, therefore, the sampling probability for the ith sample $\pi_i = \frac{1}{n}$.
- **Leverage-based Sampling (LS)**. A wide accepted sampling method, with the sampling probability $\pi_i \propto \mathbf{X}_i^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}_i$.
- **Gradient-based Sampling (GS)[112]**. This is a state-of-the-art sampling method, which is based on the gradient with least square loss.

4.6.1 Synthetic Data

Data Set: We construct the sample matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ by drawing elements independently from mixture Gaussian distribution as follows:

$$\mathbf{X}_{ij} \sim \frac{1}{2}\mathcal{N}(-u, \tau^2) + \frac{1}{2}\mathcal{N}(u, \eta^2\tau^2) \tag{4-41}$$

We set the sample size n as 20,000, and the number of variables p as 100 for all toy data. Following [112] ,we synthesize three kinds of toy dataset:

- **Toy-1**: $u = 0$ and $\eta = 1$, this is a standard Gaussian distribution.
- **Toy-2**: $u = 0$ and $\eta = 5$, this is a mixture Gaussian distribution with two variance.
- **Toy-3**: $u = -5$ and $\eta = 1$, this is a mixture Gaussian distribution with two peaks.

Given $\mathbf{X}$, we then generate the output $\mathbf{Y} \in \mathbb{R}^n$ by the following liner model:

$$\mathbf{Y} = \mathbf{X}\omega + \epsilon, \tag{4-42}$$

where $\omega \in \mathbb{R}^d$ is simulated from normal distribution $\mathcal{N}(0, 1)$, and ϵ denotes sample independent noise. To verify the robustness of our method, we consider four kinds of noise as below:

Table 10: Average MSE and Standard Deviation on Synthetic Data

Data	Kernel	Noise	US	LS	GS	Ours
Toy-1	Gaussian	Gaussian	4.8349±0.0796	4.8120±0.0821	4.7081±0.1183	**4.5083±0.1035**
		Chi-square	2.4734±0.0819	2.4680±0.0661	2.6903±0.1311	**2.3732±0.0697**
		Exponential	2.9665±0.0639	2.9734±0.0879	2.8941±0.1195	**2.7024±0.0675**
		Student's-t	2.4138±0.0931	2.4542±0.0919	2.6443±0.2945	**2.2544±0.0919**
	Sigmoid	Gaussian	5.0240±0.0666	5.0260±0.0758	4.7162±0.0945	**4.5397±0.0901**
		Chi-square	2.7509±0.0828	2.7429±0.0676	2.7033±0.1100	**2.3366±0.0668**
		Exponential	3.1510±0.0698	3.1204±0.0697	2.8428±0.0853	**2.6020±0.0843**
		Student's-t	2.7725±0.0956	2.7519±0.0933	2.6249±0.2015	**2.2229±0.0664**
Toy-2	Gaussian	Gaussian	4.8709±0.0825	4.8717±0.0888	4.6780±0.1080	**4.5529±0.1081**
		Chi-square	2.4966±0.0778	2.5273±0.0691	2.6930±0.1103	**2.3965±0.0859**
		Exponential	2.9446±0.0726	2.9584±0.0565	2.8889±0.1026	**2.6831±0.0394**
		Student's-t	2.4675±0.0763	2.4821±0.0723	2.6181±0.1611	**2.2531±0.0679**
	Sigmoid	Gaussian	5.0313±0.0577	5.0185±0.0580	4.7132±0.1008	**4.5254±0.0654**
		Chi-square	2.7783±0.0853	2.7925±0.0826	2.7639±0.2082	**2.3537±0.0808**
		Exponential	3.1786±0.0838	3.1524±0.0842	2.8994±0.1135	**2.6133±0.0935**
		Student's-t	2.7960±0.1272	2.7884±0.1344	2.7694±0.5226	**2.2593±0.1233**
Toy-3	Gaussian	Gaussian	4.9725±0.0750	4.9791±0.0876	4.6846±0.1110	**4.5338±0.0615**
		Chi-square	2.5907±0.0841	2.5999±0.1005	2.7708±0.1457	**2.4366±0.0854**
		Exponential	3.0206±0.0788	2.9943±0.0835	2.8780±0.0931	**2.6503±0.0635**
		Student's-t	2.5314±0.0641	2.6059±0.0787	2.6321±0.1527	**2.2828±0.0605**
	Sigmoid	Gaussian	4.8090±0.0630	4.8163±0.0647	4.6962±0.1236	**4.4520±0.0789**
		Chi-square	2.6280±0.0539	2.6352±0.0809	2.7690±0.1432	**2.3264±0.0472**
		Exponential	2.9715±0.0854	2.9918±0.0906	2.8513±0.0818	**2.6119±0.0712**
		Student's-t	2.6184±0.0853	2.6283±0.1073	2.9514±1.0399	**2.2650±0.0940**

