Journal of Machine Learning Research 27 (2026) 1-38 Submitted 3/23; Revised 12/25; Published 5/26

Knowledge Cascade: Reverse Knowledge Distillation on Nonparametric Multivariate Functional Estimation

Luyang Fang* LUYANG.FANG@UGA.EDU

Haoran Lu* HAORAN.LU@UGA.EDU

Yongkai Chen YONGKAI.CHEN@UGA.EDU

Wenxuan Zhong† WENXUAN@UGA.EDU

Ping Ma† PINGMA@UGA.EDU

Department of Statistics
University of Georgia, Athens, GA, 30602, USA

Editor: Ji Zhu

Abstract

As machine learning models and datasets continue to grow, developing complex models has become increasingly computationally demanding. Knowledge distillation reduces deployment cost by compressing a large, well-trained teacher model into a compact student model, but it does not address settings where constructing the teacher itself is the bottleneck. Motivated by this challenge, we introduce Knowledge Cascade, a reverse knowledge distillation framework that uses information from a small, inexpensive student model to guide the development of a more complex teacher model. Although this direction is counterintuitive because the teacher typically has greater representational capacity, we show that student-to-teacher transfer can be principled when supported by statistical scaling relationships. We first develop Knowledge Cascade for nonparametric multivariate functional estimation in reproducing kernel Hilbert spaces via smoothing splines, where selecting multiple smoothing parameters is a major computational bottleneck. Knowledge Cascade transfers student-selected smoothing parameters to the full-sample regime through asymptotic scaling laws, substantially reducing computational cost for high-dimensional and large-scale datasets while retaining theoretical guarantees. Beyond smoothing splines, we illustrate the same principle through kernel density estimation and deep learning hyperparameter transfer. Simulations and real-data experiments show that Knowledge Cascade achieves substantial computational savings while maintaining strong statistical performance, and can sometimes outperform the corresponding full-sample procedure.

Keywords: nonparametric functional estimation, RKHS, deep learning, computationally efficient learning, hyperparameter scaling, convergence

1. Introduction

Machine learning models have recently seen a rapid increase in complexity, particularly with the success of large-scale deep neural networks across language, vision, multimodal

∗. Equal contribution.
†. Corresponding authors.

learning, and generative modeling (Hinton et al., 2012; Wolf et al., 2020; Hoffmann et al., 2022; Dehghani et al., 2023; Achiam et al., 2023; Grattafiori et al., 2024). Despite their effectiveness, developing such models often requires substantial computational resources and human effort, especially in the presence of large data volumes, high dimensionality, and complex data structures (Strubell et al., 2019; Hoffmann et al., 2022; Fang et al., 2025). As models and datasets continue to grow, it becomes increasingly important to develop principled strategies that use inexpensive preliminary models to guide the development of more complex models.

Traditionally, knowledge distillation (KD) has served as a widely used framework for mitigating computational costs (Hinton et al., 2015). In the standard KD paradigm, a large and well-trained teacher model transfers information to a smaller student model, allowing the student to retain much of the teacher's predictive performance with lower computational cost (Zhang et al., 2019; Gou et al., 2021; Fang et al., 2024; Ma et al., 2026; Fang et al., 2026a,b). While this teacher-to-student paradigm has been highly effective for model compression and efficient deployment, it leaves a critical bottleneck unresolved: standard KD relies on the prerequisite of a fully trained, highly capable teacher model, offering no relief for the exorbitant initial cost of developing the teacher itself.

To address this fundamental limitation in the training phase, we ask whether the mechanics of knowledge transfer can be inverted. Specifically, can a small and computationally efficient student model provide useful information for *training* a larger teacher model? We propose *Knowledge Cascade* (KCas), a reverse knowledge distillation framework in which information learned from a student model is transferred upward to guide the teacher model. In KCas, the student is not intended to replace the teacher; rather, it serves as a low-cost preliminary model that extracts useful information for the teacher at a substantially reduced computational cost. While transferring knowledge upward to a model with strictly greater representational capacity may initially seem counterintuitive, the key observation behind KCas is that the student does not need to learn everything the teacher will eventually learn. It only needs to extract transferable information that remains structurally useful when moving from a simpler setting to a more complex one. This dynamic is analogous to a statistical pilot study, where a small preliminary experiment cannot replace the full study but provides valuable guidance for conducting it more efficiently. KCas exploits this principle by learning from a low-cost student model and transferring the insights to the teacher through statistical theory or empirical scaling laws.

We formalize this principle first within the context of nonparametric multivariate functional estimation in reproducing kernel Hilbert spaces (RKHS). In this setting, a subsample-based estimator naturally serves as the student and the full-sample estimator as the teacher, since the effective complexity of the estimator grows with sample size (Gu, 2013). The main computational bottleneck is selecting multiple smoothing parameters, whose number and search cost increase rapidly with the number of predictors and interaction terms. KCas addresses this bottleneck by selecting smoothing parameters on a small subsample and transferring them to the full sample through asymptotic sample-size scaling. The teacher model is then fitted on the full data using these transferred parameters, avoiding costly full-sample tuning while retaining the statistical benefit of using all observations. Our theory and experiments show that KCas substantially reduces computation and can sometimes improve estimation accuracy by avoiding unstable or overly adaptive full-sample tuning.

While smoothing spline ANOVA models provide the primary theoretical anchor for this paper, they represent just one concrete instance of a fundamental principle: whenever an appropriate scaling relationship exists, information learned from a low-cost student model can be transferred to guide a more expensive teacher model. To demonstrate this versatility, we extend the KCas to kernel density estimation, where bandwidths follow classical asymptotic scaling laws, and to deep learning, where hyperparameters selected on compact student networks can be used to guide larger teacher networks. Ultimately, these extensions position KCas not merely as a specialized technique for RKHS, but as a general, theoretically grounded paradigm for reverse knowledge transfer across diverse machine learning domains.

Our contributions:

1. We introduce Knowledge Cascade (KCas), a reverse knowledge distillation framework in which small, computationally efficient student models guide larger and more complex teacher models, providing a new perspective beyond conventional teacher-to-student distillation.

2. We show that student-to-teacher knowledge transfer can be principled when supported by asymptotic scaling laws, using nonparametric multivariate functional estimation theory to extrapolate information from student models to teacher models.

3. We develop KCas for smoothing spline ANOVA models, design the associated algorithm, and establish consistency theory. Under the proposed subsampling scheme, KCas reduces the computational complexity of smoothing-parameter selection from $O(n^3)$ to $O(n^{\frac{3}{4}})$ while preserving statistical accuracy.

4. We illustrate the broader applicability of KCas beyond smoothing splines through kernel density estimation and deep learning hyperparameter transfer.

5. Through extensive simulations and real-data experiments, we show that KCas achieves substantial computational savings and can even outperform full-sample tuning in some settings.

2. Related Work

An idea related to KCas is self-distillation (SD), which is also developed to deploy complex models effectively. SD methods use the same architecture for both the teacher and student models, and facilitate the training procedure by letting the knowledge be transferred/exchanged among a group of models or within a single model (Zhang and Sabuncu, 2020; Mobahi et al., 2020; Zhang et al., 2019; Phuong and Lampert, 2019; Yang et al., 2019; Hou et al., 2019; Lan et al., 2018). However, models in standard SD methods are still relatively large, and the model training procedure is accelerated and improved by distilling knowledge from itself. In this sense, KCas differs from SD by using information from some 'actually small' student models that are much easier to train. In scientific scenarios where the pilot study is needed to determine the design of extensive and detailed experiments, KCas can use the pilot data to construct student models to avoid wasting valuable data in the pilot study. Thus, KCas is highly desirable in these settings. Note that in Yuan et al. (2020), the authors also discussed the possibility of reversing the knowledge distillation procedure,

but their methodology is still under the SD framework. Yuan et al. (2020) reverses the KD procedure as a motivating example for proposing the Teacher-free Knowledge Distillation (Tf-KD) framework. They prove the equivalence between KD and label smoothing regularization in a certain sense, and using this fact, Tf-KD lets a student model learn from itself or manually designed regularization distribution. Therefore, the student model in Tf-KD serves the purpose of regularization, while the student model in KCas serves the role of extracting information, and KCas amplifies this information to help the teacher model. Thus, our proposed KCas and the associated theories are significantly different from Yuan et al. (2020) and various self-distillation methods.

Further, Xie et al. (2020) proposes a similar idea with KCas of distilling the knowledge from the student model to help train the teacher model. They first train a noisy student model on the available labeled data, which is then used to generate pseudo-labels for the unlabeled data. Subsequently, the teacher model is then trained on the combined set of labeled and pseudo-labeled data. On the contrary, our research focuses on addressing situations where the model training process requires an exceptionally high computational burden or when completing the necessary computations becomes impractical. Therefore, Xie et al. (2020) addresses scenarios where the training model has sufficient computational resources but insufficient data, whereas our work focuses on situations where there is abundant data but limited computational resources available for training the model.

From a methodological perspective, KCas can be viewed as a strategy for efficient hyperparameter selection. Recent advancements in hyperparameter selection techniques continue to refine the efficacy of model selection. General approaches of hyperparameter selection include the classic cross-validation (Stone, 1974, 1978), criterion-based methods (Akaike, 1998; Schwarz, 1978), and a comprehensive review of recent developments can be found in Bischl et al. (2023). In the context of smoothing spline models, a series of methods based on generalized cross-validation (GCV) has been developed (Gu, 2014; Gu and Xiang, 2001; Hall, 1990; Wahba, 1985; Gu and Wahba, 1991). In this paper, we propose to conduct hyperparameter selection via the idea of knowledge cascade.

3. Preliminaries

In this section, we provide the necessary background of nonparametric functional estimation in reproducing kernel Hilbert space before introducing our proposed method. To estimate a function of interest η on a generic domain $\mathcal{X}$, we consider the nonparametric penalized loss functional,

$$PL = L(\eta) + \lambda J(\eta), \tag{1}$$

where $L(\eta)$ is the goodness-of-fit (loss) functional and $J(\eta)$ is the smoothness (penalty) functional. The smoothing parameter λ controls the trade-off between the smoothness of $\eta(x)$ and its fidelity to the data.

Functional ANOVA Decomposition. The estimation of function η on the product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$ has long been a crucial problem in statistical learning, with numerous proposed methods over the years (Jeon and Lin, 2006; Chen et al., 2016; Lin and Zhang, 2006; Pérez et al., 2009; Bosq, 2012). However, most of them have been challenged by the curse of dimensionality, as the estimation of multivariate functions is intrinsically difficult. To overcome this challenge, one effective approach is to decompose multivariate functions

using techniques similar to the classical analysis of variance (ANOVA) decomposition and its associated notions of the main effect and interaction (Gu et al., 2013; Gu and Wang, 2003; Kim and Gu, 2004; Huang, 1998; Jeon and Lin, 2006). In this functional ANOVA model, higher-order interactions are often excluded in practical estimation to control model complexity; excluding all interactions yields the popular additive models. On the product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$, the function η can be decomposed as a sum of a constant term, one-dimensional functions (main effects), two-dimensional functions (two-way interactions), and so on, as in the following decomposition,

$$\eta(x) = \eta\left(x_{\langle 1\rangle}, \ldots, x_{\langle d\rangle}\right) = \eta_{\emptyset} + \sum_{j} \eta_j\left(x_{\langle j\rangle}\right) + \sum_{j<k} \eta_{j,k}\left(x_{\langle j\rangle}, x_{\langle k\rangle}\right) + \ldots, \tag{2}$$

with the constant in $\eta_{\emptyset}$, the main effects in η_j, the two-way interactions in $\eta_{j,k}$, etc. Higher-order interactions are eliminated to ease the curse of dimensionality.

Reproducing Kernel Hilbert Space. By adding the smoothness penalty $J(\eta)$ to $L(\eta)$ in loss (1), we consider the space $\mathcal{H} \subseteq \{\eta : J(\eta) < \infty\}$ in which $J(\eta)$ is a square semi-norm with a finite-dimensional null space $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$. To assist analysis and computation, a metric and geometry should be defined in this space, and the loss (1) needs to be continuous in η under this metric. Since the reproducing kernel Hilbert spaces (RKHSs) are well suited for this purpose, we consider the space $\mathcal{H}$ as an RKHS with the continuous evaluation $[x]f = f(x)$, reproducing kernel $R(\cdot, \cdot)$, a non-negative definite function satisfying $\langle R(x, \cdot), f(\cdot)\rangle = f(x), \forall f \in \mathcal{H}$, where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathcal{H}$. The existence of the minimizer in RKHS is guaranteed by Wahba (1990). Details are discussed in Appendix A.

