Statistica Sinica Preprint No: SS-2025-0144	
Title	Conditional Density Estimation with Deep Neural Networks
Manuscript ID	SS-2025-0144
URL	http://www.stat.sinica.edu.tw/statistica/
DOI	10.5705/ss.202025.0144
Complete List of Authors	Chenxuan He, Yuan Gao, Liping Zhu and Jian Huang
Corresponding Authors	Liping Zhu
E-mails	zhu.liping@ruc.edu.cn
Notice: Accepted author version.	

CONDITIONAL DENSITY ESTIMATION WITH DEEP NEURAL NETWORKS

Chenxuan He[1], Yuan Gao[2], Liping Zhu[1] and Jian Huang[3]

[1]*Renmin University of China,* [2]*Nankai University*

and [3]*The Hong Kong Polytechnic University*

Abstract: Estimating conditional density functions is a fundamental problem in statistics. This task is crucial for understanding the underlying relationships between variables and for making informed predictions in various applications. In this paper, we introduce a novel deep nonparametric approach for estimating conditional density functions from data. Our method leverages the flexibility and expressiveness of deep neural networks to model the conditional density without imposing restrictive parametric assumptions. We formulate the problem of conditional density estimation as a nonparametric least squares problem, which allows us to harness the strengths of deep learning in a principled manner. By framing the problem this way, we can effectively utilize deep neural networks to approximate the conditional density function. We demonstrate that our proposed approach achieves the minimax optimal convergence rate for conditional density estimation. Additionally, we show that the convergence rate can be further improved for high-dimensional data satisfying a low-dimensional manifold assumption. To validate the performance of our approach, we conduct extensive numerical evaluations on both simulated and real-world datasets. These experiments reveal that our method consistently outperforms several established techniques, highlighting its superior accuracy and robustness in diverse scenarios.

Key words and phrases: Conditional density estimation, Deep neural networks, Optimal convergence rate, Nonparametric regression.

1. Introduction

Conditional density estimation is a fundamental task in statistical analysis. Let $(\mathbf{X}, Y) \in \mathbb{R}^d \times \mathbb{R}$ be a pair of random variables. The conditional probability density function of $Y = y$ given $\mathbf{X} = \mathbf{x}$ is

$$f_*(y \mid \mathbf{x}) = \frac{f(\mathbf{x}, y)}{f(\mathbf{x})},$$

where $f(\mathbf{x})$ and $f(\mathbf{x}, y)$ are the probability density functions of $\mathbf{X}$ and $(\mathbf{X}, Y)$, respectively, with $f(\mathbf{x})$ being positive. Estimating the conditional density is crucial for understanding and modeling the conditional distribution, particularly in scenarios involving multi-modal and skewed distributions. By capturing the data distribution, we can extract various quantities of interest, such as expectations, quantiles, modes, and more. We can then enhance our understanding of the underlying data and conduct more detailed statistical analyses. The applications of conditional density estimation are diverse, encompassing a wide range of fields, including galaxy spectra and photometric redshift estimation in astronomy (Izbicki and Lee 2016,Izbicki and Lee 2017, Izbicki et al. 2017) and typhoon intensity estimation in meteorology (Cloud et al. 2019, Huberman et al. 2022).

Numerous approaches have been proposed for conditional density estimation. Classical kernel approaches use kernel functions to approximate the conditional density (Rosenblatt 1956, Parzen 1962). Fan et al. (1996) introduced a local linear estimator to estimate the density at a query point using kernel functions. Hyndman and Yao (2002) improved upon it and provided a convergence analysis. In mixture models, we typically assume the density to be a mixture of parametric densities and estimate it by learning the parameters of

the mixture density model. These models include Gaussian mixtures, log-normal mixtures, Cauchy mixtures, and others (Bishop 1994, Ruzgas et al. 2021, Samani et al. 2021, Yan et al. 2023). Eigenfunction-based approaches decompose the density function into a series of basis functions (Lindgren 2012, Efromovich 2007, Izbicki and Lee 2016). However, most existing work fails to address the high-dimensional settings or derive convergence rates. Specifically, classical kernel-based methods face significant computational challenges and achieve unsatisfactory performance as data dimensionality increases (Scott, 2015). In contrast, popular deep learning-based approaches perform well on high-dimensional data, but they fall short in the convergence analysis for the conditional density estimation, particularly with regard to the optimal convergence rate (Zhou et al., 2023).