- **Gaussian noise**: $\epsilon \sim \mathcal{N}(0, 2^2)$, a Gaussian distribution with mean 0, and variance 2^2.
- **Chi-square noise**: $\epsilon \sim \mathcal{X}^2(1)$, a Chi-square distribution with 1 degree of freedom.
- **Exponential noise**: $\epsilon \sim E(1.5)$, a Exponential distribution with rate parameter 1.5.
- **Student-T noise**: $\epsilon \sim \mathcal{T}(4)$, a Student-T distribution with 4 degrees of freedom.

Experiment results. We set the sampling rate at 0.05 for all datasets and evaluate these methods via average mean square error (MSE) and standard deviation with 20 repetitions.

As the current method just restricts the conditional density mode at zero, it can employ various kernels (*e.g.*, Gaussian kernel, Sigmoid kernel). The empirical results on toy data are reported in Table 1, which shows that our method outperforms the other sampling methods on prediction accuracy and stability.

Following [106], we further introduce the average and standard deviation of the coverage possibilities to better characterize the robustness of our model. The coverage possibility describes the average coverage probability (in 20 replicates) of data for prediction intervals with similar lengths centered around each estimated regression line. The Gaussian kernel is used for density estimation. Table 2 shows that our method has the highest coverage probability in all cases except some Gaussian noise data, where σ is defined in Step 2 in Section 2.2 for each training data. This empirical result further supports the robustness of our approach. As least squares under MSE criterion can be considered as optimal measure for data with Gaussian noise, it is reasonable that well performance of gradient-based sampling [112] for the Gaussian noise setting.

4.6.2 Real-World Data

We further evaluate our method on eight real-world datasets: *CASP, YearPrediction-MSD, CBMD-1, CBMD-2* from UCI (UCI); *cadata* from StatLib (StatLib); *cpusmall, letterscale, shuttlescale* from libSVM (libSVM).

Similar to toy data, we still calculate the average MSE and standard deviation over 20 repetitions for different sampling methods. Beside considering the conditional density estimation with Gaussian kernel, we also implement the proposed method with some non-Gaussian kernels (*e.g.*, Epanechnikov kernel, Quadratic kernel, Sigmoid kernel, and Logistic kernel). The experiment results are shown in Table 3, which demonstrate the competitive performance of our approach over the other baseline methods.

4.6.3 Running Time

As mentioned at the beginning, sampling can reduce computational burden of modal regression. To verify this motivation for algorithm design, we compare the running time

between our sampling modal regression and

modal regression with full samples (MR). All experiments are conducted on a quad-core Intel(R) i7 CPU @ 2.20GHz laptop with 16GB memory. The operating system is OS X 10.10.4 and the software we use is MATLAB R2016a (64-bit) 9.0.0.

Experimental results are shown in Figure 1. Note that there is a great gap, about e^5, between modal regression with whole data and sampling data, which represents more than 100 times running time difference. Hence, our method can improve the computation feasibility of modal regression efficiently.

4.7 Conclusion

This paper proposed a new fast modal regression algorithm with robust sampling strategy. Our method addresses the computational efficiency issue for large-scale data computation, where the existing modal regression approaches often fail due to the high computational cost. Our model is more robust than the sampling methods for least squares to handle the data with outliers, heavy-tailed noise, and skewed noise. Theoretical foundations have been provided for the proposed method, such as the approximation bound and robustness characterization. The extensive empirical evaluations are provided to support the promising performance of the proposed approach. Our theoretical analysis enriches the learning theory for robust sampling technique and fast modal regression.