Density Estimation. Consider the situation that we have independently identically distributed (i.i.d.) data points $x_i,\ i = 1, \cdots n$, from an underlying data distribution $p(x)$ on a bounded domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$. We aim to estimate $p(x)$ based on observations x_i. For the nonparametric setting, a naive maximum likelihood density estimation is meaningless without any nonintrinsic constraint, since it will fit a sum of delta function spikes at the sample points x_i, which is obviously not an appealing estimate when the domain X is continuous. Thus, a penalized likelihood estimate (PLE) is a good candidate. Two intrinsic constraints come from the definition of a probability density that $p(\cdot) \geq 0$ and $\int_{\mathcal{X}} p dx = 1$. Since these two constraints are difficult to handle directly in computation, a common approach (Gu and Qiu, 1993; Silverman, 1982) is to estimate the log-density $\eta(\cdot)$, which is free of the constraints through the transformation

$$p(x) = \frac{e^{\eta(x)}}{\int_{\mathcal{X}} e^{\eta(x)} dx}.$$

Silverman (1982) proposed and studied the theoretical properties of the penalized likelihood over a Hilbert space $\mathcal{H}$:

$$-\frac{1}{n}\sum_{i=1}^{n} \eta\left(x_i\right) + \log \int_{\mathcal{X}} e^{\eta(x)} dx + \frac{\lambda}{2} J(\eta). \tag{3}$$

Nonparametric Regression. Consider the exponential family with the densities of the form

$$f(y \mid x) = \exp\{\frac{y\eta(x) - h(\eta(x))}{a(\phi)} + c(y, \phi)\},$$

where $a(\cdot) > 0$, h, and c are known functions, $\eta(x)$ is the regression function via the link η, and ϕ is the parameter that is independent of x. Observing $Y_i \mid x_i \sim f(y \mid x_i)$, $i = 1, \cdots, n$, we estimate $\eta(x)$ via the penalized likelihood functional

$$-\frac{1}{n}\sum_{i=1}^{n}\left\{Y_i\eta\left(x_i\right) - h\left(\eta\left(x_i\right)\right)\right\} + \frac{\lambda}{2}J(\eta), \tag{4}$$

where the term $c(y, \phi)$ is dropped as it is independent of $\eta(x)$, and $a(\phi)$ is absorbed into λ.

4. Methodology

Nonparametric penalized estimation of the function of interest η is a general question in lots of fields (Sun et al., 2016; Helwig et al., 2016). However, the computational burden of training complex models limits the applicability of many existing methods to large datasets. To tackle this challenge, we propose a reverse version of knowledge distillation, named knowledge cascade, by cascading the knowledge learned from a student model (trained on a small sample) to the teacher model (trained on a large sample). Specifically, we illustrate our method in the context of nonparametric functional estimation, with two important cases: density estimation and regression functional estimation.

4.1 Minimizer of the Penalized Loss Functional

We first introduce the computation for the minimizer of the general penalized loss functional (1). Consider a tensor sum decomposition of the RKHS $\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J$. Let $\{\phi_v\}_{v=1}^{M}$ be a basis of $\mathcal{N}_J = \{\eta : J(\eta) = 0\}$ and R_J be the reproducing kernel in H_J. Taking the ANOVA decomposition (2) into consideration, the RKHS $\mathcal{H}_J$ can be further decomposed into $\mathcal{H}_J = \oplus_{\beta=1}^{g}\mathcal{H}_\beta$ with the reproducing kernel $R_J = \sum_{\beta=1}^{g}\theta_\beta R_\beta$, where R_β is the reproducing kernel in $\mathcal{H}_\beta$. Here the θ_β's are an extra set of smoothing parameters adjusting the contribution of the corresponding components. The minimizer of loss (1) is achieved in the tensor product reproducing kernel Hilbert space $\mathcal{H}$ with the smoothness penalty $J(\eta) = J(\eta, \eta) = \sum_{\beta=1}^{g}\theta_\beta^{-1}(\eta, \eta)_\beta$, where $(\eta, \eta)_\beta$ are inner products in $\mathcal{H}_\beta$ with reproducing kernels R_β. Without loss of generality, we define $J(\eta)$ with tensor-product cubic splines throughout the paper. To ease the notation, we give an example of a tensor product cubic spline on $[0, 1]^2$ in Appendix C, and please refer to Gu (2013) for the explicit forms of $\{R_\beta\}_{\beta=1}^{g}$ and $J(\eta)$.

According to the Kimeldorf-Wahba representer theorem (Wahba, 1990; Kimeldorf and Wahba, 1971; Wang, 2011), the minimizer of loss (1) has the following form

$$\eta(x) = \sum_{v=1}^{M} d_v\phi_v(x) + \sum_{i=1}^{n} c_i R_J(x_i, x) = \boldsymbol{\varphi}(x)^T\boldsymbol{d} + \boldsymbol{\xi}(x)^T\boldsymbol{c}, \tag{5}$$

where $\boldsymbol{d} = (d_1, \cdots, d_M)^T$, $\boldsymbol{c} = (c_1, \cdots, c_n)^T$ are unknown coefficients, $\boldsymbol{\varphi}(x) = (\phi_1, \cdots, \phi_M)^T$, $\boldsymbol{\xi}(x) = (R_J(x_i, \cdot), \cdots, R_J(x_n, \cdot))^T$ are vectors of functions. Taking advantage of the representer theorem, the infinite-dimensional optimization problem is transferred into a finite-dimensional one, thereby facilitating the estimation.

For the density estimation problem, plugging the representer of $\eta(x)$ (5) into the penalized likelihood of density estimation (3), the estimation reduces to the minimization of

$$-\frac{1}{n}\mathbf{1}^T(Q\mathbf{c}+S\mathbf{d})+\log\int\exp\left(\boldsymbol{\varphi}(x)^T\mathbf{d}+\boldsymbol{\xi}(x)^T\mathbf{c}\right)\mathrm{d}x+\frac{\lambda}{2}\mathbf{c}^TQ\mathbf{c}, \tag{6}$$

where Q is $n\times n$ with the (i,j)th entry $R_J(x_i,x_j)$ and S is $n\times M$ with the (i,v)th entry $\phi_v(x_i)$. Similarly, the minimization of the penalized likelihood functional (4) for regression can be achieved via the minimizer of:

$$\frac{1}{n}\left(\tilde{\boldsymbol{Y}}-S\mathbf{d}-Q\mathbf{c}\right)^T W\left(\tilde{\boldsymbol{Y}}-S\mathbf{d}-Q\mathbf{c}\right)+\frac{\lambda}{2}\mathbf{c}^TQ\mathbf{c}, \tag{7}$$

where W is the weight matrix, and the explicit form of $\tilde{\boldsymbol{Y}}$ and W can be found in Appendix B. Fixing smoothing parameters λ and θ, we can estimate the coefficients $\boldsymbol{d}$ and $\boldsymbol{c}$ in (6) or (7) using Cholesky decomposition (Golub and Van Loan, 2013) or Newton-Raphson method (Gu and Qiu, 1993; Gu, 2013).

To make the estimation practical, a critical aspect is selecting appropriate values for λ and θ that result in satisfactory performance, since the solution of the penalized loss functional (1) is sensitive to λ and θ (Jeon and Lin, 2006; Gu, 2013). Smoothing parameters control the trade-off between the smoothness of $\eta(x)$ and its fidelity to the data. Selecting a large smoothing parameter will lead to oversmoothing, while a small one will lead to undersmoothing.

One of the most efficient criteria for selecting the smoothing parameters is generalized cross-validation (GCV) (Gu, 1992; Gu and Wahba, 1991), which achieves the selection via cross-validation targeting the Kullback–Leibler (KL) loss. GCV consists of two main steps: (i) minimizing the KL loss with respect to λ for fixed θ; (ii) updating θ according to the updated λ. However, the computational cost of tuning the parameters, particularly λ, can be prohibitively high in high-dimensional settings. With all S smoothing parameters tunable, the above iterative algorithm takes $O(Sn^3)$ flops per iteration and needs tens of iterations to converge (Gu and Wahba, 1991). Here the number of smoothing parameters S increases as the number of multi-way interaction terms grows. In particular, $\eta(x)=\eta\left(x_{\langle 1\rangle},\ldots,x_{\langle d\rangle}\right)=\eta_\emptyset+\sum_j\eta_j\left(x_{\langle j\rangle}\right)+\sum_{j<k}\eta_{j,k}\left(x_{\langle j\rangle},x_{\langle k\rangle}\right)$, the ANOVA decomposition model (2) truncated at two-way interactions contains $S=d+\frac{3}{2}d(d-1)$ smoothing parameters. Considering the computational burden, in the case of a particularly large sample size, it is impractical to apply GCV on the full sample, i.e., train the teacher model directly, to accomplish the regression and density estimation tasks using the smoothing spline ANOVA model. To tackle this problem, we propose the knowledge cascade (KCas) method, which enables the determination of the smoothing parameters without incurring a significant computational burden during estimation.

4.2 Knowledge Cascade

In KCas, we aim to let student models learn the smoothing parameters through optimization, with a significantly lower computational burden compared to the teacher, and then transfer the smoothing parameters to teacher models. We first illustrate the definitions of the student and teacher models in our context of nonparametric functional estimation. The teacher

model is a complex model trained on the full sample, and the student model is a simple model trained on a subset of the sample with a size of b. In (5), we can see that the number of kernels, i.e., $R_J(x_i, x)$, equals the sample size n. Since the number of kernels highly affects the representation power of the model, the model complexity differs significantly for large and small sample sizes and thus distinguishes the student and teacher models. We first illustrate the simplest version of KCas in the general regression model with additive noise,

$$Y_i = \eta(x_i) + \epsilon(x_i), \tag{8}$$

where $\epsilon(\cdot)$ is the white noise process satisfying $\mathrm{E}(\epsilon(x_i)) = 0$, $\mathrm{E}(\epsilon(x_i)\epsilon(x_j)) = \sigma^2$ if $x_i = x_j$, $\mathrm{E}(\epsilon(x_i)\epsilon(x_j)) = 0$ otherwise. Define Hilbert space $\mathcal{H}^{(m)}$ by

$$\mathcal{H}^{(m)} = \Big\{\eta : \eta^{(v)} \text{absolutely continuous for } v = 0, 1, \ldots, m-1, \eta^{(m)} \in \mathcal{L}_2[0,1], \\ \eta^{(v)}(0) - \eta^{(v)}(1) = 0 \text{ for } v = 0, 1, \ldots, m-1\Big\},$$

where $\eta^{(v)} = \frac{\mathrm{d}^v \eta}{\mathrm{d}x^v}$, $\mathcal{L}_2[0,1] = \left\{f : \int_0^1 f^2 dx < \infty\right\}$ is the Hilbert space formed by all square-integrable functions, and m is a constant indicating the order of smoothness of $\mathcal{H}^{(m)}$. For smoothing splines in $\mathcal{H}^{(m)}$, the optimal smoothing parameter λ, ignoring $o(1)$ terms, is

$$\lambda = Cn^{-\frac{2m}{2mp+1}},$$

where C is an unknown constant depending on unknown function η (Wahba, 1977) and $p \in [1, 2]$ indicates different additional smoothness conditions (Wahba, 1977; Craven and Wahba, 1978; Wahba, 1985). The estimation of C is infeasible since it depends on the unknown true function η. However, KCas can infer the information of C from a well-trained subsample model (student) and apply it to the full data model (teacher). Specifically, notice that the asymptotically optimal $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}$ when sample size equals b is

$$\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b) = Cb^{-\frac{2m}{2mp+1}},$$

for the same C. We estimate the optimal $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}$ on the subsample to infer the constant C, and then employ the same C for the full data (Sun et al., 2021). That is, we have the following estimation of λ by KCas,

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b) = \lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)\left(\frac{n}{b}\right)^{-\frac{2m}{2mp+1}}. \tag{9}$$

Since the smoothing parameters are used to determine the proportion of the smoothness penalty on different terms in (2) and this proportion should be stable over different sample sizes, we directly use the optimal $\theta_{\mathrm{GCV}}^{\mathrm{sub}}(b)$ in the full sample. We then generalize the estimator (9) from the regression model with additive noise (8) to a wide range of penalized likelihood estimation problems, including the nonparametric regression in the exponential family and density estimation.

We propose to use KCas to transfer the knowledge of smoothing parameters in (3) or (4) to the teacher model. The KCas algorithm is summarized in the following Algorithm 1. In the first step, we apply uniform sampling to get a subsample X_b, and our experiments show that

uniform sampling can achieve good performance. Other more dedicated sampling methods can also be applied to improve the performance further (Wang et al., 2018; Meng et al., 2020; Daszykowski et al., 2002). The total number of operations required for each iteration is generally $\frac{4}{3}n^3 + O(n^2)$, in which the selection of the smoothing parameter takes the major burden. The GCV algorithm for the student model takes $O(Sb^3)$ flops per iteration, and thus KCas algorithm reduces the computational cost from $O(Sn^3)$ to $O(Sb^3)$. According to the justifications in Gu and Kim (2002); Kim and Gu (2004); Ma et al. (2017); Zhang et al. (2023), it is sufficient to take subsample size $O(n^{\frac{2}{9}})$ to maintain the performance of smoothing spline estimation, and we thus take a slightly larger subsample size $b = O(n^{\frac{1}{4}})$ for practical use in our algorithm. Consequently, our proposed KCas method, as detailed in Algorithm 1, substantially lowers the burden of the smoothing parameter estimation process to $O(Sn^{\frac{3}{4}})$. This underscores the pivotal role of KCas.