Recently, deep learning has emerged as a powerful tool for (conditional) density estimation, with various network architectures proposed to capture complex data distributions. There are several methods that integrate neural networks with kernel mixture models (Ambrogioni et al., 2017; Rothfuss et al., 2019) and a wide range of generative models, such as generative adversarial networks, normalizing flows, and diffusion models (Goodfellow et al., 2014; Rezende and Mohamed, 2015; Ho et al., 2020). While these approaches have shown remarkable empirical success, establishing their theoretical foundations remains an active research topic. A more detailed review of these methods and their theoretical properties is provided in Section 7.

In this work, we propose a nonparametric approach to estimating the conditional density $f_*(y \mid \mathbf{x})$ using deep neural networks and establish its convergence rate under some regularity conditions. To implement this approach, we first recast the conditional density estimation as a nonparametric regression problem via kernel approximations. The practicable kernel func-

tions range from bounded to unbounded, and can be selected based on the smoothness of the functions. Subsequently, we solve the nonparametric regression problem using deep neural networks. We prove that our proposed approach achieves the minimax optimal convergence rate (Stone, 1982). We apply our approach to analyze four simulated datasets and two real datasets and demonstrate its effectiveness with numerical comparisons. Our contributions are summarized as follows:

1. We propose a deep nonparametric regression approach to conditional density estimation. The approach is model-free since it does not depend on explicit parametric assumptions and adapts well to complex data structures.

2. We prove that the proposed estimator achieves the minimax optimal convergence rate, up to a polynomial factor in $\log\ n$. We propose a new oracle inequality for error decomposition that properly addresses the kernel-induced bias. To the best of our knowledge, this technique is novel and essential to our analysis.

3. Our method reduces computational complexity by estimating conditional density globally over $\mathbf{x}$, which circumvents point-wise weight computations. In high-dimensional scenarios, this global approach gains in more importance since the local methods may incur substantial computational costs.

4. Our theoretical results shed light on the efficacy of the proposed approach in addressing the curse of dimensionality. We prove that the convergence rate can be further improved for high-dimensional data satisfying a low-dimensional manifold assumption. The capacity of tackling high-dimensional data with low-dimensional structures supports the practical scalability and versatility of our approach in conditional density

estimation tasks.

The rest of this paper is organized as follows. We first provide some preliminaries in Section 2. In Section 3, we introduce our methodology for conditional density estimation. We give the convergence rate of our estimator in Section 4. We further extend this approach to the case with varying response y in Section 5. Simulations and real data analyses are presented in Section 6. We discuss related work in Section 7, and the conclusion is given in Section 8.

2. Preliminaries

In the preliminaries, we present essential notations and definitions for our subsequent analyses. Additionally, we introduce deep neural networks that serve as a crucial component of our approach.

2.1 Notations and Definitions

Let $\mathbb{S}_{\mathbf{X}}$ and $\mathbb{S}_Y$ denote the supports of $\mathbf{X}$ and Y. Let d denote the dimension of the exposures $\mathbf{X}$. Let $\mathbf{Z} := (\mathbf{X}, Y) \in \mathbb{S}_{(\mathbf{X},Y)} \subseteq \mathbb{R}^{d+1}$ be a random vector, where $\mathbb{S}_{(\mathbf{X},Y)}$ is the support of $(\mathbf{X}, Y)$, $Y \in \mathbb{S}_Y \subseteq \mathbb{R}$ is univariate, and $\mathbf{X} = (X_1, \ldots, X_d)^\top \in \mathbb{S}_{\mathbf{X}} \subseteq \mathbb{R}^d$. Let $\mathcal{D}_n := \{\mathbf{Z}_i\}_{i=1}^n$ with $\mathbf{Z}_i = (\mathbf{X}_i, Y_i)$ be a sample drawn from the distribution of $\mathbf{Z}$. Let $\mathbb{E}_{\mathbf{Z}}$ denote expectation with respect to the distribution of a single observation $\mathbf{Z}$, and let $\mathbb{E}_{\mathcal{D}_n}$ denote expectation with respect to the joint distribution of the sample $\mathcal{D}_n = \{\mathbf{Z}_i\}_{i=1}^n$. We also write $\mathbb{E}_n(f) = \sum_{i=1}^n f(\mathbf{Z}_i)/n$ for the empirical mean. For two sequences a_n and b_n, let $a_n = O(b_n)$ denote that there exists a constant C_0 such that $a_n \leq C_0 b_n$. We use $a_n \lesssim b_n$ to show $a_n = O(b_n)$. Similarly, let $a_n \asymp b_n$ denote that $a_n = O(b_n)$ and $b_n = O(a_n)$. We use $\lceil a \rceil$ to denote

the smallest integer no less than a, and $\lfloor a \rfloor$ to denote the largest integer no more than a. Let $a \wedge b$ denote $\min(a, b)$ and $a \vee b$ denote $\max(a, b)$. We use $U(a, b)$ to denote a uniform distribution on the interval (a, b), and $N_{[a,b]}$ to denote the truncated normal distribution within the interval $[a, b]$, where $a < b$. We use $\mathbb{N}^+$ to denote the set of positive natural numbers.

