# KNOWLEDGE CASCADE: REVERSE KNOWLEDGE DISTILLATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

With the rapidly growing model complexity in the state-of-the-art machine learning methods, the expensive model training process has rendered the algorithm design and computation resources allocation challenging. To tackle the challenges, we propose the knowledge cascade (KCas), a strategy that reverses the idea of knowledge distillation (KD). While KD compresses and transfers the knowledge learned by a large-and-complex model (teacher model) to a small-and-simple model (student model), KCas inversely transfer the knowledge in a student model to a teacher model. Despite the fact that teacher models are more sophisticated and capable than student models, we show that in KCas, student models can effectively facilitate teacher models building by taking advantage of the statistical asymptotic theories. We demonstrate the outstanding performance of KCas on the nonparametric multivariate functional estimation in reproducing kernel Hilbert space. One of the crucial problems in accomplishing the estimation is the daunting computational cost of selecting smoothing parameters, whose number will increase exponentially as the number of predictors increases. KCas transfers the knowledge about the smoothing parameters of the target function learned from the student model to the teacher model based on empirical and asymptotic results. KCas significantly reduces the computational cost of the smoothing parameter selection process from $O(n^3)$ to $O(n^{3/4})$, while preserving excellent performance. Theoretical analysis of asymptotic convergence rates and extensive empirical evaluations on simulated and real data validate the effectiveness of KCas.

## 1 INTRODUCTION

In recent years, the model complexity of state-of-the-art machine learning models has been growing rapidly. In particular, super-large deep neural networks (DNN) have achieved remarkable success in various applications (Wolf et al., 2020; Dosovitskiy et al., 2020; You et al., 2019; Goodfellow et al., 2014; Hinton et al., 2012). Despite these impressive achievements, such complex models have also brought significant challenges in model development and model deployment. For model development, it usually takes enormous amounts of computation resources and manpower to develop the complex model due to the large data volume, high dimensionality, and complicated data structure. For model deployment, it is challenging to deploy large and complex models to small/moderate computing devices, such as a sensor in a sensor network, due to their limited computation and storage capacities. To surmount the challenges in model deployment, researchers have developed many model compression methods, among which knowledge distillation plays a key role (Urner et al., 2011; Urban et al., 2016; Hinton et al., 2015). The success of the knowledge distillation methods highly depends on a well-trained teacher model, which needs to overcome the challenges of model development. In this paper, we propose the knowledge cascade (KCas), which reverses the process of knowledge distillation, to tackle the challenges in model development.

We shall now present a brief review of knowledge distillation (KD) methods. KD is similar to the process in which students learn from their teachers, where a small-and-simple model (student model) is generally supervised by a large-and-complex model (teacher model). The main idea is to compress knowledge learned by the teacher model to the student model. Specifically, in deep learning, knowledge distillation targets to train a compact neural net (student) by learning the "knowledge" of the complex DNN (teacher), to obtain competitive or even superior performances. The crucial problem is to design an effective learning mechanism. Extensive works have focused on this problem

with various explorations on the choice of : (1) what knowledge the student model should learn from the teacher model (Hinton et al., 2015; Romero et al., 2014; Huang & Wang, 2017; Ahn et al., 2019; Zagoruyko & Komodakis, 2016) (2) distillation schemes (Passalis & Tefas, 2018; Mirzadeh et al., 2020; Li et al., 2020; Asif et al., 2019; Mirzadeh et al., 2020) (3) the design of suitable structures for the teacher and student model (Wang et al., 2018b; Zhu et al., 2018; Polino et al., 2018; Wei et al., 2018; Howard et al., 2017; Xie et al., 2020; Zhang et al., 2018). See Gou et al. (2021) for a comprehensive survey on the recent developments in KD methods.

Despite the success of KD on model deployment, significant challenges still lie in model development. KD methods critically hinge on a well-trained teacher model, which usually are super-large deep neural networks. Such super-large teacher models consume enormous computational resources to train and may not always be available. (Vaswani et al., 2017; Devlin et al., 2018; Brown et al., 2020; Ramesh et al., 2021; Thoppilan et al., 2022) The huge consumption of computational resources also brings environmental concerns due to carbon emissions caused by fueling modern processing hardware. (Strubell et al., 2019; Bender et al., 2021; Patterson et al., 2021). Furthermore, the model training process may break down due to severe convergence problems, i.e., the model does not converge or converge to a wrong one. To thoroughly tackle these challenges in model development, we propose the reverse knowledge distillation, named knowledge cascade (KCas). By reversing the direction of transferring knowledge, KCas lets the knowledge learned from small-and-simple models (student) cascade to large-and-complex models (teacher). This key idea is similar to the conduction of a pilot study in many scientific fields, where the scientists get a relatively small/limited model from the pilot study and use it to provide information and directions for further research and the development of a large-and-complex model.

A natural question is whether reversing the distillation direction is possible, i.e., can a teacher model effectively learn knowledge from a student model? In general, this may not be realistic since a teacher model has a much more complex structure and is capable of capturing more information. In this paper, we show that it is feasible to reverse the distillation direction using our proposed KCas framework, which takes advantage of the statistical asymptotic theories. Specifically, we demonstrate the outstanding performance of KCas on the nonparametric multivariate functional estimation in reproducing kernel Hilbert space (RKHS). In nonparametric functional estimation, the model complexity (number of parameters) scales with the sample size (Gu, 2013b). Thus, the model trained on a large sample is the complex teacher model and a model trained on a small sample is the simple student model. This coincides with the general definition of teacher and student models in the sense that the former is more powerful but more expensive to train. We aim to facilitate the training of the teacher model by transferring knowledge from student models. In the estimation procedure, RKHS is constructed on each dimension of the multivariate function, and then all RKHSs scaled by smoothing parameters are combined together to form the final RKHS. One of the crucial problems in accomplishing this is the daunting computational cost of selecting smoothing parameters, the number of which increases exponentially as the number of predictors (dimension of multivariate function) increases. In KCas, the asymptotic theorem describes the large-sample behavior of the optimal smoothing parameters, which builds a bridge (by a formula) between student and teacher models. Instead of selecting the smoothing parameters on the full sample, we only need to plug in the optimal smoothing parameters estimated by the student model and solve the formula to extrapolate the optimal smoothing parameters to the teacher model. KCas method significantly reduces the computational burden of selecting smoothing parameters in high-dimensional and large samples. Moreover, we demonstrate an amazing fact that the estimators under KCas often perform better than the full-sample estimator, which is supposed to utilize all the available information and takes much more computation time. This fact suggests that KCas can utilize the information more efficiently and is able to avoid possible over-fitting of the full sample estimator.