Table 11: Average and Standard Deviation of Coverage Probability

Data	Width	Noise	US	LS	GS	Ours
Toy-1	0.01σ	Gaussian	0.0110±0.0009	0.0105±0.0013	0.0263±0.0012	**0.0309±0.0024**
		Chi-square	0.0221±0.0020	0.0225±0.0019	0.0262±0.0018	**0.0357±0.0024**
		Exponential	0.0191±0.0020	0.0188±0.0016	0.0253±0.0016	**0.0327±0.0027**
		Student's-t	0.0225±0.0016	0.0225±0.0023	0.0296±0.0020	**0.0368±0.0022**
	0.05σ	Gaussian	0.0542±0.0034	0.0536±0.0035	0.1332±0.0080	**0.1365±0.0079**
		Chi-square	0.1159±0.0107	0.1143±0.0090	0.1343±0.0091	**0.1620±0.0087**
		Exponential	0.0924±0.0072	0.0923±0.0064	0.1247±0.0047	**0.1470±0.0081**
		Student's-t	0.1097±0.0069	0.1129±0.0116	0.1565±0.0072	**0.1665±0.0061**
	0.1σ	Gaussian	0.0542±0.0034	0.0536±0.0035	0.1332±0.0080	**0.1365±0.0079**
		Chi-square	0.1159±0.0107	0.1143±0.0090	0.1343±0.0091	**0.1620±0.0087**
		Exponential	0.0924±0.0072	0.0923±0.0064	0.1247±0.0047	**0.1470±0.0081**
		Student's-t	0.1097±0.0069	0.1129±0.0116	0.1565±0.0072	**0.1665±0.0061**
	0.2σ	Gaussian	0.1082±0.0075	0.1091±0.0085	**0.2696±0.0143**	0.2597±0.0141
		Chi-square	0.2202±0.0161	0.2241±0.0195	0.2843±0.0116	**0.3130±0.0195**
		Exponential	0.1853±0.0171	0.1867±0.0147	0.2621±0.0098	**0.2747±0.0167**
		Student's-t	0.2139±0.0170	0.2211±0.0186	0.3019±0.0131	**0.3135±0.0172**
Toy-2	0.01σ	Gaussian	0.0108±0.0010	0.0110±0.0013	0.0262±0.0018	**0.0306±0.0017**
		Chi-square	0.0221±0.0021	0.0228±0.0021	0.0263±0.0010	**0.0360±0.0027**
		Exponential	0.0178±0.0015	0.0195±0.0018	0.0255±0.0016	**0.0327±0.0021**
		Student's-t	0.0217±0.0018	0.0221±0.0016	0.0312±0.0017	**0.0367±0.0023**
	0.05σ	Gaussian	0.0518±0.0045	0.0538±0.0048	0.1257±0.0053	**0.1316±0.0080**
		Chi-square	0.1088±0.0063	0.1091±0.0078	0.1265±0.0052	**0.1573±0.0067**
		Exponential	0.0920±0.0074	0.0907±0.0062	0.1205±0.0066	**0.1407±0.0078**
		Student's-t	0.1127±0.0076	0.1096±0.0062	0.1450±0.0076	**0.1599±0.0073**
	0.1σ	Gaussian	0.1041±0.0104	0.1110±0.0097	**0.2854±0.0107**	0.2645±0.0164
		Chi-square	0.2293±0.0136	0.2402±0.0210	0.3147±0.0121	**0.3287±0.0180**
		Exponential	0.1828±0.0206	0.1922±0.0161	0.2826±0.0092	**0.3001±0.0132**
		Student's-t	0.2192±0.0117	0.2239±0.0147	0.3232±0.0129	**0.3322±0.0146**
	0.2σ	Gaussian	0.2080±0.0161	0.2166±0.0195	**0.4987±0.0203**	0.4918±0.0255
		Chi-square	0.4243±0.0360	0.4468±0.0300	0.5833±0.0216	**0.5874±0.0188**
		Exponential	0.3536±0.0202	0.3616±0.0129	0.5217±0.0244	**0.5412±0.0287**
		Student's-t	0.4157±0.0281	0.4163±0.0269	0.5610±0.0136	**0.5618±0.0301**
Toy-3	0.01σ	Gaussian	0.0106±0.0009	0.0103±0.0010	0.0265±0.0017	**0.0305±0.0017**
		Chi-square	0.0219±0.0017	0.0226±0.0020	0.0255±0.0011	**0.0353±0.0020**
		Exponential	0.0182±0.0018	0.0178±0.0014	0.0242±0.0020	**0.0319±0.0024**
		Student's-t	0.0215±0.0023	0.0225±0.0026	0.0298±0.0020	**0.0356±0.0020**
	0.05σ	Gaussian	0.0530±0.0035	0.0533±0.0041	0.1215±0.0076	**0.1298±0.0074**
		Chi-square	0.1051±0.0073	0.1081±0.0073	0.1184±0.0048	**0.1522±0.0094**
		Exponential	0.0903±0.0086	0.0895±0.0082	0.1130±0.0057	**0.1400±0.0083**
		Student's-t	0.1071±0.0073	0.1071±0.0070	0.1381±0.0064	**0.1551±0.0075**
	0.1σ	Gaussian	0.1055±0.0060	0.1078±0.0096	**0.2552±0.0190**	0.2503±0.0150
		Chi-square	0.2199±0.0144	0.2165±0.0208	0.2704±0.0140	**0.2994±0.0152**
		Exponential	0.1779±0.0139	0.1830±0.0136	0.2507±0.0123	**0.2690±0.0174**
		Student's-t	0.2156±0.0184	0.2187±0.0146	0.2860±0.0094	**0.3002±0.0172**
	0.2σ	Gaussian	0.2043±0.0153	0.2096±0.0135	**0.4833±0.0231**	0.4646±0.0176
		Chi-square	0.4209±0.0237	0.4106±0.0314	0.5417±0.0253	**0.5750±0.0348**
		Exponential	0.3485±0.0237	0.3546±0.0203	0.4888±0.0246	**0.5155±0.0279**
		Student's-t	0.4073±0.0296	0.4041±0.0256	0.5293±0.0157	**0.5417±0.0231**