We provide the formal definition of the L_p-norm and the covering number, which is essential for deriving convergence rates.

Definition 1 (L_p-norm)**.** Let ν be a probability measure on $\mathbb{R}^d$, and let $f : \mathbb{R}^d \to \mathbb{R}$ be an arbitrary function. We define the $L_p(\nu)$-norm and the empirical $L_p(\nu_n)$-norm with the sample $\mathcal{D}_n$ as

$$\|f\|_p = \left(\int |f(\mathbf{x})|^p \mathrm{d}\nu \right)^{1/p}, \quad \|f\|_{p,n} = \left(\sum_{i=1}^{n} |f(\mathbf{X}_i)|^p / n \right)^{1/p},$$

if $p \in [1, \infty)$ and

$$\|f\|_p = \sup_{\mathbf{x} \in \mathbb{S}_\mathbf{X}} |f(\mathbf{x})|, \quad \|f\|_{p,n} = \max_{i=1,\ldots,n} |f(\mathbf{X}_i)|,$$

if $p = \infty$.

Definition 2. (Covering number, Györfi et al., 2002, Definitions 9.1, 9.2, 9.3) Let $\varepsilon > 0$ and let $\mathcal{F} := \{f : \mathbb{R}^d \to \mathbb{R}\}$ be a class of functions. A finite collection $\{f_1, \ldots, f_N \in \mathcal{F}\}$ is called an ε-cover of $\mathcal{F}$ with respect to the norm $\|\cdot\|_p$ if for every $f \in \mathcal{F}$, there exists a

$j \in \{1, \ldots, N\}$ such that

$$\|f - f_j\|_p < \varepsilon.$$

The cardinality of the smallest ε-cover of $\mathcal{F}$ with respect to $\|\cdot\|_p$ is denoted by $\mathcal{N}(\varepsilon, \mathcal{F}, \|\cdot\|_p)$. This quantity is known as the ε-covering number of $\mathcal{F}$ with respect to $\|\cdot\|_p$, and is abbreviated as $\mathcal{N}_p(\varepsilon, \mathcal{F})$.

When $\|f - f_j\|_p$ is replaced by its empirical counterpart $\|f - f_j\|_{p,n}$, the empirical ε-covering number is denoted by $\mathcal{N}_p(\varepsilon, \mathcal{F}, \mathcal{D}_n)$, where $\mathcal{D}_n$ represents the sample.

2.2 ReLU Feedforward Neural Networks

We provide a brief review of regression function estimators based on feedforward neural networks with the rectified linear unit (ReLU) activation function, defined as $\sigma(x) := \max(0, x)$. The function class $\mathcal{F}_n$ is specified as $\mathcal{F}_{\mathcal{T},\mathcal{W},\mathcal{U},\mathcal{S},\mathcal{B}}$, representing a class of feedforward neural networks $f_\phi : \mathbb{R}^d \to \mathbb{R}$ with parameters ϕ, depth $\mathcal{T}$, width $\mathcal{W}$, size $\mathcal{S}$, and number of neurons $\mathcal{U}$. Moreover, each network satisfies $\|f_\phi\|_\infty \leq \mathcal{B}$ for some $0 < \mathcal{B} < \infty$. Note that the parameters may be indexed by n, and we omit n for simplicity.

To be specific, let $\mathcal{T}$ denote the depth of the hidden layers, and the total number of layers is $\mathcal{T} + 2$. The width of each layer is represented by the $(\mathcal{T} + 2)$-vector $(p_0, \ldots, p_{\mathcal{T}+1})^\top$, where $\mathcal{W}$ is defined as $\mathcal{W} := \max(p_1, \ldots, p_{\mathcal{T}})$. The total number of neurons $\mathcal{U}$ is defined as $\mathcal{U} = \sum_{i=1}^{\mathcal{T}} p_i$. Additionally, $\mathcal{S}$ represents the total number of computational units or parameters, satisfying $\mathcal{S} = \sum_{i=0}^{\mathcal{T}} \{p_{i+1}(p_i + 1)\} = O(\mathcal{W}^2\mathcal{T})$.