Algorithm 1 KCas for nonparametric function estimation.

Input: Data X, subsample size b.

1: Use uniform sampling to select a subsample X_b of size b from the full sample X of size n. Apply GCV on X_b to estimate the smoothing parameters, denoting by $\lambda^{\text{sub}}_{\text{GCV}}(b)$ and $\theta^{\text{sub}}_{\text{GCV}}(b)$.

2: Get the estimation of smoothing parameters for the full sample X using $\lambda^{\text{full}}_{\text{KCas}}(n;b) = \lambda^{\text{sub}}_{\text{GCV}}(b)(\frac{n}{b})^{-\frac{2m}{2mp+1}}$ and $\theta^{\text{full}}_{\text{KCas}}(n;b) = \theta^{\text{sub}}_{\text{GCV}}(b)$.

3: Fit smoothing splines via penalized likelihood on X with $\lambda^{\text{full}}_{\text{KCas}}(n;b)$ and $\theta^{\text{full}}_{\text{KCas}}(n;b)$ to get the function estimation $\widehat{\eta}$.

Output: Estimation $\widehat{\eta}$.

In practice, hyperparameters m and p need to be selected properly. For the univariate setting, the commonly used smooth level is $m = 2$, which means the penalty is $J(\eta,\eta) = \int_0^1 \left(\eta^{(2)}\right)^2 \, \mathrm{d}x$ on $[0,1]$. For the multivariate setting, we suggest applying the commonly used tensor product cubic spline, which has $2 - \epsilon < m < 2, \forall \epsilon > 0$ (Wahba, 1990). When $\eta^{(2)}$ is square-integrable, we have $p = 1$, and when $\eta^{(4)}$ is square-integrable, we have $p = 2$. The selection of p can vary across datasets. In this work, we take $m = 2$ and $p = 2$ empirically.

We should mention that GCV is typically used for Gaussian-type regression. For regression with responses from exponential families which results in a nonquadratic loss (4), the computation of η_λ requires a bunch of iterations even with fixed smoothing parameters and thus results in a more expensive computational cost. To address this issue, we use the well-accepted generalized approximate cross-validation (GACV) method (Xiang and Wahba, 1996; Gu and Xiang, 2001) or its variants to reduce the computational burden. Similarly, a variant of GCV (Gu et al., 2013) will be used to effectively select the smoothing parameter. To make the notation concise, variants of GCV, suitable for different problems, are collectively referred to as 'GCV'.

4.3 Theoretical Analysis

In this section, we present the theoretical properties of the smoothing parameters λ selected according to Algorithm 1. For notational simplicity, in the following, we will use λ to

represent all the smoothing parameters, not just the λ in front of $J(\eta)$. All proofs for this section are relegated to Appendix E.

Denote by $\widehat{\eta}$ the estimate through the minimization of (3) and η_0 the true function to be estimated, the asymptotic convergence rate based on the selected smoothing parameter $\lambda^{\text{full}}_{\text{KCas}}(n;b)$ is established through Theorem 1.

Theorem 1 (rate for density estimates)**.** *For the density estimation problem as in (3), denote η_0 the true log-density to be estimated. Under the regularity conditions D.1 to D.5 in Appendix D, assuming $\lambda^{\text{sub}}_{\text{GCV}}(b) \to 0$ and $b(\lambda^{\text{sub}}_{\text{GCV}}(b))^{\frac{1}{2m}} \to \infty$ as $b \to \infty$, we have the convergence rate for density estimates,*

$$(V + \lambda^{\text{full}}_{\text{KCas}}(n;b)J)(\widehat{\eta} - \eta_0) = O_p\left(n^{-1}\lambda^{\text{full}}_{\text{KCas}}(n;b)^{-\frac{1}{2m}} + \lambda^{\text{full}}_{\text{KCas}}(n;b)^p\right), \tag{10}$$

where $V(\cdot)$ is an interpretable metric such that a small $V(\widehat{\eta} - \eta_0)$ indicates a good estimate, p and m are constant parameters specified in Appendix D.

With Theorem 1, the consistency is ensured, and the convergence rate is specified for the estimation $\widehat{\eta}$ in density estimation based on KCas. Denote by $\widehat{\eta}$ the estimate through the minimization of (4), the following Theorem 2 further demonstrates the theoretical properties of KCas in the context of nonparametric regression.

Theorem 2 (rate for regression estimates)**.** *For the regression in exponential families as in (4), denote η_0 the true function to be estimated. Under the regularity conditions D.1, D.2, D.3, D.5, and D.6 in Appendix D, assuming $\lambda^{\text{sub}}_{\text{GCV}}(b) \to 0$ and $b(\lambda^{\text{sub}}_{\text{GCV}}(b))^{\frac{1}{m}} \to \infty$ as $b \to \infty$, we have the convergence rate for regression estimates,*

$$(V + \lambda^{\text{full}}_{\text{KCas}}(n;b)J)(\widehat{\eta} - \eta_0) = O_p\left(n^{-1}\lambda^{\text{full}}_{\text{KCas}}(n;b)^{-\frac{1}{2m}} + \lambda^{\text{full}}_{\text{KCas}}(n;b)^p\right), \tag{11}$$

where $V(\cdot)$, p and m are defined in the same way as in Theorem 1.

Remark. Note that the rates (10) and (11) are free of dimension d, which is because we adopt the dimensionless approach as in Chapter 9 in Gu (2013), which essentially incorporates assumptions on the smoothness to get rid of the dimension in the rates. Specifically, Condition D.3 implicitly puts an assumption on the smoothness of η, which makes the error rates (10) and (11) to be free of dimension d. The condition becomes more stringent as dimension d becomes larger. The results can be adapted to the case with dimension using arguments in Lin (2000).

4.4 Knowledge Cascade in Kernel Density Estimation

We next illustrate the KCas framework in the classical univariate kernel density estimation (KDE) problem. Let $x_1, \ldots, x_n$ be i.i.d. observations from an unknown density f on $\mathbb{R}$. For a symmetric kernel K and a bandwidth $h > 0$, the Gaussian kernel density estimator is

$$\widehat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right),$$

where we take K to be the standard normal density throughout (Silverman, 2018). For twice differentiable f, the classical asymptotic mean integrated squared error (AMISE) theory yields an optimal bandwidth of the form

$$h_{\mathrm{AMISE}} = \left\{ \frac{R(K)}{m_2(K)^2 \, R\big(f''\big)} \right\}^{1/5} n^{-1/5} = C \, n^{-1/5},$$

where $R(g) = \int g^2(x)\,dx$, $m_2(K) = \int x^2 K(x)\,dx$, and C is an unknown constant depending on f and K (Silverman, 2018; Jones et al., 1996).

In KCas, we treat a KDE fitted on a small subsample as the student model and the full-sample KDE as the teacher model. The key idea is to estimate the constant C from a subsample of size b and then transfer it to the full data via the AMISE scaling. Specifically, the AMISE-optimal bandwidth for the subsample satisfies

$$h^{\mathrm{sub}}_{\mathrm{AMISE}}(b) = C \, b^{-1/5},$$

so that $C = h^{\mathrm{sub}}_{\mathrm{AMISE}}(b)\, b^{1/5}$. Plugging this expression into the full-sample formula gives the KCas bandwidth

$$h^{\mathrm{full}}_{\mathrm{KCas}}(n; b) = h^{\mathrm{sub}}_{\mathrm{AMISE}}(b) \Big(\frac{n}{b} \Big)^{-1/5}. \tag{12}$$

In our implementation, $h^{\mathrm{sub}}_{\mathrm{AMISE}}(b)$ is not computed by oracle access to f but by a data-driven selector applied to the subsample; the KCas bandwidth then extrapolates this choice to the full data. This is a direct KDE analogue of the smoothing-parameter scaling used in smoothing splines.

The same knowledge cascade principle extends naturally beyond KDE to a broad class of kernel-based methods. In many kernel procedures, such as kernel regression (Wand and Jones, 1994), local polynomial smoothing (Fan, 2018), and kernel-based covariance estimation (Ferraty and Vieu, 2006), the optimal tuning parameters follow explicit sample-size dependent scaling laws derived from asymptotic theories. KCas exploits this structure by learning the problem-dependent constants from a small subsample and transferring them to the full sample through the corresponding scaling relations, thereby reducing computational cost while preserving statistical efficiency. This perspective suggests KCas as a general strategy for tuning-parameter transfer in kernel methods with known asymptotic rates.

4.5 Knowledge Cascade in Deep Learning

Although KCas is developed in the context of nonparametric multivariate functional estimation, the underlying principle extends naturally to settings in which computational constraints make full hyperparameter tuning difficult. In many deep learning applications, the cost of identifying stable and high-performing hyperparameter configurations, particularly the learning rate schedule, dominates the total training budget. Motivated by this challenge, we investigate a simple extension of KCas that uses a compact student network to guide the hyperparameter choices for a larger teacher network.

In this setting, the student model is trained on a reduced architecture and its optimal hyperparameters are obtained through standard tuning procedures. KCas then scales these hyperparameters to the teacher model using a batch-size dependent rule. This approach

acknowledges empirical findings that learning rate schedules tuned on smaller networks often transfer to larger networks when batch sizes are adjusted appropriately. Consequently, KCas reduces hyperparameter search cost by allowing tuning to be performed once on the student model and then extrapolated to the teacher.

Mathematically, let α_{student} denote the optimal learning rate for the student model, and let B_{student} and B_{teacher} be the respective batch sizes. The KCas learning rate for the teacher model is defined as

$$\alpha_{\text{teacher}} = \alpha_{\text{student}} \cdot g\left(\frac{B_{\text{teacher}}}{B_{\text{student}}}\right),$$

where $g(\cdot)$ is a monotone scaling function. We consider two commonly used choices. The first is the linear rule $g(r) = r$, inspired by the large-batch scaling observations in Goyal et al. (2017). Since teacher architectures in our experiments are substantially more complex and often benefit from more conservative step sizes, we also examine the square-root rule $g(r) = \sqrt{r}$. These two options illustrate how KCas can adapt scaling behavior to model complexity.

Although this extension does not rely on the RKHS-based theory that supports KCas in nonparametric functional estimation, it follows the same guiding philosophy: use a computationally efficient student model to extract stable hyperparameter information and propagate it upward to a more complex teacher model. As demonstrated in Section 6.2, this strategy provides a practical reduction in tuning cost while maintaining competitive predictive performance for modern deep learning architectures.

5. Simulation Study

We conduct simulation studies to evaluate the proposed KCas framework from two perspectives. First, we examine its performance in the main setting of this paper: smoothing spline ANOVA models for density estimation and nonparametric regression. These experiments assess whether smoothing parameters learned from a subsample-based student model can be effectively transferred to the full-sample teacher model while reducing computational cost. Second, we investigate KCas in KDE to illustrate the same student-to-teacher transfer principle in a different nonparametric setting where bandwidths follow classical asymptotic scaling laws.

5.1 Simulation 1: Density Estimation with Smoothing Splines

We first evaluate the proposed KCas method for smoothing splines on synthetic density estimation problems. We consider the following two data-generating scenarios.
Scenario 1: A d-dimensional Gaussian mixture model, $\frac{1}{d}\sum_{i=1}^{d} \text{MVN}(e_i, I_d)$, where e_i is the vector with the ith entry being 1 and others being 0. We consider $d = 3, 6$.
Scenario 2: A d-dimensional density is constructed by independently combining a 5-dimensional Gaussian with mean zero and variance $0.5(\mathbf{1}\mathbf{1}^T) + 1.5I_d$, and the remaining $d-5$ variables are i.i.d. from $\text{Unif}(0, 1)$. We consider $d = 15, 20$.

We evaluate the methods using the relative Kullback–Leibler (KL) divergence with respect to the benchmark method, defined as

$$\mathrm{RelKL}\left(\widehat{p}, p^*, p\right) = \frac{D_{\mathrm{KL}}(\widehat{p}\|p)}{D_{\mathrm{KL}}(p^*\|p)},$$

where p is the true distribution, $\widehat{p}$ is the estimate from the method being evaluated, and p^* is the estimate from the benchmark method. Here $D_{\mathrm{KL}}(q\|p)$ denotes the KL divergence of q relative to p. A smaller relative KL divergence indicates better performance. In each replication, full-sample GCV is used as the benchmark, and the relative KL divergence is computed with respect to the corresponding GCV estimate from that replication. For visualization, we report log-transformed RelKL.

For the smoothing spline experiments, including Simulation 1 here and following Simulation 2, we compare KCas with the following baseline methods:

- GCV (Gu et al., 2013) on the full sample: Full-sample GCV uses all available observations for smoothing-parameter selection and is therefore used as the benchmark. All other methods are evaluated using relative performance measures with respect to full-sample GCV.

- GCV on subsample (SUB): GCV is performed on a randomly selected subsample to reduce computational cost.