An idea related to KCas is self-distillation (SD), which is also developed for the effective deployment of complex models. 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 & Sabuncu, 2020; Mobahi et al., 2020; Zhang et al., 2019; Phuong & Lampert, 2019; Yang et al., 2019; Hou et al., 2019; Liu et al., 2018; Lan et al., 2018). However, models in standard SD methods are relatively large, and the model training procedure is accelerated and improved by distilling knowledge from itself. In this sense, KCas differs from SD by utilizing 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 knowledge learned from them. 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) reversed 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. Therefore, our proposed KCas and the associated theories are significantly different from Yuan et al. (2020) and various self-distillation methods.

**Our contributions:**
1. We introduce a novel concept of knowledge cascade—a reverse knowledge distillation, where the teacher models learn knowledge from the student models. 2. We demonstrate that the reverse knowledge distillation is feasible by integrating the asymptotic theory of nonparametric functional estimation to enable the extrapolation from student models to teacher models. 3. We develop the KCas method in the context of multivariate nonparametric function estimation, design the associated algorithm and establish the consistency theory. The estimation by KCas reduces the time complexity of the smoothing parameter estimation process from $O(n^3)$ to $O(n^{3/4})$. 4. With extensive simulation study and real data analysis, we empirically show the effectiveness of KCas, which even outperforms the gold standard in some cases.

## 2 PRELIMINARIES

### 2.1 NONPARAMETRIC FUNCTIONAL ESTIMATION IN REPRODUCING KERNEL HILBERT SPACE

To estimate a function of interest $\eta$ on a generic domain $\mathcal{X}$, we consider the nonparametric penalized loss function,

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

where $L(\eta)$ is the goodness-of-fit (loss) functional, e.g., likelihood function in regression and hinge loss in support vector machine, $J(\eta)$ is the smoothness (penalty) functional, and $\lambda$ is a Lagrange multiplier, controlling the trade-off between the smoothness of $\eta(x)$ and its fidelity to the data.

***Functional ANOVA decomposition.*** Estimating $\eta$ on the product domain $\mathcal{X} = \prod_{j=1}^{d} \mathcal{X}_j$ is a fundamental problem in statistical learning. Numerous methods have been proposed to solve this problem over the years (Jeon & Lin, 2006a; Chen et al., 2016; Lin & 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. One method that alleviates the curse of dimensionality is the decomposition of multivariate functions similar to the classical analysis of variance (ANOVA) decomposition and the associated notions of the main effect and interaction (Gu et al., 2013; Gu & Wang, 2003; Kim & Gu, 2004; Huang, 1998; Jeon & Lin, 2006b). 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 $\eta$ can be decomposed as a sum of a constant term, on-dimensional functions (main effects), two-dimensional functions (two-way interactions), and so on:

$$\eta(x) = \eta\left(x_1, \ldots, x_d\right) = \eta_\emptyset + \sum_j \eta_j\left(x_j\right) + \sum_{j<k} \eta_{j,k}\left(x_j, x_k\right) + \ldots, \tag{2}$$

with the constant in $\eta_\emptyset$, the main effects in $\eta_j$, 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 roughness penalty $J(\eta)$ to $L(\eta)$ in (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 $\mathcal{N}_\eta = \{\eta : J(\eta) = 0\}$. To assist analysis and computation, a metric and geometry should be defined in this space, and the loss (1) needs to be continuous in $\eta$ under this metric. Since the reproducing kernel Hilbert space (RKHS) adequately equips for the purpose, we

consider the space $\mathcal{H}$ as a RKHS with the continuous evaluation $[x]f = f(x)$, reproducing kernel $R(\cdot, \cdot)$, a non-negative definite function satisfying $\langle R(x, \cdot), f(\cdot)\rangle = f(x), \forall \eta \in \mathcal{H}$, where $\langle \cdot, \cdot \rangle$ is the inner product in $\mathcal{H}$. The following theorem guarantees the existence of the minimizer in RKHS.

**Theorem 2.1 ((Wahba, 1990a))** *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 likelihood function, it is usually convex in $\eta$. The quadratic functional $J(\eta)$ is convex (Gu & 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 (1) exists in most cases.

## 3 METHODOLOGY

Nonparametric penalized estimation of the function of interest $\eta$ is a general question that has been applied in lots of fields (Sun et al., 2016; Helwig et al., 2016). Numerous methods are limited to handling large data due to the computational cost of training complex models. In this project, we develop a knowledge cascade method for a general loss function (1), and we illustrate our method in two important cases: density estimation and regression function estimation.