Table 12: Average MSE and Standard Deviation on Real-World Data

Data	Kernel	US	LS	GS	Ours
CASP	Gaussian	0.9308±0.2485	0.7726±0.0255	0.8149±0.0137	**0.7631±0.0184**
CBMD-1	Gaussian	0.5587±0.0810	0.6030±0.1440	0.5257±0.0119	**0.3620±0.0157**
CBMD-2	Gaussian	1.0308±0.0793	0.9603±0.0893	0.8104±0.0102	**0.4149±0.1096**
yearsMDS	Gaussian	0.8726±0.0091	0.8367±0.0115	0.8188±0.0022	**0.7730±0.0018**
Cadata	Epanechnikov	0.3937±0.0193	0.3958±0.0054	0.4064±0.0049	**0.3838±0.0126**
CPUsmall	Sigmoid	0.5529±0.2169	0.4631±0.1841	0.4250±0.0259	**0.3555±0.0750**
Letterscale	Logistic	0.8166±0.0548	0.7989±0.0423	0.7316±0.0062	**0.7284±0.0062**
shuttlescale	Quadratic	0.3452±0.1003	0.3891±0.0152	0.3961±0.0075	**0.3103±0.0075**

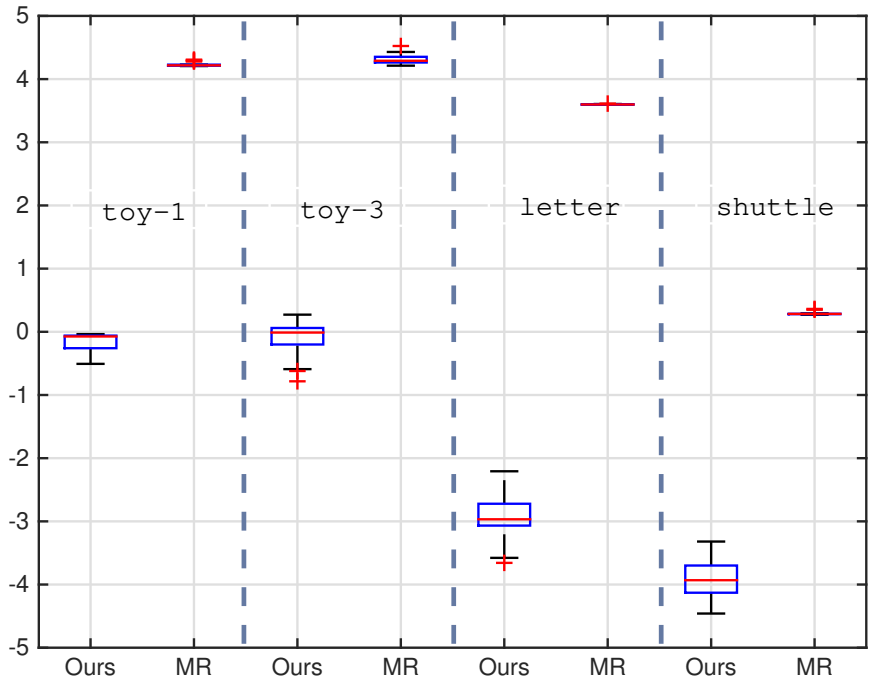


Figure 4: Boxplot of Logarithm of Different Average Running Time on Four Datasets

5.0 Conclusion

This body of work stands as a contribution to the evolving of machine learning, more specifically in tackling the challenges posed by nonlinear, high-dimensional data, complex models, and large-scale data processing.

A major focus of this research is to address the intrinsic nonlinearity inherent in machine learning data. Acknowledging that the flexibility of representation provided by nonlinear models can lead to a more accurate fit of data, we sought to balance this need for flexibility with the resulting increase in model complexity and loss of interpretability. To this end, we introduced a novel methodology, the Sparse Shrunk Additive Models (SSAM). This method is unique in that it leverages the structure information among features in high-dimensional nonparametric regression, allowing for flexible interactions among features. It essentially bridges the gap between sparse kernel regression and sparse feature selection, which previously stood as separate methodologies. Notably, we have established theoretical results regarding the convergence rate and sparsity characteristics of this novel method. These results were obtained by employing innovative analysis techniques incorporating integral operator and concentration estimate.

The complexity of nonlinear models often necessitates tuning of multiple, sometimes thousands, of hyperparameters. This process is pivotal in achieving model generalization. To streamline this process, we developed a new hyperparameter optimization method, the Zeroth-Order Hyper-Gradients (HOZOG). This part of our work was rooted in the need for a scalable, efficient method for hyperparameter optimization, and we provided the feasibility analysis of using HOZOG in such contexts. We proved that under the condition of Lipschitz continuity, HOZOG can effectively optimize hyperparameters, as confirmed by extensive experiments.