- GCV in generalized additive models (GAM) (Wood, 2004): GCV is performed on the full sample using a simplified additive model with only main effects.

- SKIP method (Gu, 2014): SKIP accelerates computation by selecting an appropriate starting point and bypassing subsequent iterations. Since it often fails to converge in relatively high-dimensional density estimation settings, we include it only for the nonparametric regression problem.

- Order-based method (ORD) (Hall, 1990): ORD directly sets the smoothing parameter to $\lambda = n^{-\frac{2m}{2mp+1}}$, where n is the full sample size.

- Kernel density estimation (KDE): For density estimation, we further include the KDE method of Nagler and Czado (2016) as a comparison for high-dimensional data.

For KCas, we use uniform sampling to select a subsample of size $b = 50n^{\frac{1}{4}}$. The full sample sizes are $5,000$, $10,000$, and $20,000$. Following the discussion in Section 4.2, we set $m = 2$ and $p = 2$ in practice. All results are based on 30 replications.

Figure 1 illustrates the log-transformed RelKL of KCas and the competing methods. KCas performs favorably across the considered settings, with log-RelKL values close to or below zero in many cases. Negative values indicate that KCas outperforms the full-sample GCV benchmark in that replication. This suggests that smoothing parameters learned from the student model can effectively guide the teacher model and, in some settings, may also reduce the effect of overly adaptive full-sample tuning.

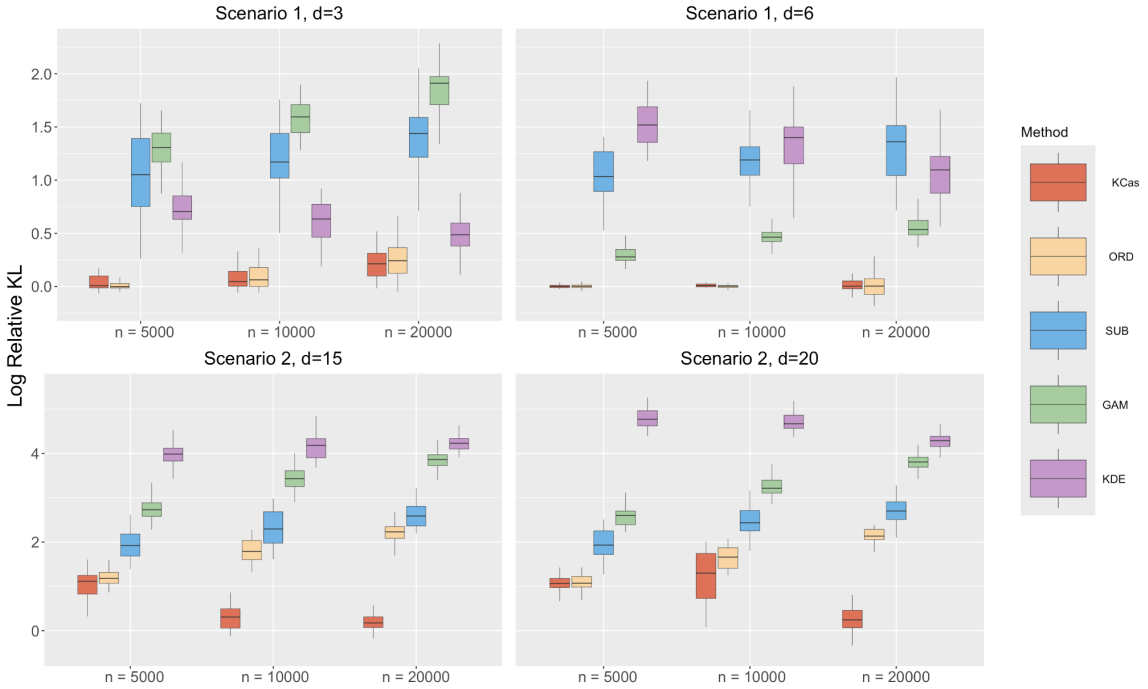

Figure 1: Performance comparisons of different methods in density estimation problems using the log-transformed relative KL divergence. The lower relative KL divergence indicates better performance. Two scenarios of data generation processes are provided, each including several different settings of dimension d and sample size n.

5.2 Simulation 2: Nonparametric Regression with Smoothing Splines

For the nonparametric regression problem, we consider the model

$$y_i \sim \text{Ber}\left(\frac{\exp\left(\eta\left(x_i\right)\right)}{1+\exp\left(\eta\left(x_i\right)\right)}\right),$$

where $\text{Ber}(p)$ denotes the Bernoulli distribution with probability p, $x_i = \left(x_{i\langle 1\rangle}, \ldots, x_{i\langle d\rangle}\right)^{\mathrm{T}}$ is the d-dimensional predictor for the ith observation, with each entry independently drawn from $\text{Unif}(0,1)$. η is the nonparametric function determining the success probability in the Bernoulli trial, and $y_i \in \{0,1\}$ is the response variable for the ith observation.

We evaluate the methods by the relative MSE, defined by

$$\text{RelMSE}\left(\widehat{\eta}, \eta^*, \eta\right) = \frac{\sum_{i=1}^{n}\left\{\widehat{\eta}\left(x_i\right) - \eta\left(x_i\right)\right\}^2}{\sum_{i=1}^{n}\left\{\eta^*\left(x_i\right) - \eta\left(x_i\right)\right\}^2},$$

where η is the true function, $\widehat{\eta}$ is the estimation by the method being evaluated, and η^* is the benchmark method. We compare KCas with the same baseline methods as in Simulation 1.

We consider two different scenarios with different dimensions. We report log-transformed RelMSE for plotting clearness, and each RelMSE is computed based on a full GCV baseline

for that particular replication.

Scenario 1:

$$\eta_{m1}(x) = \sum_{i=1}^{3} g_1\left(x_{\langle i\rangle}\right) + g_2\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}\right) + g_2\left(x_{\langle 1\rangle}, x_{\langle 3\rangle}\right) + g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right).$$

Scenario 2:

$$\eta_{m2}(x) = \sum_{i=1}^{3} i g_1\left(x_{\langle i\rangle}\right) + \sum_{i=4}^{6} i g_5\left(x_{\langle i\rangle}\right) + \sum_{i=7}^{9} g_4\left(x_{\langle i\rangle}\right) + \sum_{i=1}^{3}\sum_{j>i}^{4} 3i g_2\left(x_{\langle i\rangle}, x_{\langle j\rangle}\right) + 6 g_2\left(x_{\langle 5\rangle}, x_{\langle 6\rangle}\right) + 8 g_6\left(x_{\langle 7\rangle}, x_{\langle 8\rangle}\right) + 10 g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right).$$

The explicit forms of the functions g_i are provided in Appendix F.1. Scenario 1 is a well-established setting for nonparametric multivariate functional estimation in RKHS (Jeon and Lin, 2006; Gu and Wahba, 1991; Sun et al., 2021; Gu and Wang, 2003). We consider two cases, with $d = 3$ and $d = 6$ predictors. When $d = 3$, all predictors contribute to η_{m1} and hence to the response y. When $d = 6$, the last three predictors are irrelevant to η_{m1}, representing a setting with redundant covariates. Scenario 2 is a more complex high-dimensional setting, where we consider $d = 15$ and $d = 20$.

	GCV	KCas	GAM	SUB	ORD	SKIP
$n = 5000$	42.2	6.1	1.3	5.5	16.2	2.1
$n = 10000$	72.9	9.4	2.5	7.0	23.1	3.3
$n = 20000$	102.5	8.1	3.2	5.0	30.8	3.5

Table 1: Comparison of median computational time (min) under the most difficult simulation setting (simulation 2, scenario 2, $d = 20$).

Figure 2 displays the log-transformed RelMSE relative to full-sample GCV. In Scenario 1 with $d = 3$, KCas and SKIP show comparable performance and outperform the other methods. Their median log-RelMSE values are close to, and sometimes below, zero, indicating performance comparable to or better than full-sample GCV. When $d = 6$, KCas remains close to full-sample GCV, whereas SKIP performs substantially worse, with median log-RelMSE values greater than 1. This is likely because SKIP uses the starting values introduced by Gu (2014) as the final estimate; when η becomes more complex, such starting values may be far from the optimum, leading to inaccurate estimation. In Scenario 2, similar patterns are observed for both $d = 15$ and $d = 20$, with KCas achieving the best overall performance.

To empirically compare computational efficiency, we report the running time under the most challenging simulation setting, Scenario 2 with $d = 20$, in Table 1. KCas substantially reduces computation time compared with full-sample GCV while maintaining comparable estimation accuracy. It is important to note that while methods such as GAM, SUB, and SKIP require less computation time than KCas, they do not achieve the same level of accuracy.

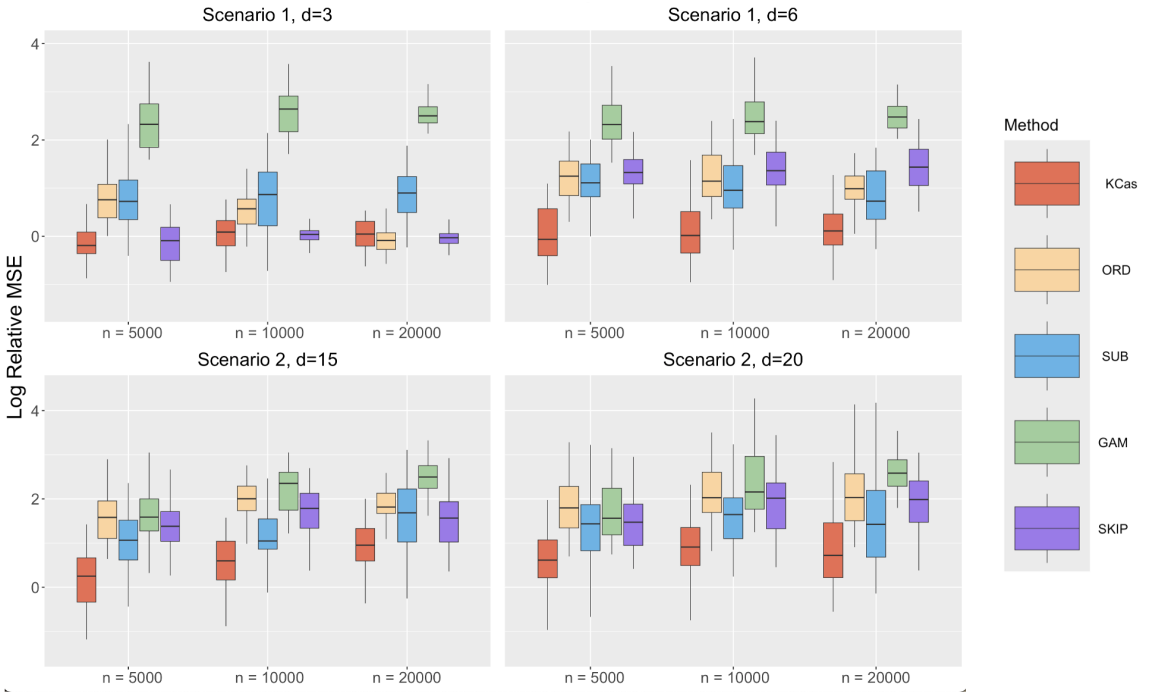

Figure 2: Performance comparisons of different methods in nonparametric regression problems using log RelMSE. The lower log RelMSE indicates better performance. Two scenarios of data generation processes are provided, each including several different settings of dimension d and sample size n.

5.3 Simulation 3: Kernel Density Estimation

We next conduct an additional simulation to evaluate KCas in classical kernel density estimation. This experiment is separate from the smoothing spline simulations above and uses bandwidth-selection methods as baselines. The goal is to examine whether information learned from a small subsample can be transferred to the full-sample KDE estimator through the classical $n^{-1/5}$ bandwidth scaling rule.

We consider six univariate benchmark densities widely used in KDE studies (e.g., Silverman, 2018; Jones et al., 1996): (1) standard normal density $\mathcal{N}(0,1)$; (2) symmetric bimodal Gaussian mixture $0.6\mathcal{N}(-1,1^2)+0.4\mathcal{N}(2,0.5^2)$; (3) lognormal density Lognormal$(0,1)$; (4) trimodal Gaussian mixture $0.4\mathcal{N}(-2,0.5^2)+0.3\mathcal{N}(0,0.6^2)+0.3\mathcal{N}(3,0.8^2)$; (5) gamma density Gamma$(2,1.5)$; and (6) skewed bimodal Gaussian mixture $0.7\mathcal{N}(-1,1^2)+0.3\mathcal{N}(3,0.5^2)$. For each density, we generate samples of size $n \in \{200,500,1000,2000,5000\}$. For a given density and n, we draw 50 independent replications and, for each replication, compute several bandwidth selectors and their associated KDEs.

We compare the following methods:

(i) **Oracle AMISE**: the theoretical AMISE-optimal bandwidth h_{AMISE} obtained by numerically evaluating $R(f'')$ from the known f. This serves as a lower bound on the achievable mean integrated squared error (MISE).