### 3.1 MINIMIZER OF THE LOSS FUNCTION

We then introduce the computation for the minimizer of the loss function (1). Consider a tensor sum decomposition of the RKHS $\mathcal{H} = \mathcal{N}_J \oplus \mathcal{H}_J$. Without loss of generality, we define $J(\eta)$ with tensor-product cubic splines in the following paper. The space $\mathcal{H}_J$ is an RKHS with $J(\eta)$ as the square norm. 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 $H_J$, the minimizer of (1) has the following form according to the Kimeldorf–Wahba representer theorem (Kimeldorf & Wahba, 1971; Wahba, 1990a; Wang, 2011; Gu, 2013c):

$$\eta(x) = \sum_{v=1}^{m} d_v \phi_v(x) + \sum_{i=1}^{n} c_i R_J(x_i, x) = \boldsymbol{\phi}^T \boldsymbol{d} + \boldsymbol{\xi}^T \boldsymbol{c}, \tag{3}$$

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

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 $R_\beta$ is the reproducing kernel in $\mathcal{H}_\beta$. Here the $\theta_\beta s$ are an extra set of smoothing parameters to be selected. We refer to (Gu et al., 2013; Gu, 2013c) for the explicit forms of $\{R_\beta\}_{\beta=1}^{g}$ and $J(\eta)$.

Plugging equation (3) into the penalized likelihood of density estimation (10), the estimation reduces to the minimization of

$$-\frac{1}{n}\mathbf{1}^T(Q\mathbf{c} + S\mathbf{d}) + \log \int \exp\left(\phi^T \mathbf{d} + \xi^T \mathbf{c}\right) \mathrm{d}x + \frac{\lambda}{2}\mathbf{c}^T Q \mathbf{c}, \tag{4}$$

where $Q$ is $n \times n$ with the $(i, j)$th entry $R_J(x_i, x_j)$. Similarly, the estimation of the penalized likelihood functional (12) of regression has the form:

$$\frac{1}{n}\left(\tilde{\boldsymbol{Y}} - S\mathbf{d} - Q\mathbf{c}\right)^T W \left(\tilde{\boldsymbol{Y}} - S\mathbf{d} - Q\mathbf{c}\right) + \frac{\lambda}{2}\mathbf{c}^T Q \mathbf{c}, \tag{5}$$

where $W$ is the weight matrix, and the explicit form of $\tilde{\boldsymbol{Y}}$ and $W$ can be found in Appendix.

### 3.2 KNOWLEDGE CASCADE

Fixing smoothing parameters $\lambda$ and $\theta$, we can estimate the coefficients $\boldsymbol{d}$ and $\boldsymbol{c}$ in (4) or (5) using Cholesky decomposition (Golub & Van Loan, 2013) or Newton-Raphson method (Gu & Qiu, 1993;

Gu, 2013c). Smoothing parameters control the trade-off between the smoothness of $\eta(x)$ and its fidelity to the data. The selection of a large smoothing parameter will lead to oversmoothing, while the selection of a small one will lead to undersmoothing. To make the estimation work in practice, a critical aspect is the selection of $\lambda$ and $\theta$ that delivers reasonable performance, since the solution of (1) is sensitive to $\lambda$ and $\theta$ (Jeon & Lin, 2006a; Gu, 2013a).

One of the most efficient criteria to select the smoothing parameters is generalized cross-validation (GCV) (Gu, 1992; Gu et al., 2013; Gu & Wahba, 1991), which achieves the selection via cross-validation targeting the Kullback-Leibler (KL) loss. Basically, this method consists of two steps: (i) for fixed $\theta$, minimize the KL loss with respect to $\lambda$; (ii) update $\theta$ according to the updated $\lambda$. However, the parameter tuning, especially for $\lambda$ is highly computationally intensive in the high dimensional setting. With all $S$ smoothing parameters tunable, the above iterative algorithm takes $O(Sn^3)$ flops per iteration (Gu & Wahba, 1991) and needs tens of iterations to converge. Here the number of smoothing parameters $S$ increases as the number of multi-way interaction terms grows. In particular, $\eta(x) = \eta\left(x_1, \ldots, x_d\right) = \eta_\emptyset + \sum_j \eta_j\left(x_j\right) + \sum_{j<k} \eta_{j,k}\left(x_j, x_k\right)$, the ANOVA decomposition model 2 truncated at two-way interactions contains $S = d + 3d(d-1)/2$ smoothing parameters. Thus, in our case of the particularly large sample size, it is impractical to apply GCV on the full sample to accomplish the regression and density estimation tasks using the smoothing spline ANOVA model. We propose the knowledge cascade (KCas) method to reduce the computational burden of selecting smoothing parameters to achieve the estimation. In KCas, we aim to let student models learn the smoothing parameters through optimization, whose computational burden is much less than the teacher model, and then transfer the smoothing parameters to teacher models. Here the complex teacher model is the model trained on the full sample, and the simple student model is the model trained on a subsample with sample size $b$. We first illustrate the simplest version of KCas in the regression model with additive noise

$$Y_i = \eta(x_i) + \epsilon(x_i) \tag{6}$$

where $\epsilon_i$ is the white noise process satisfying $E\epsilon(x_i) = 0$, $E(\epsilon(x_i)\epsilon(x_j)) = \sigma^2$ if $x_i = x_j$, $E(\epsilon(x_i)\epsilon(x_j)) = 0$ otherwise. For smoothing splines in $\mathcal{H}^{(m)}$, defined by

$$\begin{aligned}\mathcal{H}^{(m)} = \Big\{ f : f^{(v)} \text{ absolutely continuous for } v = 0, 1, \ldots, m-1, f^{(m)} \in \mathcal{L}_2[0,1], \\ f^{(v)}(0) - f^{(v)}(1) = 0 \text{ for } v = 0, 1, \ldots, m-1 \Big\},\end{aligned} \tag{7}$$