The handling of large-scale data presents a multitude of computational challenges that algorithms may struggle to address. To confront this, we proposed a novel regularized modal regression model that incorporates a robust sampling strategy. Our method deviates from conventional sampling for large-scale least squares. Instead, our sampling probabilities are

dependent on a robust loss function with the goal of learning the conditional mode. In support of this approach, we provided a comprehensive theoretical analysis that includes an approximation bound established via error analysis with Rademacher complexity. Furthermore, we offered a robustness characterization based on finite sample breakdown point analysis. The empirical results obtained from experiments conducted on synthetic and real-world datasets serve to further underscore the promising performance of the proposed estimator.

In summary, the thesis presented here offers some progress in machine learning methodologies, providing viable solutions for handling nonlinearity, model complexity, and the computational challenges of large-scale data. These methodologies, as they continue to evolve, hold promise for advancing some machine learning fields. I hope these works can stand as a step stone for future research and inspire new methodologies and enhancing our understanding of machine learning and its applications.

Bibliography

[1] Ahmed Alaoui and Michael W Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *NIPS*, 2015.

[2] Zeyuan Allen-Zhu and Elad Hazan. Variance reduction for faster non-convex optimization. In *International conference on machine learning*, pages 699–707, 2016.

[3] N Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[4] F.R. Bach. Consistency of the group lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008.

[5] F.R. Bach, G.R.G. Lanckrit, and M.I. Jordan. Multiple kernel learning, conic duality and the smo agorithm. In *ICML*, 2004.

[6] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002.

[7] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[8] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ. Press, Cambridge, U.K., 2004.

[9] Kristian Bredies and Dirk A Lorenz. Iterative soft-thresholding converges linearly. *arXiv preprint arXiv:0709.1598*, 2007.

[10] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[11] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. *Ann. Statist.*, 35(6):2313–2351, 2007.