(ii) **Improved Sheather-Jones (ISJ)**: the diffusion-based plugin selector of Botev et al. (2010), which refines the original Sheather–Jones solve-the-equation method (Sheather and Jones, 1991) and is known to perform well across a wide range of densities.

(iii) **Least-squares cross-validation (LSCV)**: the classical least-squares cross-validation bandwidth (Rudemo, 1982; Bowman, 1984), which minimizes an unbiased estimate of the integrated squared error.

(iv) **Silverman's rule of thumb**: the normal-reference bandwidth $h_{\mathrm{S}} = 0.9 \min\{\widehat{\sigma}, \mathrm{IQR}/1.34\}$ $n^{-1/5}$ (Silverman, 2018), where $\widehat{\sigma}$ is the sample standard deviation and IQR is the sample interquartile range, defined as the difference between the third and first sample quartiles, $\mathrm{IQR} = Q_3 - Q_1$.

(v) **KCas-Silverman**: a KCas estimator based on the Silverman rule computed on a uniform subsample of size $b = 200$, i.e. $h_{\mathrm{KCas},S}(n; b)$ obtained by plugging $h^{\mathrm{sub}}_{\mathrm{AMISE}}(b) = h^{\mathrm{sub}}_{\mathrm{S}}(b)$ into (12).

(vi) **KCas-ISJ**: a KCas estimator based on the ISJ selector computed on a subsample of size $b = 200$, i.e. $h_{\mathrm{KCas,ISJ}}(n; b)$ from (12) with $h^{\mathrm{sub}}_{\mathrm{AMISE}}(b)$ taken as the subsample ISJ bandwidth.

(vii) **KCas-CV**: a KCas estimator based on least-squares cross-validation (LSCV) computed on a subsample of size $b = 200$, i.e. $h_{\mathrm{KCas,CV}}(n; b)$ obtained by plugging the subsample LSCV bandwidth $h^{\mathrm{sub}}_{\mathrm{CV}}(b)$ into (12).

In each method, we fit a Gaussian KDE on an equally spaced grid of 4096 points covering the effective support of each density. The integrated squared error (ISE) for a replication is approximated via the trapezoidal rule,

$$\mathrm{ISE}(\widehat{f}_h) \approx \sum_k \left\{\widehat{f}_h(x_k) - f(x_k)\right\}^2 \Delta x,$$

and we summarize performance using the median ISE and interquartile range across the 50 replications. To facilitate comparison, we also report the ratio of the median ISE to that of the oracle AMISE bandwidth for each density and n, as well as log–log slopes of median bandwidth versus sample size to check the $n^{-1/5}$ scaling.

Figure 3 displays the distribution of ISE values across replications for all selectors, stratified by density and sample size. As expected, the oracle AMISE bandwidth achieves the smallest median ISE in every setting and provides a lower bound for the other procedures. Among implementable full-sample selectors, the diffusion-based ISJ bandwidth is consistently the strongest competitor, with median ISEs very close to the oracle across all n and for both simple (normal, gamma) and more complex (bimodal, trimodal, skewed) densities, in line with previous empirical studies (Sheather and Jones, 1991; Jones et al., 1996; Botev et al., 2010). Silverman's rule-of-thumb performs competitively for the unimodal, approximately Gaussian

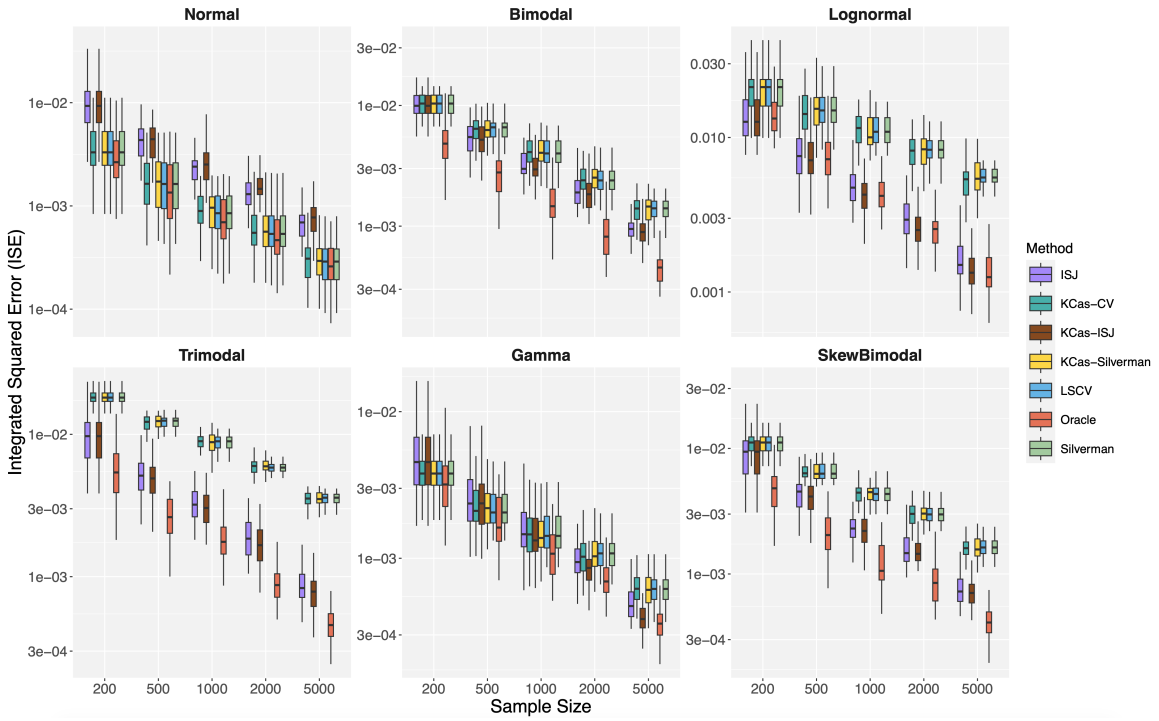

Figure 3: KCas for Kernel Density Estimation. Integrated squared error of KDE bandwidth selection methods across six benchmark densities and five sample sizes. Lower values indicate better performance.

densities, but becomes noticeably suboptimal for heavier-tailed and highly multimodal shapes, where its normal-reference assumption leads to oversmoothing. The LSCV selector exhibits the largest variability, especially for smaller sample sizes, with wider ISE distributions and occasional outlying bandwidths, reflecting its well-known finite-sample instability (Rudemo, 1982; Bowman, 1984; Jones et al., 1996).

The KCas variants largely inherit the strengths and weaknesses of their corresponding base selectors while operating on only a small subsample. For the normal and gamma densities, KCas-Silverman attains median ISE ratios on the order of 1.05–1.20 relative to the oracle across the range of n, indicating only a modest loss in efficiency despite learning bandwidths from $b = 200$ observations. For multimodal and skewed densities, KCas-ISJ is more adaptive and typically dominates KCas-Silverman, with median ISE curves that nearly overlap those of the full-sample ISJ selector. Across all densities, the variability of KCas-ISJ remains close to that of ISJ itself, suggesting that the additional randomness induced by subsampling does not substantially degrade performance. KCas-CV performs reasonably well in smooth unimodal settings, particularly for the normal density, but is generally less competitive for heavier-tailed and multimodal densities and exhibits larger variability, reflecting the weaker finite-sample stability of cross-validation based selectors. Overall, KCas-Silverman and KCas-ISJ consistently outperform KCas-CV, benefiting from more structured statistical foundations and stronger theoretical guarantees.

From the computational perspective, the cost of KCas is dominated by fitting the student model on $b = 200$ points, rendering the bandwidth selection cost essentially independent of n once b is fixed. Taken together, these results demonstrate that, in the KDE setting, KCas provides a practical mechanism for transferring the adaptivity of sophisticated selectors such as ISJ to large-sample regimes, while maintaining optimal scaling and substantially reducing the computational burden of hyperparameter tuning.

Simulation results across various settings and methods show that KCas can effectively transfer knowledge of smoothing parameters from the student model on the subsample to the teacher model on the full sample, consistent with our theoretical analysis.

6. Real Data Analysis

We apply KCas to real datasets in two settings: smoothing-spline-based nonparametric estimation and deep learning hyperparameter transfer. Specifically, we consider density estimation and nonparametric regression on benchmark datasets, as well as image classification on CIFAR-10.

6.1 Nonparametric Estimation

We first evaluate KCas on smoothing-spline-based nonparametric estimation tasks. The density estimation study uses four benchmark datasets, and the nonparametric regression study uses five benchmark datasets. Details of the datasets are provided in Appendix G. Each dataset is randomly split into 80% training and 20% testing sets. Features are scaled using a min–max transformation fitted on the training set and then applied to the testing set. All relative metrics in this subsection are computed with respect to full-sample GCV.

Density estimation. We compare KCas with GAM, SUB, ORD, and KDE, in density estimation. Since there is no ground truth for the density function, we evaluate the performance based on the average log-likelihood on the test set, as suggested by previous studies (Papamakarios et al., 2017; Gao et al., 2022). We consider the ANOVA decomposition of η including all main effects and all two-way interactions. The model terms are selected using the model diagnosis suggested by Gu (2004). Our results, presented in Table 2, demonstrate that KCas outperforms all four benchmark methods in terms of log-likelihood across all data sets. Notably, on the ESC and MFCC datasets, KCas even outperforms GCV on the full sample, both in terms of log-likelihood and computational time.

Nonparametric regression. We compare KCas with GAM, SUB, ORD, and SKIP, in nonparametric regression for exponential families. Since we do not know the underlying probability of each data point, as suggested by Wang et al. (2018), we calculate RelMSE over GCV. The main effect and interaction terms are selected using the model diagnosis suggested by Gu (2004). Table 3 shows that KCas outperforms all four benchmark methods in terms of MSE. Although KCas is not the fastest among the methods, it is faster than the full sample estimator in all studies, while obtaining the best performance among the comparison methods.

	Relative log-likelihood					Relative computation time				
Method	KCas	GAM	SUB	ORD	KDE	KCas	GAM	SUB	ORD	KDE
CD14	**0.9998**	0.8095	0.7642	0.9990	0.8342	0.86	0.84	0.26	0.48	7.40
AReM	**0.9995**	0.9657	0.9823	0.9978	0.9568	0.73	0.82	0.10	0.24	1.02
ESC	**1.0475**	0.5514	1.0369	1.0191	0.2503	0.53	0.87	0.42	0.44	0.69
MFCC	**1.0054**	0.9572	0.9908	0.9988	0.2528	0.24	0.20	0.19	0.19	1.73

Table 2: Relative log-likelihood and relative computational time for different methods in density estimation with real data. GCV is taken as the benchmark method. Higher relative log-likelihood indicates better performance. The highest relative log-likelihood values for each dataset are marked in bold.

	Relative MSE					Relative computation time				
Method	KCas	GAM	SUB	ORD	SKIP	KCas	GAM	SUB	ORD	SKIP
SUSY	**0.7963**	0.8333	1.2185	0.9385	0.8252	0.15	0.09	0.11	0.01	0.09
WFRN	**1.0434**	1.0535	1.3604	1.1071	1.0827	0.41	0.02	0.21	0.01	0.03
OCUP	**0.9999**	4.1504	2.0734	1.0055	1.0010	0.10	0.16	0.03	0.07	0.13
SHILL	**0.9825**	0.9839	1.0218	1.0217	1.0162	0.27	0.03	0.26	0.01	0.01
CIFAR-10	**1.0040**	1.0710	1.0513	1.0264	1.0171	0.53	0.06	0.21	0.01	0.03

Table 3: Relative MSE and relative computational time for different methods in nonparametric regression with real data. GCV is taken as the benchmark method. Lower relative MSE indicates better performance. The lowest relative MSE values for each dataset are marked in bold.

6.2 Image Classification with Deep Learning

We next apply KCas to deep learning hyperparameter transfer on the CIFAR-10 dataset (Krizhevsky, 2009). CIFAR-10 contains $50,000$ training images and $10,000$ test images from ten evenly represented object classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.

Model Structure. We consider two student–teacher pairs. MobileNetV2 with 2.2M parameters (Sandler et al., 2018) and ResNet18 with 11.7M parameters (He et al., 2016) serve as student models, while ResNet50 with 25.6M parameters serves as the teacher model. All architectures are adapted to the CIFAR-10 resolution by removing max-pooling and replacing the initial layers with a 3×3 convolution. Standard dataset normalization is applied.

	Student 1	Student 2	Teacher
Model Structure	MobileNetV2	ResNet18	ResNet50
Number of Parameters	2.2M	11.7M	25.6M

Table 4: Model structure and model size.

KCas Hyperparameter Search. To obtain student configurations for transfer, each student model is tuned by grid search over 12 hyperparameter combinations: batch sizes $128, 256, 512$, learning rates $0.05, 0.1$, and weight-decay values $5e{-}4, 1e{-}3$. Students are trained for 200 epochs with SGD using cosine decay and a 5% warmup. The best configuration is selected using a 10% holdout validation set. KCas then scales the selected learning rate to the teacher model.