Wahba (1977); Craven & Wahba (1978); Wahba (1985) derive that the optimal smoothing parameter $\lambda$, ignoring $o(1)$ terms, is

$$Cn^{-2m/(2mp+1)}, \tag{8}$$

where $C$ is an unknown constant depending on unknown function $\eta$ (Wahba, 1977) and $p \in [1, 2]$ indicates different additional smoothness conditions. The estimation of $C$ is infeasible since it depends on the unknown true function $\eta$, 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 $\lambda$ when sample size equals $b$ is $Cb^{-2m/(2mp+1)}$ for the same $C$. We estimate the optimal $\lambda^{\text{sub}}_{\text{GCV}}$ estimated on the subsample to infer the constant $C$, and then employ the same $C$ for the full data (Sun et al., 2021). That is,

$$\lambda^{\text{full}}_{\text{KCas}}(n; b) = \lambda^{\text{sub}}_{\text{GCV}}(b)(n/b)^{-2m/(2mp+1)}. \tag{9}$$

Since the smoothing parameters are used to determine the proportion of the roughness penalty on different terms in (2) and this proportion should be stable over different sample sizes, we directly use the optimal $\theta^{\text{sub}}_{\text{GCV}}(b)$ in the full sample. We then generalize the estimator (9) from the regression model with additive noise (6) to a wide range of penalized likelihood estimation problems, including the nonparametric regression in exponential family and density estimation, etc.

***Density estimation.*** Consider the situation that we have independently identically distributed (iid) 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 apparently not an appealing estimate when the domain $X$ is continuous. Thus, a penalized likelihood estimate (PLE)

is a good candidate. Two intrinsic constraints coming from the definition of a probability density that $p(\cdot) \geq 0$ and $\int_{\mathcal{X}} p dx = 1$, As the two constraints are not computationally amicable, a typical approach (Silverman, 1982; Gu & Qiu, 1993) is to estimate the log-density $\eta(\cdot)$, which is free of the constraints through the transformation $p(x) = e^{\eta(x)}/\int_{\mathcal{X}} e^{\eta(x)} dx$. Silverman (1982) proposed and studied the theoretical properties of the penalized likelihood over a Hilbert space $\mathcal{H}$:

$$-\frac{1}{n}\sum_{i=1}^{n} \eta(X_i) + \int_{\mathcal{X}} e^{\eta(x)} dx + \frac{\lambda}{2} J(\eta). \tag{10}$$

***Nonparametric regression.*** Consider the exponential family with the densities of the form

$$f(y \mid x) = \exp\{(y\eta(x) - b(\eta(x)))/a(\phi) + c(y, \phi)\}, \tag{11}$$

where $a(\cdot) > 0$, $b$, and $c$ are known functions, $\eta(x)$ is the regression function via the link $\eta$, and $\phi$ 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 penalized likelihood functional

$$-\frac{1}{n}\sum_{i=1}^{n} \{Y_i \eta(x_i) - b(\eta(x_i))\} + \frac{\lambda}{2} J(\eta). \tag{12}$$

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

We propose to use KCas to transfer the knowledge smoothing parameters in (10) or (12) to the teacher model. The KCas algorithm is summarized in the following Algorithm 1.

**Algorithm 1** KCas for nonparametric function estimation

**Input:** Data $X$, subsample size $b$

1: Select a subsample $X_b$ of size $b$ from the full sample of size $n$ using uniform sampling and apply GCV on $X_b$ to estimate the smoothing parameters.
2: Find the solution of (1) for the full sample of size $n$ using the fixed smoothing parameter $\lambda^{\text{full}}_{\text{KCas}}(n; b) = \lambda^{\text{sub}}_{\text{GCV}}(b)(n/b)^{-2m/(2mp+1)}$ 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 estimator $\widehat{\eta}$.

**Output:** Estimator $\widehat{\eta}$.

In the first step, we apply uniform sampling to select subsample and 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., 2018a; Meng et al., 2020; Daszykowski et al., 2002). The total number of operations required for each iteration is generally $4n^3/3 + O(n^2)$, in which the selection of 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 $O(Sb^3)$. As suggested by our simulation study, it is sufficient to take $b = O(n^{1/4})$ to obtain excellent performance as good as the full-sample estimator. Therefore, Algorithm 1 can significantly reduce the computational cost of the smoothing parameter estimation process to $O(Sn^{3/4})$. Thus the vital role of KCas is further demonstrated.

### 3.3 Theoretical analysis

In this section, we present the theoretical properties of the smoothing parameters $\lambda$ selected according to Algorithm 1. For notational simplicity, in the following, we will use $\lambda$ to represent all the smoothing parameters, not just the $\lambda$ in front of $J(\eta)$. Please see Appendix B for the proofs.

**Theorem 3.1 (convergence rate of the estimation)** *For the regression in exponential families as in (11), such that $\sum_\nu \rho_\nu^p \eta_{\nu,0}^2 < \infty$ for some $p \in [1, 2]$, under the regularity conditions A.1 to A.4 in Appendix A, assuming $\lambda^{\text{sub}}_{\text{GCV}}(b) \to 0$ and $b(\lambda^{\text{sub}}_{\text{GCV}}(b))^{2/r} \to \infty$ as $b \to \infty$ , we have*

$$(V + \lambda^{\text{full}}_{\text{KCas}}(n; b) J)(\widehat{\eta} - \eta) = O_p\left(n^{-1}\lambda^{\text{full}}_{\text{KCas}}(n; b)^{-1/r} + \lambda^{\text{full}}_{\text{KCas}}(n; b)^p\right). \tag{13}$$

*where $V(\cdot)$ is a interpretable metric such that a small $V(\widehat{\eta} - \eta)$ indicates a good estimate.*

With theorem 3.1, the consistency is ensured and the convergence rate is specified for the estimation $\widehat{\eta}$ based on KCas. For the density estimation problem, the GCV score does not have an explicit form and is expensive to compute. Thus, we use the well-accepted generalized approximate cross-validation (GACV) method (Gu & Xiang, 2001) to release the computational burden. The convergence rate of the KCas estimation based on GACV has not been approved rigorously, however, we demonstrate in numerical examples that the KCas empirically works very well for the density estimation problems, as it can outperform the full sample GACV.