[12] Hong Chen, Xiaoqian Wang, Cheng Deng, and Heng Huang. Group sparse additive machine. In *NIPS*, 2017.

[13] Hong Chen, Yulong Wang, Feng Zheng, Cheng Deng, and Heng Huang. Sparse modal additive model. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, In press.

[14] Hong Chen, Haifeng Xia, Weidong Cai, and Heng Huang. Error analysis of generalized nyström kernel regression. In *NIPS*, 2016.

[15] Y.C. Chen, C.R. Genovese, R. J. Tibshirani, and L. Wasserman. Nonparametric modal regression. *Ann. Statist.*, 44(2):489–514, 2016.

[16] Andreas Christmann and Ding Xuan Zhou. Learning rates for the risk of kernel-based quantile regression estimators in additive models. *Analysis and Applications*, 14(3):449–477, 2016.

[17] G. Collomb, W. Hardle, and S. Hassani. A note on prediction via estimation of the conditional mode function. *J. Statist. Plann. Infer.*, 15(2):227–236, 1987.

[18] Felipe Cucker and Ding Xuan Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

[19] Jianbo Cui, Jialin Hong, and Zhihui Liu. Strong convergence rate of finite difference approximations for stochastic cubic schrödinger equations ?? *Journal of Differential Equations*, 263:S002203961730253X, 2017.

[20] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in neural information processing systems*, pages 1646–1654, 2014.

[21] P. Dhillon, Y. Lu, D.P. Foster, and L. Ungar. New subsampling algorithms for fast least squares regression. In *NIPS*, pages 360–368, 2013.

[22] P. Drineas, M. Magdon-Ismail, M.W. Mahoney, and D.P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, 13:3475–3506, 2012.

[23] J. Einbeck and G. Tutz. Modelling beyond regression functions: an application of multimodal regression to speed-flow data. *J. Royal. Statist. Soc C.*, 55(4):461–475, 2006.

[24] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, pages 1436–1445, 2018.

[25] Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, pages 689–699, 2018.

[26] Herbert Federer. *Geometric measure theory*. Springer, 2014.

[27] Y. Feng, J. Fan, L. Shi, and J. Suykens. A statistical learning approach to modal regression. *arXiv:1702.05960v2*, 2017.

[28] Y. Feng, X. Huang, L. Shi, Y. Yang, and J. Suykens. Learning with the maximum correntropy criterion induced losses for regression. *J. Mach. Learn. Res.*, 16:993–1034, 2015.

[29] Yunlong Feng, Shaogao Lv, Hanyuan Hang, and Johan AK Suykens. Kernelized elastic net regularization: Generalization bounds, and sparse recovery. *Neural Comput.*, 28(3):525–562, 2016.

[30] Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1165–1173. JMLR. org, 2017.

[31] Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1563–1572, 2018.

[32] Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[34] G.R.G.Lanckrit, L.E Ghaoui N.Cristianini, P.Barlett, and M.I. Jordan. Learning the kernel matrix with semidefinite programming. *J. Mach. Learn. Res.*, 5:27–72, 2004.

[35] Bin Gu and Zhouyuan Huo. Asynchronous doubly stochastic group regularized learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*, 2018.

[36] Bin Gu, Zhouyuan Huo, Cheng Deng, and Heng Huang. Faster derivative-free stochastic algorithm for shared memory machines. In *International Conference on Machine Learning*, pages 1807–1816, 2018.

[37] Bin Gu and Charles Ling. A new generalized error path algorithm for model selection. In *International Conference on Machine Learning*, pages 2549–2558, 2015.

[38] Bin Gu, Guodong Liu, Yanfu Zhang, Xiang Geng, and Heng Huang. Optimizing large-scale hyperparameters via automated learning algorithm. *arXiv preprint arXiv:2102.09026*, 2021.

[39] Zhengchu Guo and Ding-Xuan Zhou. Concentration estimates for learning with unbounded sampling. *Adv. Comput. Math.*, 38(1):207–223, 2013.

[40] Trevor Hastie and Robert Tibshirani. *Generalized Additive Models*, volume 43. Chapman and Hall press, London, 1990.

[41] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York, 2001.

[42] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[43] R. He, W.S. Zheng, and B.G. Hu. Maximum correntropy criterion for robust face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(8):1561–1576, 2011.