3. Method

In this section, we first discuss conditional density estimation from a regression perspective and provide a review of some classical kernel estimators in Section 3.1. Then, we present our approach in Section 3.2.

3.1 A Regression View of Conditional Density Estimation

We recast the conditional density estimation as a regression problem and formulate the response using kernel functions. We list some basic properties of the kernel functions in Assumption 1.

Assumption 1. We require the kernel function $K : \mathbb{R} \to \mathbb{R}$ to be a symmetric probability density function around zero satisfying

(i) $\int_{\mathbb{R}} K(u)\mathrm{d}u = 1$,

(ii) $\int_{\mathbb{R}} K^2(u)\mathrm{d}u < \infty$,

(iii) $K(u)$ decays exponentially as $u \to \infty$, i.e., $K(u)/\exp(-|u|)$ is bounded for all $u \in \mathbb{R}$.

In addition, the order of a kernel is defined as the first non-zero moment of the kernel (Li and Racine, 2006). A kernel is said to be an l-order kernel if for all $m = 0, \ldots, \lceil l \rceil - 1$ and for l,

(iv) $\int_{\mathbb{R}} u^m K(u)\mathrm{d}u = 0$,

(v) $\int_{\mathbb{R}} u^l K(u)\mathrm{d}u = \kappa_l < \infty$.

Remark 1. Assumption 1-(i), (ii) are standard requirements for kernel functions. Assumption 1-(iii) is relatively mild, as it is satisfied by many commonly used kernels, including

bounded kernels, the Gaussian kernel, the logistic kernel, and the sigmoid kernel (Li and Racine, 2006). Assumption 1-(iv), (v) are related to the order of kernels. Some well-known kernels such as the Gaussian kernel are second-order kernels. According to Li and Racine (2006), the literature also includes kernels of various other orders.

Assumption 1 lists some properties of kernel functions that facilitate the approximation of the conditional density using the kernel functions. In addition, we require an assumption regarding the smoothness of the conditional density function $f_*(y \mid \mathbf{x})$. We consider the following Hölder class, which is widely used in nonparametric statistics (Györfi et al. 2002, Schmidt-Hieber 2020, Jiao et al. 2023).

Assumption 2. Let $\mathbf{z} = (\mathbf{x}, y) \in \mathbb{R}^{d+1}$. We assume that the true conditional density function $f_*(y \mid \mathbf{x})$ belongs to the Hölder class $\mathcal{H}^\beta([0,1]^{d+1}, \mathcal{B})$, where $\beta = s + r > 0$, $r \in (0, 1]$, $s = \lceil \beta \rceil - 1 \in \mathbb{N}$, and

$$\begin{aligned} \mathcal{H}^\beta([0,1]^{d+1}, \mathcal{B}) = \Big\{ f : [0,1]^{d+1} \to \mathbb{R}, \max_{\|\boldsymbol{\alpha}\|_1 \le s} \|\partial^{\boldsymbol{\alpha}} f\|_\infty \le \mathcal{B}, \\ \max_{\|\boldsymbol{\alpha}\|_1 = s} \sup_{\mathbf{z}_1 \neq \mathbf{z}_2} |\partial^{\boldsymbol{\alpha}} f(y_1 \mid \mathbf{x}_1) - \partial^{\boldsymbol{\alpha}} f(y_2 \mid \mathbf{x}_2)| \le \mathcal{B} \|\mathbf{z}_1 - \mathbf{z}_2\|_2^r \Big\}, \end{aligned} \tag{3.1}$$

where $\mathcal{B}$ is a positive constant, $\partial^{\boldsymbol{\alpha}} = \partial^{\alpha_1} \cdots \partial^{\alpha_{d+1}}$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{d+1})^\top \in \mathbb{N}^{d+1}$, and $\|\boldsymbol{\alpha}\|_1 = \sum_{i=1}^{d+1} \alpha_i$.

We utilize the same constant $\mathcal{B}$ to uniformly bound both the Hölder class and the neural network class. Under Assumption 2, all partial derivatives of f_* up to the s-th order exist. Specifically, f_* is a Lipschitz function when $\beta = 1$. The Hölder class is a commonly used function class in nonparametric regression and density estimation (Stone 1982, Györfi et al. 2002). Given Assumptions 1 and 2, we can derive Lemma 1, which shows that the condi-

tional density estimation can be approximated by solving a regression problem using kernel functions.