## 4 Simulation Study

Simulation studies are carried on to assess the performance of the proposed KCas method on both the density estimation problem and the regression problem for exponential families. We compare our method with GCV/GACV(Gu et al., 2013), GCV in generalized additive models (GAM) (Wood, 2004), SKIP method (Gu, 2014), and order-based method (ORD) (Hall, 1990). For the order-based method, we directly use $n^{-r/(pr+1)}$ as the smoothing parameter $\lambda$ where $n$ is the full sample size. GCV is taken as the benchmark method to compare. SKIP method speeds up the computation by picking a suitable starting point and skipping the subsequent iterative steps. However, SKIP fails to converge in density estimation with relatively high dimensions, so SKIP is only included for comparison in the nonparametric regression problem. For the density estimation problem, we further include kernel density estimation (KDE), where we utilize the version that is proposed by Nagler & Czado (2016) to fit high-dimensional data. Note that all relative measures in the following content mean dividing by the performance of GCV/GACV on the full sample.

For the proposed KCas method, we use the uniform sampling scheme to select the subsample, and the subsample size is set to be $b = 50n^{1/4}$. The full sample size is set to be $5,000$, $10,000$, and $20,000$. All the results are based on 30 replications.

### 4.1 Simulation 1: density estimation

We evaluate the methods using log-transformed relative efficacy. The relative efficacy is defined as $D_{\mathrm{KL}}(\widehat{P}\|P)/D_{\mathrm{KL}}(\tilde{P}\|P)$, where $D_{\mathrm{KL}}(Q\|P)$ is the Kullback–Leibler divergence from $P$ to distribution $Q$, $P$ is the true distribution, $\widehat{P}$ is the estimator for the method being evaluated, and $\tilde{P}$ is the benchmark method. The lower the log-transformed relative efficacy, the better the performance.
**Scenario 1:** A $d$-dimensional Gaussian mixture model is constructed with the density $1/d\sum_{i=1}^{d} Gaussian(e_i, I_d)$, where $e_i$ is the vector with the $i^{\text{th}}$ entry equals to 1 and others equal to 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 $0.5(\mathbf{1}\mathbf{1}') + 1.5I$, and the remaining $d - 5$ variables are iid from Unif$(0, 1)$. We consider $d = 15, 20$.

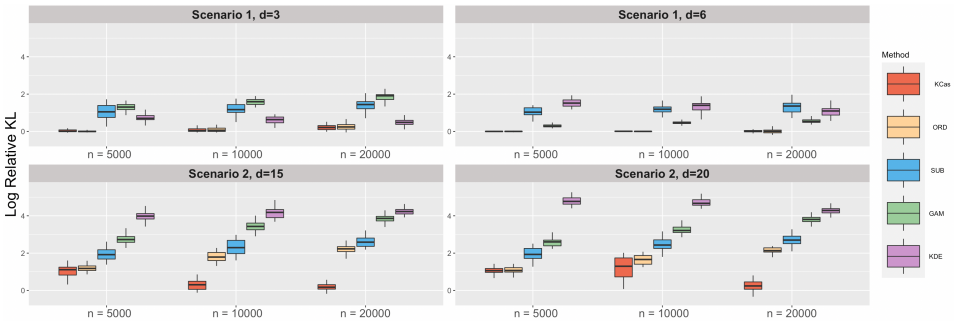


Figure 1: Comparison of five methods on density estimation problem using log relative efficacy.

The log-transformed relative efficacies of the proposed KCas method and the other four methods are shown in Fig. 1. The performance of KCas compares favorably with the other four methods, with log-transformed relative efficacies close to or even lower than 0. The negative values mean that our proposed method performs better than the benchmark method GCV. One reasonable explanation is

that the knowledge learned by the student model can help the teacher model get rid of the influence of noise, thus further improving the teacher model. The SKIP method also has competitive performance with the proposed KCas method when the dimension is low and the sample size is small. But as the dimension and sample size increase, i.e. the model complexity increases, it will lose its power. Specifically, in scenario 2 when the sample size is $10,000$ or $20,000$, the median of log-transformed relative efficacies is about 2, which implies that the median KL divergence of the order-based method is $e^2 \approx 7.4$ times as large as the GCV method on the full sample.

### 4.2 Simulation 2: nonparametric regression

For the nonparametric regression problem, we consider the model: $y_i \sim Ber(\frac{\exp(\eta(x_i))}{1+\exp(\eta(x_i))})$, where $Ber(p)$ is 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 $i$th observation, where each entry is independently draw from $\mathrm{Unif}(0,1)$. $\eta$ is the nonparametric function determining the success probability in the Bernoulli trial, and $y_i \in \mathbb{R}$ is the response variable for the $i$th observation. We evaluated the methods by log-transformed relative efficacy $RMSE = \sum_{i=1}^{n}\left\{\widehat{\eta}\left(x_i\right)-\eta\left(x_i\right)\right\}^2 / \sum_{i=1}^{n}\left\{\tilde{\eta}\left(x_i\right)-\eta\left(x_i\right)\right\}^2$, where $\eta$ is the true function, $\widehat{\eta}$ is the estimator for the method being evaluated, and $\tilde{\eta}$ is the benchmark method. We considered two different scenarios with different dimensions.