Baselines. We compare KCas with three baselines. **(1) Student model**: the accuracy of the tuned student model. **(2) Cookbook**: hand-selected hyperparameters commonly recommended for the teacher architecture. **(3) Retune**: a full grid search for the teacher over 36 combinations: batch sizes $256, 512, 1024$, learning rates $0.02, 0.05, 0.1, 0.2$, and weight-decay values $1e{-}4, 5e{-}4, 1e{-}3$. This baseline represents the standard but computationally expensive approach to teacher hyperparameter tuning.

Results. For each method, we record test accuracy and training time, averaged over three repetitions.

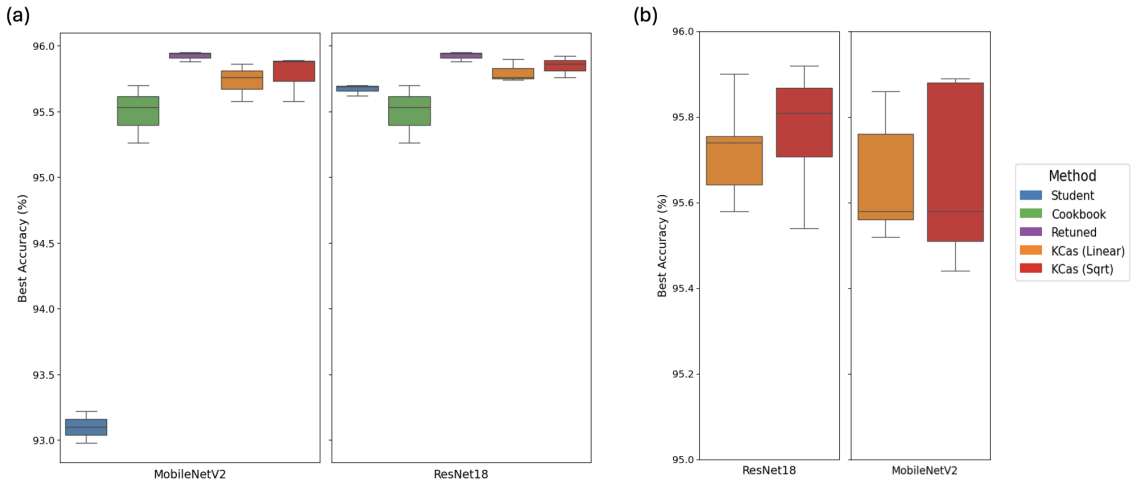

Figure 4: KCas for deep learning. We evaluate KCas for transferring learning-rate schedules from a compact student network to a larger teacher network. Two student architectures, MobileNetV2 and ResNet18, are used to guide the training of teacher models. (a) Accuracy comparisons among baselines. (b) Direct comparison of the two KCas scaling rules.

Figure 4 shows that KCas can effectively transfer learning-rate information from smaller networks to a larger teacher network, achieving teacher performance competitive with full hyperparameter re-tuning. Panel (a) compares KCas with the baselines. Both KCas variants, using linear and square-root scaling, substantially improve over the student models and reach accuracy levels close to those of fully re-tuned teachers for both MobileNetV2 and ResNet18. The cookbook rule provides moderate gains but consistently falls short of the proposed KCas approaches. An additional pattern in panel (a) is that teacher performance is higher when ResNet18 is used as the student rather than MobileNetV2. This likely reflects the stronger standalone performance of ResNet18, which allows it to provide more informative

hyperparameters for transfer. The results suggest that the student model should not be too weak in practice. Although MobileNetV2-based transfer still outperforms the cookbook baseline, a stronger student yields clearer benefits for the teacher.

Panel (b) directly compares the two KCas scaling rules across all batch-size choices, rather than only the best setting. Linear and square-root scaling achieve similar accuracy overall, while the square-root rule shows slightly greater stability across repetitions. Together with panel (a), these results suggest that square-root scaling tends to perform better in this experiment.

Time (h)	Student	KCas	Cookbook	Retune
MobileNetV2	3.5	6.9	3.4	121.1
ResNet18	18.5	21.8	3.4	121.1

Table 5: Comparison of median computational time (hours) for the image classification task.

Table 5 reports the computational time. Compared with full re-tuning of the teacher model, which requires more than 120 hours, KCas uses only a small fraction of the computational budget. The savings are more pronounced with a smaller student model, since the total cost of KCas depends on the student-training time. In practical scenarios where a trained student model is already available, the additional cost of KCas reduces to training a single teacher model, making it comparable to the cookbook baseline. Although KCas is slightly more expensive than the cookbook approach when the student must be trained from scratch, it consistently achieves better performance. These results illustrate that KCas offers a favorable balance between efficiency and predictive performance for deep learning hyperparameter transfer.

7. Discussion

In this paper, we propose Knowledge Cascade (KCas), a student-to-teacher knowledge transfer framework developed primarily for nonparametric functional estimation in RKHS. In this setting, a student model fitted on a small subsample guides smoothing-parameter selection for the full-sample teacher model, reducing tuning cost while retaining strong statistical performance. Our simulation and real-data results show that KCas compares favorably with several computationally efficient alternatives and can even outperform full-sample GCV in some settings. Beyond the main RKHS setting, we further illustrate the same student-to-teacher transfer principle through kernel density estimation and deep learning hyperparameter transfer. These examples suggest that KCas is not limited to smoothing spline models, but can provide a broader strategy for scalable model development whenever useful information learned from a smaller model can be transferred to a larger one. More broadly, KCas offers a new perspective on knowledge distillation by showing that small models can sometimes guide, rather than merely approximate, larger models. It remains an open question whether such knowledge cascades can be constructed without the support of asymptotic or empirical scaling rules.

When and why to use KCas. KCas is most useful when a smaller student model can learn information that remains informative for developing a larger teacher model. The key requirement is not that the student approximates the teacher perfectly, but that it captures transferable information that can reduce the cost of training or tuning the teacher. Such information may include smoothing parameters, bandwidths, training configurations, or other structural features that are expensive to learn directly at full scale. When this information follows a suitable asymptotic or empirical scaling relationship, KCas provides a way to extract it from a low-cost student model and propagate it to the teacher.

KCas is not intended to replace the teacher with a smaller approximation. Instead, the student serves as a computationally efficient guide for building the teacher. In this paper, we instantiate this idea mainly through hyperparameter transfer. For example, in smoothing spline ANOVA models, KCas learns smoothing parameters from a subsample-based student model and transfers them to the full-sample teacher model through an asymptotic scaling rule. Thus, KCas avoids repeated full-scale tuning while still fitting the final model on all observations. More broadly, the same principle suggests a general strategy for reducing the cost of teacher-model development whenever useful information can be learned cheaply from a smaller model and reliably transferred to a larger one.

KCas can outperform the full sample estimator. In multiple scenarios in our simulation and real data analysis, we have observed an interesting phenomenon that KCas, based on a random subsample, can outperform the full sample GCV estimator. One potential explanation is that the statistical theory underlying KCas helps make hyperparameter transfer more efficient. Although this finding appears to conflict with the traditional intuition, recent studies (Nakkiran et al., 2021; Guo et al., 2022; Yang et al., 2022; Sorscher et al., 2022; Gadre et al., 2023), have observed similar phenomena that models trained on subsamples sometimes can achieve better performance in the context of deep models. Further, a related paper (Kolossov et al., 2024) observed such a phenomenon for simpler settings under empirical risk minimization, and they aim to develop some theoretical justification for it. Although they work under different model settings, it is an interesting future direction to explore this phenomenon theoretically and empirically in nonparametric estimation or deep learning settings.

Connection to broader statistical methodology. The KCas principle may also be relevant to other statistical methods whose effective complexity changes with sample size. Examples include basis-expansion methods, wavelet estimators, series regression, and other multiresolution procedures. In these settings, quantities such as the number of basis terms, resolution levels, thresholding parameters, or regularization strengths often need to be adjusted as the sample size increases. This makes them natural candidates for KCas-type extensions, since a student model fitted at a smaller scale may help estimate problem-specific quantities that guide the corresponding full-scale estimator. Establishing suitable scaling rules and theoretical guarantees for these broader classes of methods remains an important direction for future work.

Code availability. The source code of KCas is available at: `https://github.com/LuyangFang/KCas`.

Acknowledgments

This work was partially supported by the U.S. National Science Foundation under grants DMS-1903226, DMS-1925066, DMS-2124493, DMS-2311297, DMS-2319279, DMS-2318809, and by the U.S. National Institutes of Health under grant R01GM152814. The authors declare no competing interests.

Appendix A. Existence of the Minimizer

The following theorem guarantees the existence of the minimizer of (1) in RKHS.

Theorem A.1 (Existence of the minimizer, Wahba (1990)). *Suppose $L(\eta)$ is a continuous and convex functional in a Hilbert space $\mathcal{H}$ and $J(\eta)$ is a square (semi) norm in $\mathcal{H}$ with a null space $\mathcal{N}_J$, of finite dimension. If $L(\eta)$ has a unique minimizer in $\mathcal{N}_J$, then $L(\eta) + \frac{\lambda}{2}J(\eta)$ has a minimizer in $\mathcal{H}$.*

When $L(\eta)$ is the negative log-likelihood function, it is usually convex in η. The quadratic functional $J(\eta)$ is also convex (Gu and Qiu, 1993). A minimizer of $L(\eta)$ is unique in $\mathcal{N}_J$ if the convexity is strict in it, which is often the case. Thus, the solution for Equation (1) exists in most cases.

Appendix B. Minimizer of the Penalized Loss Functional

Consider exponential family distributions with densities of the form

$$f(y \mid x) = \exp\{\frac{y\vartheta(x) - b(\vartheta(x))}{a(\phi)} + c(y, \phi)\}, \tag{B.1}$$

where $a > 0, b$, and c are known functions, $\vartheta(x)$ is the canonical parameter dependent on a covariate x, and ϕ is either known or considered as a nuisance parameter that is independent of x. Fixing the smoothing parameters, the penalized likelihood functional (4) is strictly convex in η. Thus, given the current $\tilde{\eta}$, the Newton iteration can be used to update $\tilde{\eta}$ by the minimizer of the penalized weighted least square functional

$$\frac{1}{n}\left(\tilde{\boldsymbol{Y}} - S\mathbf{d} - Q\mathbf{c}\right)^T W \left(\tilde{\boldsymbol{Y}} - S\mathbf{d} - Q\mathbf{c}\right) + \frac{\lambda}{2}\mathbf{c}^T Q\mathbf{c}, \tag{B.2}$$

where $\tilde{Y}_i = \tilde{\eta}(x_i) - \frac{\tilde{u}_i}{\tilde{w}_i}$, $\tilde{w}_i = \ddot{b}(\tilde{\eta}(x_i))$, and $\tilde{u}_i = -Y_i + \dot{b}(\tilde{\eta}(x_i))$. Here $\tilde{w}_i$ is the ith diagonal element of the matrix W, $\tilde{Y}_i$ is the ith element of $\tilde{\boldsymbol{Y}}$, $\dot{b}$ and $\ddot{b}$ are the first and second derivatives of the function b.

Appendix C. Example of the Tensor Product Cubic Spline on $[0, 1]^2$

For the univariate η on $\mathcal{X}$, the most popular choice of the smoothness penalty $J(\eta)$ is

$$J(\eta, \eta) = \int_{\mathcal{X}} \left(\eta^{(m)}\right)^2 \, \mathrm{d}x, \tag{C.1}$$

where $\eta^{(m)} = \mathrm{d}^m\eta/\mathrm{d}x^m$. A cubic estimator of the minimizer of (1) is obtained by setting $m = 2$. This idea can be extended to multivariate settings. Consider the ANOVA decomposition (2). $\mathcal{H}$ can be decomposed into the space of constants, the spaces of main effects, and the corresponding spaces of interaction terms lying in the tensor product space of the interacting main-effect spaces. For the two-dimensional problem, one has the following space

decomposition in each variable (Gu (2013), section 2.3):

$$\begin{aligned}\left\{\eta : \eta^{(2)} \in \mathcal{L}_2[0,1]\right\} =& \{\eta : \eta \propto 1\} \oplus \{\eta : \eta \propto k_1\} \\ & \oplus \left\{\eta : \int_0^1 \eta \, \mathrm{d}x = \int_0^1 \eta^{(1)} \mathrm{d}x = 0, \eta^{(2)} \in \mathcal{L}_2[0,1]\right\} \\ =& \mathcal{H}_{00} \oplus \mathcal{H}_{01} \oplus \mathcal{H}_1,\end{aligned} \tag{C.2}$$

where $k_1(x) = x - 0.5$. The space of constant terms is $\mathcal{H}_{00\langle 1\rangle} \otimes \mathcal{H}_{00\langle 2\rangle}$; the space of main effects is spanned by $\mathcal{H}_{00\langle 1\rangle} \otimes \left(\mathcal{H}_{01\langle 2\rangle} \oplus \mathcal{H}_{1\langle 2\rangle}\right)$ and $\mathcal{H}_{00\langle 2\rangle} \otimes \left(\mathcal{H}_{01\langle 1\rangle} \oplus \mathcal{H}_{1\langle 1\rangle}\right)$; and the subspace $\left(\mathcal{H}_{01\langle 1\rangle} \oplus \mathcal{H}_{1\langle 1\rangle}\right) \otimes \left(\mathcal{H}_{01\langle 2\rangle} \oplus \mathcal{H}_{1\langle 2\rangle}\right)$ spans the space of interactions. Let $\mathcal{H}_{\nu,\mu} = \mathcal{H}_{\nu\langle 1\rangle} \otimes \mathcal{H}_{\mu\langle 2\rangle}$ for $\nu, \mu = 00, 01, 1$, with inner products $(\eta, \eta)_{\nu,\mu}$ and reproducing kernels $R_{\nu,\mu} = R_{\nu\langle 1\rangle} R_{\mu\langle 2\rangle}$, using the tensor product cubic spline, we have

$$\begin{aligned}J(\eta, \eta) =& \theta_{1,00}^{-1}(\eta, \eta)_{1,00} + \theta_{00,1}^{-1}(\eta, \eta)_{00,1} \\ & + \theta_{1,01}^{-1}(\eta, \eta)_{1,01} + \theta_{01,1}^{-1}(\eta, \eta)_{01,1} + \theta_{1,1}^{-1}(\eta, \eta)_{1,1},\end{aligned} \tag{C.3}$$

and the null space of $J(\eta, \eta)$ is

$$\mathcal{N}_J = \mathcal{H}_{00,00} \oplus \mathcal{H}_{01,00} \oplus \mathcal{H}_{00,01} \oplus \mathcal{H}_{01,01}. \tag{C.4}$$