[44] Jian Huang, Joel L Horowitz, and Fengrong Wei. Variable selection in nonparametric additive models. *Ann. Statist.*, 38(4):2282–2313, 2010.

[45] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[46] P.J. Huber. Finite sample breakdown of m-and p-estimators. *Ann. Statist.*, 12(1):119–126, 1984.

[47] S.M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

[48] S.M. Kakade, K. Sridharan, and A. Tewari. Regularization techniques for learning with matrices. *J. Mach. Learn. Res.*, 13:1865–1890, 2012.

[49] Kirthevasan Kandasamy and Yaoliang Yu. Additive approximation in high dimensional nonparametric regression via the salsa. In *ICML*, 2016.

[50] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[52] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the nyström method. *J. Mach. Learn. Res.*, 13:981–1006, 2012.

[53] Myoung-Jae Lee. Mode regression. *J. Econometrics*, 42(3):337–349, 1989.

[54] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

[55] Xiangru Lian, Huan Zhang, Cho-Jui Hsieh, Yijun Huang, and Ji Liu. A comprehensive linear speedup analysis for asynchronous stochastic parallel optimization from zeroth-order to first-order. In *Advances in Neural Information Processing Systems*, pages 3054–3062, 2016.

[56] Yi Lin and Hao Helen Zhang. Component selection and smoothing in smoothing spline analysis of variance models. *Ann. Statist.*, 34(5):2272–2297, 2006.

[57] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

[58] Guodong Liu. Effective medical code prediction via label internal alignment. *arXiv preprint arXiv:2305.05162*, 2023.

[59] Guodong Liu, Hong Chen, and Heng Huang. Sparse shrunk additive models. In *International Conference on Machine Learning*, pages 6194–6204. PMLR, 2020.

[60] W. Liu, P.P. Pokharel, and J.C. Príncipe. Correntropy: properties and applications in non-gaussian signal processing. *IEEE Trans. Signal Processing*, 55(11):5286–5298, 2007.

[61] P. Ma, M. Mahoney, and B. Yu. A statistical perspective on algorithmic leveraging. In *ICML*, pages 2061–2070, 2014.

[62] Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning*, pages 2113–2122, 2015.

[63] E. Matzner-Løfber, A. Gannoun, and J.G. De Gooijer. Nonparametric forecasting: a comparison of three kernel-based methods. *Commun. Stat. Theory Methods*, 27(7):1593–1617, 1998.

[64] Lukas Meier, Sara Van de Geer, and Peter Bühlmann. High-dimensional additive modeling. *Ann. Statist.*, 37(6B):3779–3821, 2009.

[65] Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[66] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[67] Weilin Nie and Cheng Wang. Constructive analysis for coefficient regularization regression algorithms. *Journal of Mathematical Analysis and Applications*, 431(2):1153–1171, 2015.

[68] M. Nikolova and M.K. Ng. Analysis of half-quadratic minimization methods for signal and image recovery. *SIAM J. Sci. Comput.*, 27(3):937–966, 2005.

[69] Fabian Pedregosa. Hyperparameter optimization with approximate gradient. In *International Conference on Machine Learning*, pages 737–746, 2016.

[70] Iosif Pinelis. Optimum bounds for the distributions of martingales in banach spaces. *The Annals of Probability*, pages 1679–1706, 1994.

[71] J.C. Príncipe. *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*. Springer, New York, 2010.

[72] G. Raskutti and M.W. Mahoney. A statistical perspective on randomized sketching for ordinary least-squares. In *ICML*, pages 2061–2070, 2015.

[73] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax-optimal rates for sparse additive models over kernel classes via convex programming. *J. Mach. Learn. Res.*, 13:389–427, 2012.

[74] Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *J. Royal. Statist. Soc B.*, 71(5):1009–1030, 2009.

[75] Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International conference on machine learning*, pages 314–323, 2016.

[76] Volker Roth. The generalized lasso. *IEEE Trans. Neural Networks*, 15(1):16–28, 2004.

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

[78] Walter Rudin. *Principles of mathematical analysis*. 1976.

[79] Walter Rudin et al. *Principles of mathematical analysis*, volume 3. McGraw-hill New York, 1964.

[80] T.W. Sager and R.A. Thisted. Maximum likelihood estimation of isotonic modal regression. *Ann. Statist.*, 10(3):690–707, 1982.