**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:** $\eta_{m2}(x) = \sum_{i=1}^{3}\alpha_i g_1\left(x_{\langle i\rangle}\right) + \sum_{i=4}^{6}\alpha_i g_5\left(x_{\langle i\rangle}\right) + \sum_{i=7}^{9} g_4\left(x_{\langle i\rangle}\right)$
$+\sum_{i=1}^{3}\sum_{j>i}^{4}\beta_i g_2\left(x_{\langle i\rangle}, x_{\langle j\rangle}\right) + \theta_1 g_2\left(x_{\langle 5\rangle}, x_{\langle 6\rangle}\right) + \theta_2 g_6\left(x_{\langle 7\rangle}, x_{\langle 8\rangle}\right) + \theta_3 g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right)$

Explicit forms of functions $g_i(x)$'s and hyper-parameters $\alpha_1$, $\beta_1$, $\theta_i$'s can be found in Appendix D.1. Scenario 1 is widely used in the research on nonparametric multivariate functional estimation in RKHS (Jeon & Lin, 2006a; Gu & Wahba, 1991; Sun et al., 2021; Gu & Wang, 2003). We considered two situations: $d = 3$ and $d = 6$. When $d = 3$, all three variables contribute to $\eta_{m1}$ and thus contribute to observation $y$. When $d = 6$, we set the last three variables to have no effect on $\eta_{m1}$ to mimic the real situation that some variables are irrelevant to the $y$. Scenario 2 is a more complicated one in high dimensions. We considered $d = 15$ and $d = 20$.

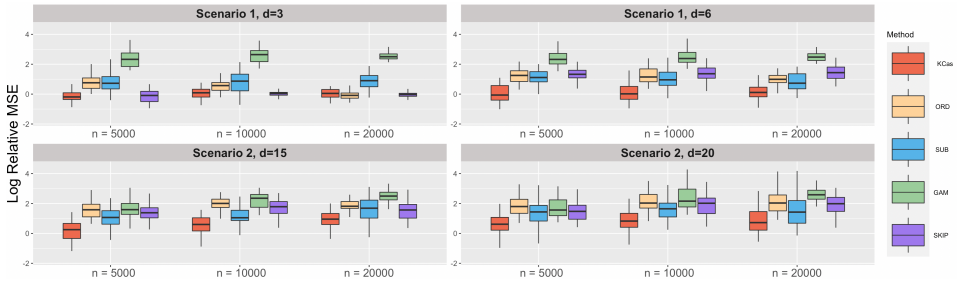


Figure 2: Comparison of five methods on nonparametric regression problem using log relative efficacy.

Table 1: Median computational time (min) for Simulation 2, Scenario 2, $d = 20$

| | 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 |

Log-transformed relative efficacies over GCV are shown in Fig. 2. In scenario 1, for $d = 3$, the performance of KCas and SKIP methods are comparable and are better than the other three methods. The medians of the relative efficacies of these two methods are close to or even less than 0. This implies that in this case KCas and SKIP perform even better than GCV using the full data, as well as benefiting from the faster computation (computational cost is shown in Table 1). For $d = 6$, KCas still shares similar results with GCV, but SKIP has poor performance with the medians of the relative efficacies greater than 1. This phenomenon is understandable since SKIP uses the good starting values introduced by Gu (2014) as the final estimate while skipping the following iterations. When $\eta$ is relatively simple ($d = 3$), SKIP may give us a good estimate by taking advantage of the good starting value, but even the good starting value will generally not be close to the optimal

value when $\eta$ is complex ($d = 6$). Thus, ignoring the subsequent iteration process will make the estimation inaccurate. Comparing the performance in scenario 2, we observe similar phenomenons for both $d = 15$ and $d = 20$ that KCas has the best performance.

## 5 REAL DATA ANALYSIS

We apply KCas for density estimation on 4 benchmark datasets, and for nonparametric regression on 2 benchmark datasets. See Appendix E for the details of the datasets. The features are scaled through a max-min transformation and then randomly split into 80% training set and 20% test.

***Density estimation.*** We compare KCas with GAM, SUB, ORD, and KDE. Since there is no ground truth of the density function, we evaluate the performance by the average log-likelihood on the test set, as suggested by (Papamakarios et al., 2017; Gao et al., 2022). We consider the ANOVA decomposition of $\eta$ including all main effects and all two-way interactions. The model terms are selected by the model diagnosis suggested by Gu et al. (2013). Table 2 shows that the proposed KCas outperforms all 4 benchmark methods in terms of log-likelihood on all data sets. On ESC and MFCC datasets, KCas is even better than GCV on the full data, both in terms of log-likelihood and computational time.

Table 2: Real data analysis: density estimation

| | 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 | 0.99084 | 0.9988 | 0.2528 | 0.24 | 0.20 | 0.19 | 0.19 | 1.73 |

***Nonparametric regression.*** We compare KCas with GAM, SUB, ORD, and SKIP. Since we do not know the underlying probability of each data point, as suggested by (Wang et al., 2018a), we calculate the relative mean square error (MSE). The main effect and interaction terms are selected by the smoothing spline ANOVA model diagnostics (Gu, 2004). Table 3 shows that KCas outperforms all 4 benchmark methods in terms of log-likelihood. 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 comparing methods.

Table 3: Real data analysis: nonparametric regression

| | 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 |

## 6 CONCLUSION

In this article, we propose the knowledge cascade (KCas), a reversed version of knowledge distillation. We show that although letting the teacher model learn from the student is challenging, KCas accomplishes this task excellently by taking advantage of the statistical asymptotic theories. We demonstrate KCas on the nonparametric functional estimation in the Hilbert space to help select smoothing parameters. Owing to the assistance of the information learned from the student model, the KCas method dramatically reduces the computational cost. Our simulation shows that KCas compares favorably with other smoothing parameter selection methods targeting to reduce the computational cost. It is worth emphasizing that KCas could perform better than the benchmark method GCV in some instances. One reasonable explanation is that the knowledge learned by the student model can help the teacher model alleviate the impact of spurious noise. Our knowledge cascade idea offers new insights into the knowledge distillation area. It is interesting to see whether the knowledge cascade is feasible without the aid of asymptotic theory.

## REFERENCES


Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Umar Asif, Jianbin Tang, and Stefan Harrer. Ensemble knowledge distillation for learning improved and efficient networks. *arXiv preprint arXiv:1909.08097*, 2019.