Appendix D. Regularity Conditions

We define the quadratic functional representing the mean square error of the estimator $\widehat{\eta}$ in estimating the target function η_0 on the domain $\mathcal{X}$ as

$$V\left(\widehat{\eta} - \eta_0\right) = \int_{\mathcal{X}} \left\{\widehat{\eta} - \eta_0(x)\right\}^2 f(x) \mathrm{d}x,$$

where $f(x)$ is the marginal density of x. We now state some regularity conditions for Theorem 1 and Theorem 2.

Condition D.1. *The functional V is completely continuous with respect to J.*

When condition D.1 is satisfied, that is, V is completely continuous with respect to J and hence to $V + J$, there exist eigenvalues λ_ν and the corresponding eigenfunctions ψ_ν such that

$$\begin{aligned}V\left(\psi_\nu, \psi_\mu\right) &= \lambda_\nu \delta_{\nu,\mu}, \text{ and} \\ \left(V + J\right)\left(\psi_\nu, \psi_\mu\right) &= \delta_{\nu,\mu},\end{aligned}$$

where $\delta_{\nu,\mu}$ is the Kronecker delta and $1 \geq \lambda_\nu \downarrow 0$; see Weinberger (1974), Silverman (1982).

Write $\phi_\nu = \lambda_\nu^{-\frac{1}{2}} \psi_\nu$. It follows that

$$\begin{aligned}V\left(\phi_\nu, \phi_\mu\right) &= \delta_{\nu,\mu}, \\ J\left(\phi_\nu, \phi_\mu\right) &= \rho_\nu \delta_{\nu,\mu},\end{aligned}$$

where $0 \leq \rho_\nu = \lambda_\nu^{-1} - 1$. We refer to ρ_ν as the eigenvalues of J with respect to V and to ϕ_ν as the associated eigenfunctions. A Fourier series expansion of η_0 satisfying $J(\eta_0) < \infty$ is $\eta_0 = \sum_\nu \eta_{\nu,0} \phi_\nu$, where $\eta_{\nu,0} = V\left(\eta_0, \phi_\nu\right)$ are the Fourier coefficients.

Condition D.2. $\sum_\nu \rho_\nu^p \eta_{\nu,0}^2 < \infty$ *for some* $p \in [1,2]$.

Condition D.3. *For* ν *sufficiently large and some* $\beta > 0$, *the eigenvalues* ρ_ν *of* J *with respect to* V *satisfy* $\rho_\nu > \beta\nu^r$, *where* $r > 1$.

Condition D.4. *For* η *in a convex set* B_0 *around* η_0 *containing* $\widehat{\eta}$ *and* $\tilde{\eta}$, *where* $\tilde{\eta}$ *is a linear approximation of* $\widehat{\eta}$, $c_1 V(f) \le V_\eta(f)$ *holds uniformly for some* $c_1 > 0$.

Condition D.5. $\mathrm{Var}\left[\phi_\nu(X)\phi_\mu(X)w\left(\eta(X), Y\right)\right] \le c_3$ *for some* $c_3 < \infty$, $\forall \nu, \mu$.

Condition D.5 requires a uniform bound for the fourth moments of $\phi_\nu(X)$.

Condition D.6. *Let* $w\left(\eta; Y\right) = \frac{d^2 l}{d\eta^2}$, *where* $l\left(\eta; Y\right)$ *is the minus log likelihood of* η *with observations* Y. *For* $\tilde{\eta}$ *in a convex set* B_0 *around* η_0 *containing* $\widehat{\eta}$, $c_1 w\left(\eta_0(x); Y\right) \le w(\tilde{\eta}(x); Y) \le c_2 w\left(\eta_0(x); Y\right)$ *holds uniformly for some* $0 < c_1 < c_2 < \infty, \forall x \in \mathcal{X}, \forall Y$.

Condition D.6 requires the equivalence of the information in B_0.

Appendix E. Proofs of Main Results

Proof [Proof of Theorem 1] We start by summarizing the notations used in the theorem. η_0 is the true function. $\widehat{\eta}$ is the estimation based on $\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b)$. For the sake of simplicity, we write $r = 2m$ in the proof. Recall that

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b) = \lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)(\frac{n}{b})^{-\frac{r}{rp+1}}.$$

It suffices to show that as $n \to \infty$,

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b) \to 0, \text{ and}$$

$$n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b))^{\frac{1}{r}} \to \infty.$$

Since $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b) \to 0$ and $(\frac{n}{b})^{-\frac{r}{rp+1}} < 1$, we have

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b) = \lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)(\frac{n}{b})^{-\frac{r}{rp+1}} \to 0.$$

Also, since $rp > 1$, we have

$$\begin{aligned} n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b))^{\frac{1}{r}} &= n(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{1}{r}}(\frac{n}{b})^{-\frac{1}{rp+1}} \\ &= n^{\frac{rp}{rp+1}} b^{\frac{1}{rp+1}} (\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{1}{r}} \\ &\ge b^{\frac{rp}{rp+1}} b^{\frac{1}{rp+1}} (\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{1}{r}} \\ &= b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{1}{r}} \to \infty. \end{aligned} \tag{E.1}$$

Therefore, $n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b))^{\frac{1}{r}} \to \infty$. According to Chapter 9 of Gu (2013), we have

$$\left(V + \lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b) J\right)\left(\widehat{\eta} - \eta_0\right) = O_p\left(n^{-1}\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b)^{-\frac{1}{r}} + \lambda_{\mathrm{KCas}}^{\mathrm{full}}(n; b)^p\right).$$

■

Proof [Proof of Theorem 2] Analogous to Theorem 1, it suffices to show that as $n \to \infty$,

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b) \to 0, \text{ and}$$
$$n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b))^{\frac{2}{r}} \to \infty.$$

Since $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b) \to 0$ and $(\frac{n}{b})^{-\frac{r}{rp+1}} < 1$, we have

$$\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b) = \lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)(\frac{n}{b})^{-\frac{r}{rp+1}} \to 0.$$

Also, since $rp > 1$, analogous to Equation (E.1), we have

$$n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b))^{\frac{2}{r}} \geq b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{2}{r}} \to \infty.$$

Therefore, $n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b))^{\frac{2}{r}} \to \infty$. According to Chapter 9 of Gu (2013), we have

$$(V + \lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)J)(\widehat{\eta} - \eta_0) = O_p\left(n^{-1}\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)^{-\frac{1}{r}} + \lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)^p\right).$$

■

Note that it has been proved rigorously that the optimal smoothing parameter $\lambda(b)$ has the form $Cb^{-\frac{r}{rp+1}}$ under some exponential regression problems, such as regression with Gaussian-type responses and periodic splines (Wahba, 1977, 1985; Craven and Wahba, 1978). In such cases, with the fact that $rp > 1$, as $b \to \infty$,

$$\lambda(b) = Cb^{-\frac{r}{rp+1}} \to 0, \text{ and}$$
$$b\lambda(b)^{\frac{2}{r}} = bC^{\frac{2}{r}}b^{-\frac{2}{rp+1}}$$
$$= C^{\frac{2}{r}}b^{\frac{rp-1}{rp+1}} \to \infty,$$

that is, $\lambda(b) \to 0$ and $b(\lambda(b))^{\frac{2}{r}} \to \infty$ is naturally satisfied. In some cases, such as the density estimation problems and more general exponential-family settings, we impose the stated assumptions on $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)$; the numerical results support their validity.

We replace $\lambda(b)$ with $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)$ chosen by GCV since it is infeasible to determine $\lambda(b)$ with the unknown function η_0. Theoretical results (Li, 1986; Craven and Wahba, 1978) have shown that $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)$ is a good estimator of $\lambda(b)$, with $\frac{L(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))}{L(\lambda(b))} = 1 + o_p(1)$. Thus, it is natural to extend the assumption $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b) \to 0$ and $b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{\frac{2}{r}} \to \infty$ to the general regression problems with responses from exponential families. The numerical results also support this assumption.

Appendix F. Simulation Details

F.1 Nonparametric regression

Scenario 1: Let

$$\eta_{m1}(x) = \sum_{i=1}^{3} g_1\left(x_{\langle i\rangle}\right) + g_2\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}\right) + g_2\left(x_{\langle 1\rangle}, x_{\langle 3\rangle}\right) + g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right),$$

Scenario 2: Let

$$\eta_{m2}(x) = \sum_{i=1}^{3} i g_1\left(x_{\langle i\rangle}\right) + \sum_{i=4}^{6} i g_5\left(x_{\langle i\rangle}\right) + \sum_{i=7}^{9} g_4\left(x_{\langle i\rangle}\right) + \sum_{i=1}^{3}\sum_{j>i}^{4} 3i g_2\left(x_{\langle i\rangle}, x_{\langle j\rangle}\right) + 6 g_2\left(x_{\langle 5\rangle}, x_{\langle 6\rangle}\right) + 8 g_6\left(x_{\langle 7\rangle}, x_{\langle 8\rangle}\right) + 10 g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right),$$

where
$g_1(x) = 10^6 x^{11}(1-x)^6$;
$g_2\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}\right) = \exp\left(3 x_{\langle 1\rangle} x_{\langle 2\rangle}\right)$;
$g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right) = \frac{15\sin\left(2\pi x_{\langle 1\rangle}\right)}{2-\sin\left(2\pi x_{\langle 2\rangle} x_{\langle 3\rangle}\right)}$;
$g_4(x) = 10^4 x^3 (1-x)^{10}$;
$g_5(x) = 15 x sin(15x)$;
$g6(x) = \frac{a p_1}{\pi\sigma_1\sigma_2}\exp\left\{-\frac{\left(x_{\langle 1\rangle}-0.2\right)^2}{\sigma_1^2} - \frac{\left(x_{\langle 2\rangle}-0.3\right)^2}{\sigma_2^2}\right\} + \frac{a p_2}{\pi\sigma_1\sigma_2}\exp\left\{-\frac{\left(x_{\langle 1\rangle}-0.7\right)^2}{\sigma_1^2} - \frac{\left(x_{\langle 2\rangle}-0.8\right)^2}{\sigma_2^2}\right\} - b,$
with $\sigma_1 = 0.3$, $\sigma_2 = 0.4$, $p_1 = 0.625$, $p_2 = 0.375$, and $a = b = 4.2$.

Appendix G. Datasets

G.1 Datasets for Density Estimation

- *CD14*: Transcriptions in CD14 single cells. The data contains the abundance information of 13 proteins in 2,096 cells. The data set is available from Stoeckius et al. (2017).

- *AReM*: Activity Recognition system based on Multisensor data fusion Data Set. The dimension is 6 and the sample size is 42,240. The time-domain features including 3 mean values and 3 standard deviations were collected from the multisensor system during a period of time. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

- *ESC*: Embryonic Stem Cell from Mouse (Ouyang et al., 2009). The data concerns mouse embryonic stem cell gene expression and transcription factor association strength. The 4 features that describe the scores of TFAS with KLF4, NANOG, OCT4, and SOX2 of 1,027 genes are used for density estimation. The data set is available at CRAN in *gss* package.

- *MFCC*: Anuran Calls (MFCCs) Data Set. The data is extracted from syllables of anuran (frogs) calls, including 22 variables with a sample size of 7,195. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

All the continuous variables in these datasets are scaled through a min-max normalization.