[81] H. Sasaki, Y. Ono, and M. Sugiyama. Modal regression via direct log-density derivative estimation. In *International Conference on Neural Information Processing*, pages 233–240. Springer, 2015.

[82] Bernhard Scholköpf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2001.

[83] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge university press, 2004.

[84] Lei Shi. Learning theory estimates for coefficient-based regularized regression. *Appl. Comput. Harmon. Anal.*, 34(2):252–265, 2013.

[85] Lei Shi, Yunlong Feng, and Ding Xuan Zhou. Concentration estimates for learning with ℓ_1-regularizer and data dependent hypothesis spaces. *Appl. Comput. Harmon. Anal.*, 31(2):286–302, 2011.

[86] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[87] Steve Smale and Ding Xuan Zhou. Learning theory estimates via integral operators and their approximations. *Constr. Approx.*, 26(2):153–172, 2007.

[88] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[89] I. Steinwart and A. Christmann. *Support Vector Machines*. Springer, New York, 2008.

[90] Ingo Steinwart, Don Hush, and Clint Scovel. Optimal rates for regularized least squares regression. In *COLT*, 2009.

[91] Hongwei Sun and Qiang Wu. Least square regression with indefinite kernels and coefficient regularization. *Appl. Comput. Harmon. Anal.*, 30(1):96–109, 2011.

[92] Hongwei Sun and Qiang Wu. Sparse representation in kernel machines. *IEEE Transactions on Neural Networks and Learning Systems*, 26(10):2576–2582, 2015.

[93] Robert Tibshirani. Regression shrinkage and selection via the lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

[94] Zhidong Tu et al. Integrative analysis of a cross-loci regulation network identifies app as a gene regulating insulin secretion from pancreatic islets. *PLoS Genet*, 8(12):e1003107, 2012.

[95] Hemant Tyagi, Anastasios Kyrillidis, Bernd Gärtner, and Andreas Krause. Learning sparse additive models with interactions in high dimensions. In *AISTATS*, 2016.

[96] AW Van der Vaart and Jon Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

[97] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[98] Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Advances in Neural Information Processing Systems*, pages 3835–3844, 2018.

[99] Cheng Wang and Ding-Xuan Zhou. Optimal learning rates for least squares regularized regression with unbounded sampling. *J. Complexity*, 27(1):55–67, 2011.

[100] Xiaoqian Wang, Hong Chen, Weidong Cai, Dinggang Shen, and Heng Huang. Regularized modal regression with applications in cognitive impairment prediction. In *Advances in neural information processing systems*, pages 1448–1458, 2017.

[101] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82, 1997.

[102] Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Multi-kernel regularized classifiers. *J. Complexity*, 23(1):108–134, 2007.

[103] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[104] Lei Yang, Shaogao Lv, and Junhui Wang. Model-free variable selection in reproducing kernel hilbert space. *J. Mach. Learn. Res.*, 17:1–24, 2016.

[105] T. Yang, L. Zhang, R. Jin, and S. Zhu. An explicit sampling dependent spectral error bound for column subset selection. In *ICML*, pages 2061–2070, 2015.

[106] W. Yao and L. Li. A new regression model: modal linear regression. *Scandinavian Journal of Statistics*, 41(3):656–671, 2014.

[107] W. Yao, B.G. Lindsay, and R. Li. Local modal regression. *Journal of Nonparametric Statistics*, 24(3):647–663, 2012.

[108] Junming. Yin, Xi Chen, and Eric P. Xing. Group sparse additive models. In *ICML*, 2012.

[109] Ming Yuan and Ding Xuan Zhou. Minimax optimal rates of estimation in high dimensional additive models. *Ann. Statist.*, 44(6):2564–2593, 2016.

[110] Tuo Zhao and Han Liu. Sparse additive machine. In *AISTATS*, pages 1435–1443, 2012.

[111] Tuo Zhao, Mo Yu, Yiming Wang, Raman Arora, and Han Liu. Accelerated mini-batch randomized block coordinate descent method. In *Advances in neural information processing systems*, pages 3329–3337, 2014.

[112] R. Zhu. Gradient-based sampling: An adaptive importance sampling for least-squares. In *NIPS*, pages 406–414, 2016.

[113] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.