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.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.

Denis Bosq. *Nonparametric statistics for stochastic processes: estimation and prediction*, volume 110. Springer Science & Business Media, 2012.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

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

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)*, pp. 1–6. IEEE, 2009.

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

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

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

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 Science & Business Media, 2013a.

Chong Gu. *Smoothing spline ANOVA models*, volume 297. Springer Science & Business Media, 2013b.

Chong Gu. *Smoothing Spline ANOVA Models (2nd Ed.)*. Springer-Verlag, New York, 2013c.

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

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*, pp. 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*, pp. 1131–1153, 2013.

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.

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.

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*, pp. 1013–1021, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

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

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Yongho Jeon and Yi Lin. An effective method for high-dimensional log-density anova estimation, with application to nonparametric graphical model building. *Statistica Sinica*, 16(2):353–374, 2006a. ISSN 10170405, 19968507. URL `http://www.jstor.org/stable/24307548`.

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

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.

Xu Lan, Xiatian Zhu, and Shaogang Gong. Self-referenced deep learning. In *Asian conference on computer vision*, pp. 284–300. Springer, 2018.

Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14639–14647, 2020.

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

Ruishan Liu, Nicolo Fusi, and Lester Mackey. Teacher-student compression with generative adversarial networks. *arXiv preprint arXiv:1812.02271*, 2018.

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.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5191–5198, 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.

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.

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 268–284, 2018.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

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*, pp. 1355–1364, 2019.

Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. *arXiv preprint arXiv:1802.05668*, 2018.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

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

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.

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.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*, 2022.

Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

Ruth Urner, Shai Shalev-Shwartz, and Shai Ben-David. Access to unlabeled data can speed up prediction time. In *ICML*, 2011.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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*, pp. 1378–1402, 1985.

Grace Wahba. *Spline models for observational data*. SIAM, 1990a.

Grace Wahba. *Spline models for observational data*. SIAM, 1990b.

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

Hui Wang, Hanbin Zhao, Xi Li, and Xu Tan. Progressive blockwise knowledge distillation for neural network acceleration. In *IJCAI*, pp. 2769–2775, 2018b.

Yuedong Wang. *Smoothing splines: methods and applications*. CRC press, 2011.

Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan. Quantization mimic: Towards very tiny cnn for object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 267–283, 2018.

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*, pp. 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.

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*, pp. 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*, pp. 2859–2868, 2019.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

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*, pp. 3903–3911, 2020.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

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*, pp. 3713–3722, 2019.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4320–4328, 2018.

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

Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. *Advances in neural information processing systems*, 31, 2018.

# Appendix for "Knowledge Cascade: Reverse Knowledge Distillation"

## A REGULARITY CONDITIONS

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

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

where $f(x)$ is the marginal density of $x$.

We now state four regularity conditions for Theorem 3.1.

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

When condition A.1 is satisfied, that is, $V$ is completely continuous with respect to $J$ and hence to $V+J$, there exists eigenvalues $\lambda_\nu$ and the corresponding eigenfunctions $\psi_\nu$ such that

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

where $\delta_{\nu,\mu}$ is the Kronecker delta and $1\geq\lambda_\nu\downarrow 0$; see weinberger1974variational, silverman1982est.

Write $\phi_\nu=\lambda_\nu^{-1/2}\psi_\nu$. 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 $\rho_\nu$ as the eigenvalues of $J$ with respect to $V$ and to $\phi_\nu$ as the associated eigenfunctions. A Fourier series expansion of $\eta$ satisfying $J(\eta)<\infty$ is $\eta=\sum_\nu\eta_{\nu,0}\phi_\nu$, where $\eta_{\nu,0}=V\left(\eta,\phi_\nu\right)$ are the Fourier coefficients.

**Condition A.2** *For $\nu$ sufficiently large and some $\beta>0$, the eigenvalues $\rho_\nu$ of $J$ with respect to $V$ satisfy $\rho_\nu>\beta\nu^r$, where $r>1$.*

**Condition A.3** *Let $w\left(\eta;Y\right)=d^2l/d\eta^2$, where $l\left(\eta;Y\right)$ is the minus log likelihood of $\eta$ with observations $Y$. For $\tilde{\eta}$ in a convex set $B_0$ around $\eta$ containing $\widehat{\eta}$, $c_1w\left(\eta(x);Y\right)\leq w(\tilde{\eta}(x);Y)\leq c_2w\left(\eta(x);Y\right)$ holds uniformly for some $0<c_1<c_2<\infty,\forall x\in\mathcal{X},\forall Y$.*

Condition A.3 asks for the equivalence of the information in $B_0$.

**Condition A.4** $\mathrm{Var}\left[\phi_\nu(X)\phi_\mu(X)w\left(\eta(X),Y\right)\right]\leq c_3$ *for some $c_3<\infty$, $\forall\nu,\mu$.*

Condition A.4 requires a uniform bound for the fourth moments of $\phi_\nu(X)$.