G.2 Datasets for Nonparametric Regression

- *SUSY*: Supersymmetric Dataset (Baldi et al., 2014). The dataset contains one response and 18 kinematic features $x_{\langle 1\rangle}, \ldots, x_{\langle 18\rangle}$. The full sample size is 5,000,000, and about 54.24% of the responses in the data are from the background process. We consider the full sample GCV as the gold standard, but it is not affordable to compute GCV on the full sample. Therefore, we randomly select a subsample of size $20,000$ and consider this sample to be the "full sample" to compute the GCV, and only use this part of data to conduct all the following analysis in the paper. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

- *WFRN:* Wall-Following Robot Navigation Dataset (Freire et al., 2009). The data is a robot navigating through the room following the wall using 24 ultrasound sensors with a sample size of 19,735. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

- *OCUP*: Room Occupancy Estimation Dataset (Singh et al., 2018). The experimental testbed for occupancy estimation was deployed in a room. The setup consisted of 7 sensor nodes and one edge node in a star configuration with the sensor nodes transmitting data to the edge every 30s using wireless transceivers. The dataset contains one response and 15 features. The full sample size is 10,129. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

- *SHILL*: Shill Bidding Dataset (Alzahrani and Sadaoui, 2018). This is a dataset with a large number of eBay auctions of a popular product. The dataset contains one response and 12 features. The full sample size is 6,321. The data set is available at UCI Machine Learning Repository (Dua and Graff, 2017).

- *CIFAR-10*: The CIFAR-10 dataset (Krizhevsky, 2009) consists of a training set of $50,000$ examples and a test set of $10,000$ examples. Each example in the dataset is a 32×32 color image, spanning 10 different classes of objects such as animals and vehicles. These classes include airplanes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks, each equally represented in the dataset.

  For this dataset, we use a pre-trained convolutional neural network to extract the features from the raw images first. This neural network model consists of two main components: a convolutional layer block and a fully-connected layer block. The convolutional block comprises two sets of convolutional layers with batch normalization and max pooling layers for feature extraction from input images. The fully-connected block contains four linear layers with ReLU activations. The third linear layer with 20 nodes serves as a feature extraction layer, providing a compressed representation of the input features for downstream tasks.

  For this experiment, we convert CIFAR-10 into a binary logistic regression task by setting $Y = 1$ for the car class and $Y = 0$ for all other classes. The MSE is computed on the test set between the binary responses and the fitted conditional probabilities, and the relative MSE is normalized by the MSE of the full-sample estimator.

All the continuous predictors in these datasets are scaled through a min-max normalization. For all the datasets, we apply the cosine diagnostics (Gu, 2013) first for the identifiability and the practical significance of the fitted terms, in order to avoid overfitting and overinterpreting.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of Hirotugu Akaike*, pages 199–213. Springer, 1998.

Ahmad Alzahrani and Samira Sadaoui. Scraping and preprocessing commercial auction data for fraud classification. *arXiv preprint arXiv:1806.00656*, 2018.

Pierre Baldi, Peter Sadowski, and Daniel Whiteson. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.

Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, et al. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 13(2):e1484, 2023.

Denis Bosq. *Nonparametric Statistics for Stochastic Processes: Estimation and Prediction*, volume 110. Springer Science & Business Media, 2012.

Zdravko I. Botev, J. F. Grotowski, and Dirk P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010.

Adrian W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

Yen-Chi Chen, Christopher R Genovese, and Larry Wasserman. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016.

Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31(4):377–403, 1978.

Michal Daszykowski, Beata Walczak, and DL Massart. Representative subset selection. *Analytica chimica acta*, 468(1):91–103, 2002.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Jianqing Fan. *Local polynomial modeling and its applications: monographs on statistics and applied probability 66*. Routledge, 2018.

Luyang Fang, Yongkai Chen, Wenxuan Zhong, and Ping Ma. Bayesian knowledge distillation: a bayesian perspective of distillation with uncertainty quantification. In *Forty-first International Conference on Machine Learning*, 2024.

Luyang Fang, Cheng Meng, Lin Zhao, Tao Wang, Tianming Liu, Wenxuan Zhong, and Ping Ma. Spot: an active learning algorithm for efficient deep neural network training. *Big Data Mining and Analytics*, 8(5):1060–1074, 2025.

Luyang Fang, Haoran Lu, Jiazhang Cai, Tao Wang, Huimin Cheng, Wenxuan Zhong, and Ping Ma. A statistical perspective on knowledge distillation: Foundations, classical methods, and llm extensions. *Annual Review of Statistics and Its Application*, 2026a. Forthcoming.

Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, et al. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions. *Artificial Intelligence Review*, 59(1):17, 2026b.

Frédéric Ferraty and Philippe Vieu. *Nonparametric functional data analysis: theory and practice*. Springer, 2006.

Ananda L Freire, Guilherme A Barreto, Marcus Veloso, and Antonio T Varela. Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *2009 6th Latin American Robotics Symposium (LARS 2009)*, pages 1–6. IEEE, 2009.

Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2023.

Jia-Xing Gao, Da-Quan Jiang, and Min-Ping Qian. Adaptive manifold density estimation. *Journal of Statistical Computation and Simulation*, pages 1–15, 2022.

Gene H Golub and Charles F Van Loan. *Matrix Computations*. JHU press, 2013.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chong Gu. Cross-validating non-gaussian data. *Journal of Computational and Graphical Statistics*, 1(2):169–179, 1992.

Chong Gu. Model diagnostics for smoothing spline anova models. *Canadian Journal of Statistics*, 32(4):347–358, 2004.

Chong Gu. *Smoothing Spline ANOVA Models*, volume 297. Springer, 2013.

Chong Gu. Smoothing spline anova models: R package gss. *Journal of Statistical Software*, 58:1–25, 2014.

Chong Gu and Young-Ju Kim. Penalized likelihood regression: General formulation and efficient approximation. *Canadian Journal of Statistics*, 30(4):619–628, 2002.

Chong Gu and Chunfu Qiu. Smoothing spline density estimation: Theory. *The Annals of Statistics*, 21(1):217–234, 1993.

Chong Gu and Grace Wahba. Minimizing gcv/gml scores with multiple smoothing parameters via the newton method. *SIAM Journal on Scientific and Statistical Computing*, 12(2): 383–398, 1991.

Chong Gu and Jingyuan Wang. Penalized likelihood density estimation: Direct cross-validation and scalable approximation. *Statistica Sinica*, pages 811–826, 2003.

Chong Gu and Dong Xiang. Cross-validating non-gaussian data: generalized approximate cross-validation revisited. *Journal of Computational and Graphical Statistics*, 10(3):581–591, 2001.

Chong Gu, Yongho Jeon, and Yi Lin. Nonparametric density estimation in high-dimensions. *Statistica Sinica*, pages 1131–1153, 2013.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022.

Peter Hall. Using the bootstrap to estimate mean squared error and select smoothing parameter in nonparametric problems. *Journal of Multivariate Analysis*, 32(2):177–203, 1990.

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.

Nathaniel E Helwig, K Alex Shorter, Ping Ma, and Elizabeth T Hsiao-Wecksler. Smoothing spline analysis of variance models: A new tool for the analysis of cyclic biomechanical data. *Journal of Biomechanics*, 49(14):3216–3222, 2016.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy. Learning lightweight lane detection CNNs by self attention distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1013–1021, 2019.

Jianhua Z Huang. Projection estimation in multiple regression with application to functional anova models. *The Annals of Statistics*, 26(1):242–272, 1998.

Yongho Jeon and Yi Lin. An effective method for high-dimensional log-density anova estimation, with application to nonparametric graphical model building. *Statistica Sinica*, pages 353–374, 2006.

M Chris Jones, James S Marron, and Simon J Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996.

Young-Ju Kim and Chong Gu. Smoothing spline gaussian regression: more scalable computation via efficient approximation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 66(2):337–356, 2004.

George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

Germain Kolossov, Andrea Montanari, and Pulkit Tandon. Towards a statistical theory of data selection under weak supervision. In *International Conference on Learning Representations*, volume 2024, pages 41947–41985, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Self-referenced deep learning. In *Asian Conference on Computer Vision*, pages 284–300. Springer, 2018.

Ker-Chau Li. Asymptotic optimality of cl and generalized cross-validation in ridge regression with application to spline smoothing. *The Annals of Statistics*, pages 1101–1112, 1986.

Yi Lin. Tensor product space anova models. *The Annals of Statistics*, 28(3):734–755, 2000.

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, 34(5):2272–2297, 2006.

Jiabo Ma, Zhengrui Guo, Fengtao Zhou, Yihui Wang, Yingxue Xu, Jinbang Li, Fang Yan, Yu Cai, Zhengjie Zhu, Cheng Jin, et al. A generalizable pathology foundation model using a unified knowledge distillation pretraining framework. *Nature Biomedical Engineering*, 10 (3):545–564, 2026.

Ping Ma, Nan Zhang, Jianhua Z Huang, and Wenxuan Zhong. Adaptive basis selection for exponential family smoothing splines with application in joint modeling of multiple sequencing samples. *Statistica Sinica*, pages 1757–1777, 2017.

Cheng Meng, Xinlian Zhang, Jingyi Zhang, Wenxuan Zhong, and Ping Ma. More efficient approximation of smoothing splines via space-filling basis selection. *Biometrika*, 107(3): 723–735, 2020.

Hossein Mobahi, Mehrdad Farajtabar, and Peter Bartlett. Self-distillation amplifies regularization in Hilbert space. *Advances in Neural Information Processing Systems*, 33: 3351–3361, 2020.

Thomas Nagler and Claudia Czado. Evading the curse of dimensionality in nonparametric density estimation with simplified vine copulas. *Journal of Multivariate Analysis*, 151: 69–89, 2016.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

Zhengqing Ouyang, Qing Zhou, and Wing Hung Wong. Chip-seq of transcription factors predicts absolute and differential gene expression in embryonic stem cells. *Proceedings of the National Academy of Sciences*, 106(51):21521–21526, 2009.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems*, 30, 2017.

Aritz Pérez, Pedro Larrañaga, and Iñaki Inza. Bayesian classifiers based on kernel density estimation: Flexible classifiers. *International Journal of Approximate Reasoning*, 50(2): 341–362, 2009.

Mary Phuong and Christoph H Lampert. Distillation-based training for multi-exit architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1355–1364, 2019.

Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78, 1982.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.

Simon J Sheather and Michael C Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)*, 53(3):683–690, 1991.

Bernard W Silverman. On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, pages 795–810, 1982.

Bernard W Silverman. *Density Estimation for Statistics and Data Analysis*. Routledge, 2018.

Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner, and Vishal Garg. Machine learning-based occupancy estimation using multivariate sensor nodes. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE, 2018.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Marlon Stoeckius, Christoph Hafemeister, William Stephenson, Brian Houck-Loomis, Pratip K Chattopadhyay, Harold Swerdlow, Rahul Satija, and Peter Smibert. Large-scale simultaneous measurement of epitopes and transcriptomes in single cells. *Nature methods*, 14(9): 865, 2017.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

Mervyn Stone. Cross-validation: A review. *Statistics: A Journal of Theoretical and Applied Statistics*, 9(1):127–139, 1978.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Xiaoxiao Sun, David Dalpiaz, Di Wu, Jun S Liu, Wenxuan Zhong, and Ping Ma. Statistical inference for time course rna-seq data using a negative binomial mixed-effect model. *BMC Bioinformatics*, 17(1):1–13, 2016.

Xiaoxiao Sun, Wenxuan Zhong, and Ping Ma. An asymptotic and empirical smoothing parameters selection method for smoothing spline anova models in large samples. *Biometrika*, 108(1):149–166, 2021.

Grace Wahba. Practical approximate solutions to linear operator equations when the data are noisy. *SIAM Journal on Numerical Analysis*, 14(4):651–667, 1977.

Grace Wahba. A comparison of gcv and gml for choosing the smoothing parameter in the generalized spline smoothing problem. *The Annals of Statistics*, pages 1378–1402, 1985.

Grace Wahba. *Spline Models for Observational Data*. SIAM, 1990.

Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.

HaiYing Wang, Rong Zhu, and Ping Ma. Optimal subsampling for large sample logistic regression. *Journal of the American Statistical Association*, 113(522):829–844, 2018.

Yuedong Wang. *Smoothing Splines: Methods and Applications*. CRC press, 2011.

Hans F Weinberger. *Variational Methods for Eigenvalue Approximation*. SIAM, 1974.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Simon N Wood. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004.

Dong Xiang and Grace Wahba. A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statistica Sinica*, pages 675–692, 1996.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. *arXiv preprint arXiv:2205.09329*, 2022.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020.

Jingyi Zhang, Cheng Meng, Jun Yu, Mengrui Zhang, Wenxuan Zhong, and Ping Ma. An optimal transport approach for selecting a representative subsample with application in efficient kernel density estimation. *Journal of Computational and Graphical Statistics*, 32 (1):329–339, 2023.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3713–3722, 2019.

Zhilu Zhang and Mert Sabuncu. Self-distillation as instance-specific label smoothing. *Advances in Neural Information Processing Systems*, 33:2184–2195, 2020.