Lemma 1. *Let $K : \mathbb{R} \to \mathbb{R}$ be a kernel function of order β that satisfies Assumption 1. Assume that Assumption 2 is satisfied. Given $y \in \mathbb{S}_Y$,*

$$\mathbb{E}_Y \{K_b(Y - y) \mid \mathbf{X} = \mathbf{x}\} = f_*(y \mid \mathbf{x}) + O(b^\beta),$$

where b denotes the bandwidth and $K_b(Y - y) = K\{(Y - y)/b\}/b$.

The detailed proof of Lemma 1 is given in Supplementary Material S2. In the literature, Fan et al. (1996) and Hyndman and Yao (2002) assumed that the conditional density function belongs to Hölder class with $\beta = 2$. Lemma 1 illustrates that the estimation of $f_*(y \mid \mathbf{x})$ can be viewed as a nonparametric regression of $K_b(Y - y)$ on $\mathbf{X}$. Given $y \in \mathbb{S}_Y$, we consider the least squares problem

$$f_0(y \mid \mathbf{x}) = \underset{f(y|\mathbf{x})}{\arg\min} \|K_b(Y - y) - f(y \mid \mathbf{x})\|_2^2. \tag{3.2}$$

At the population level and by the property of the minimization of L_2 risk, we have

$$\mathbb{E}_Y\{K_b(Y - y) \mid \mathbf{X} = \mathbf{x}\} = f_0(y \mid \mathbf{x}) = f_*(y \mid \mathbf{x}) + O(b^\beta),$$

where $f_*(y \mid \mathbf{x})$ is the true underlying conditional density. Thus, it suffices to establish the regression procedure to learn the conditional density function f_*.

Let us consider equation (3.2) at the empirical level. To be specific, for a sample $\mathcal{D}_n$ and a given value y, we can regress $K_b(Y_i - y)$ on $\mathbf{X}_i$ to get $\hat{f}_n(y \mid \mathbf{x})$. Classical estimation methods

aim to estimate $f_0(y \mid \mathbf{x})$ at a pre-specified point $(\mathbf{x}, y)$. For example, the Nadaraya–Watson estimator (NW estimator, Hyndman et al. 1996; De Gooijer and Zerom 2003) is defined as

$$\hat{f}_{NW}(y \mid \mathbf{x}) = \arg\min_{a} \sum_{i=1}^{n} \left\{ K_{b_1}(Y_i - y) - a \right\}^2 K_{b_2}(\mathbf{X}_i - \mathbf{x}), \tag{3.3}$$

where $K_{b_2}(\mathbf{X}_i - \mathbf{x})$ represents the weights to the sample near $\mathbf{x}$. As an extension, Fan et al. (1996) proposed the local linear estimator

$$\hat{f}_{LL}(y \mid \mathbf{x}) = \arg\min_{a} \sum_{i=1}^{n} \left\{ K_{b_1}(Y_i - y) - a - \boldsymbol{b}^\top (\mathbf{X}_i - \mathbf{x}) \right\}^2 K_{b_2}(\mathbf{X}_i - \mathbf{x}). \tag{3.4}$$

Compared with $\hat{f}_{NW}(y \mid \mathbf{x})$, $\hat{f}_{LL}(y \mid \mathbf{x})$ exhibits improved bias performance under some conditions according to De Gooijer and Zerom (2003). However, the estimator $\hat{f}_{LL}(y \mid \mathbf{x})$ has certain aspects that might benefit from further consideration:

1. $\hat{f}_{LL}(y \mid \mathbf{x})$ is a local estimator for a given pair $(\mathbf{x}, y)$, rather than providing a global estimate for an arbitrary $\mathbf{x}$.

2. When $\mathbf{x}$ is multi-dimensional, particularly in high-dimensional settings, the estimator $\hat{f}_{LL}(y \mid \mathbf{x})$ is computationally expensive. This complexity arises due to a substantial increase in the computation required for the kernel $K_{b_2}(\mathbf{X}_i - \mathbf{x})$ and the number of pre-set points for $\mathbf{x}$ as its dimensionality grows.