## B PROOF OF THEOREM 3.1

**Theorem 3.1** (convergence rate of the estimation) *For the regression in exponential families as in (11), such that $\sum_\nu\rho_\nu^p\eta_{\nu,0}^2<\infty$ for some $p\in[1,2]$, under Conditions A.1 - A.4, assuming $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)\to 0$ and $b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r}\to\infty$ as $b\to\infty$, we have*

$$\left(V+\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)J\right)\left(\widehat{\eta}-\eta\right)=O_p\left(n^{-1}\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)^{-1/r}+\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)^p\right).\tag{14}$$

We start with summarizing the notations used in the theorem. $\eta$ is the true function. $\widehat{\eta}$ is the estimation based on $\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b)$. Note that $r=2m$. Recall that

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

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

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

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

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

Also, since $rp > 1$, we have

$$\begin{aligned} n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b))^{2/r} &= n(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r}(n/b)^{-2/(rp+1)} \\ &= n^{(rp-1)/(rp+1)}b^{2/(rp+1)}(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r} \\ &\geq b^{(rp-1)/(rp+1)}b^{2/(rp+1)}(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r} \\ &= b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r} \to \infty. \end{aligned}$$

Therefore, $n(\lambda_{\mathrm{KCas}}^{\mathrm{full}}(n;b))^{2/r} \to \infty$. According to Chapter 9 of gu2013smooth , we have

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

Note that it has been proved rigorously that the optimal smoothing parameter $\lambda(b)$ has the form $Cb^{-r/(rp+1)}$ under some exponential regression problems such as regression with Gaussian-type responses and periodic splines wahba1977practic, wahba1985compar,craven1978smooth . In such cases, with the fact that $rp > 1$, as $b \to 0$,

$$\begin{aligned} \lambda(b) &= Cb^{-r/(rp+1)} \to 0, \text{ and} \\ b\lambda(b)^{2/r} &= bC^{2/r}b^{-2/(rp+1)} \\ &= C^{2/r}n^{(rp-1)/(rp+1)} \to \infty. \end{aligned}$$

That is, $\lambda(b) \to 0$ and $b(\lambda(b))^{2/r} \to \infty$ is naturally satisfied. In some cases, such as the density estimation problems, there is no strict proof that the optimal $\lambda$ still follows the form of $Cb^{-r/(rp+1)}$, but we can reasonably infer that it should be similar to this form. 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 $\eta$. Theoretical results li1986asymptotic,craven1978smooth have shown that $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b)$ is a good estimator of $\lambda(b)$, with $L(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))/L(\lambda(b)) = 1 + o_p(1)$. Thus, it is natural to expend the assumption $\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b) \to 0$ and $b(\lambda_{\mathrm{GCV}}^{\mathrm{sub}}(b))^{2/r} \to \infty$ to the general regression problems with responses from exponential families. The numerical examples results also support this assumption.

## C CHOICES OF HYPERPARAMETERS $m$, $p$

For $J(\eta,\eta) = \int_0^1 \left(\eta^{(2)}\right)^2 \mathrm{d}x$ on $[0,1]$, $r = 2m = 4$. When $\eta^{(2)}$ is square-integrable, we have $p = 1$, and when $\eta^{(4)}$ is square-integrable, we have $p = 2$. For the tensor product cubic spline, we have $4 - \epsilon < r < 4, \forall \epsilon > 0$ (Wahba, 1990b) . Therefore, in practice we take $r = 4$ and $p = 2$ empirically.

## D SIMULATION DETAILS

### D.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

$$\eta_{m2}(x) = \sum_{i=1}^{3} \alpha_i g_1\left(x_{\langle i\rangle}\right) + \sum_{i=4}^{6} \alpha_i g_5\left(x_{\langle i\rangle}\right) + \sum_{i=7}^{9} g_4\left(x_{\langle i\rangle}\right) + \sum_{i=1}^{3}\sum_{j>i}^{4} \beta_i g_2\left(x_{\langle i\rangle}, x_{\langle j\rangle}\right) + \theta_1 g_2\left(x_{\langle 5\rangle}, x_{\langle 6\rangle}\right) + \theta_2 g_6\left(x_{\langle 7\rangle}, x_{\langle 8\rangle}\right) + \theta_3 g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right),$$

where:

$g_1(x) = 10^6 x^{11}(1-x)^6$;

$g_2\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}\right) = \exp\left(3x_{\langle 1\rangle}x_{\langle 2\rangle}\right)$;

$g_3\left(x_{\langle 1\rangle}, x_{\langle 2\rangle}, x_{\langle 3\rangle}\right) = 15\sin\left(2\pi x_{\langle 1\rangle}\right) / \left\{2 - \sin\left(2\pi x_{\langle 2\rangle}x_{\langle 3\rangle}\right)\right\}$;

$g_4(x) = 10^4 x^3(1-x)^{10}$;

$g_5(x) = 15xsin(15x)$;

$g6(x) = \frac{ap_1}{\pi\sigma_1\sigma_2}\exp\left\{-\frac{\left(x_{\langle 1\rangle}-0.2\right)^2}{\sigma_1^2} - \frac{\left(x_{\langle 2\rangle}-0.3\right)^2}{\sigma_2^2}\right\} + \frac{ap_2}{\pi\sigma_1\sigma_2}\exp\left\{-\frac{\left(x_{\langle 1\rangle}-0.7\right)^2}{\sigma_1^2} - \frac{\left(x_{\langle 2\rangle}-0.8\right)^2}{\sigma_2^2}\right\} - b,$
with $\sigma_1 = 0.3$, $\sigma_2 = 0.4$, $p_1 = 0.625$, $p_2 = 0.375$, and $a = b = 4.2$.

$\alpha_i = i; \quad \beta_i = 3i; \quad \theta_1 = 6; \quad \theta_2 = 8; \quad \theta_3 = 10$

# E DATASETS

## E.1 DATASETS FOR DENSITY ESTIMATION

*CD14*: Transcriptions in CD14 single cells. The data contains the abundance information of 13 proteins in 2096 cells. The data set is available through (Stoeckius et al., 2017).

*AReM*: Activity Recognition system based on Multisensor data fusion Data Set. The dimension is 6 and the sample size is 42240. 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.

*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 1027 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 & Graff, 2017).

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

## E.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 golden standard, but it is not affordable to compute GCV on the full sample. Therefore, we uniformly pick a

subsample of size $20,000$. The data set is available at UCI Machine Learning Repository Dua & Graff (2017).

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

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

## F RESOURCES

All datasets in the real data analysis are public available as described in Section E. The code and instructions for the proposed KCas method are available upon request.