3.2 Problem Formulation

Following the regression view of conditional density estimation, we focus on the least squares problem (3.2) and its empirical counterpart. We first define the L_2 loss at $y \in \mathbb{R}$ as

$$L(f, y) = \{K_b(Y - y) - f(y \mid \mathbf{x})\}^2.$$

At the population level, the L_2 risk can be written as

$$\mathbb{E}_{\mathbf{Z}}\{L(f, y)\} = \mathbb{E}_{\mathbf{Z}}\{K_b(Y - y) - f(y \mid \mathbf{x})\}^2,$$

and its minimizer is given in (3.2). For a sample $\mathcal{D}_n$, we define the empirical L_2 risk and its minimizer by

$$\mathbb{E}_n\{L(f, y)\} = \frac{1}{n}\sum_{i=1}^{n}\{K_b(Y_i - y) - f(y \mid \mathbf{X}_i)\}^2,$$

$$\hat{f}_n(y \mid \mathbf{x}) = \arg\min_{f \in \mathcal{F}_n} \mathbb{E}_n\{L(f, y)\}, \tag{3.5}$$

where $\mathcal{F}_n$ denotes the neural network function class and $\hat{f}_n$ is referred to as the empirical risk minimizer. We further define the excess risk as the L_2-distance between the estimated conditional density and the true underlying density

$$\mathcal{R}(\hat{f}_n, f_*) = \mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2. \tag{3.6}$$

The excess risk is an important and widely employed criterion for evaluating the performance of $\hat{f}_n$ (Schmidt-Hieber 2020, Jiao et al. 2023). In this context, we explore how to properly

specify the bandwidth and the neural network structure to achieve the minimax optimal convergence of the excess risk.

Remark 2. Unlike the standard nonparametric regression formulation, a conditional density $f(y \mid \mathbf{x})$ must satisfy two constraints: non-negativity and the normalization condition $\int f(y \mid \mathbf{x})\,\mathrm{d}y = 1$. In our framework, non-negativity can be enforced by using a ReLU activation in the output layer. However, because our estimator is trained pointwise in y, the normalization constraint across y is not imposed directly. To alleviate this issue, we develop an extension in Section 5 that treats $(\mathbf{x}, y)$ jointly as inputs and apply a post-normalization procedure to ensure $\int \hat{f}(y \mid \mathbf{x})\,\mathrm{d}y = 1$.

4. Convergence Analysis

In this section, we conduct a convergence analysis of our estimator. We start by proving an oracle inequality, which decomposes the overall error into the stochastic error and the approximation error. The stochastic error is analyzed in Section 4.2. We then balance these error components and derive the total error bounds in Section 4.3. Finally, in Section 4.4, we explore the adaptation of the estimator to high-dimensional settings under the low-dimensional manifold assumption.

4.1 Oracle Inequality for Error Decomposition

Recall the excess risk $\mathcal{R}(\hat{f}_n, f_*)$ defined in (3.6). To derive the error bounds, we decompose the risk into several parts and tackle each part individually. To be specific, we consider two types of errors, the stochastic error arising from the estimation based on a random sample, and the approximation error arising from the neural network function class (Farrell et al.

2021, Jiao et al. 2023). We present the following oracle inequality.

Lemma 2. *For a random sample $\mathcal{D}_n$ and a specified $y \in \mathbb{R}$, let*

$$\varepsilon_i := K_b(Y_i - y) - f_*(y \mid \mathbf{X}_i), \tag{4.1}$$

$$f_n \in \arg\min_{f \in \mathcal{F}_n} \|f - f_*\|^2. \tag{4.2}$$

The excess risk $\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2$ satisfies

$$\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2 \le T_1 + 4T_2 + 2T_3,$$

where $T_1 := \mathbb{E}_{\mathcal{D}_n}(\|\hat{f}_n - f_*\|_2^2 - 2\|\hat{f}_n - f_*\|_{2,n}^2)$, $T_2 := \mathbb{E}_{\mathcal{D}_n}[\mathbb{E}_n\{\varepsilon_i(\hat{f}_n - f_n)\}]$, and $T_3 := \|f_n - f_*\|_2^2$.

In Lemma 2, the terms T_1 and T_2 are stochastic errors, while T_3 denotes the approximation error. f_n in (4.2) is defined as a minimizer of the population L_2 risk over the function class $\mathcal{F}_n$.

The proof of Lemma 2 can be found in Supplementary Material S3. In contrast to a standard regression problem (Jiao et al., 2023), an additional term T_2 arises in Lemma 2 due to the approximation of $\mathbb{E}\{K_b(Y - y) \mid \mathbf{X} = \mathbf{x}\} \approx f_*(y \mid \mathbf{x})$. Consequently, the term T_2 requires additional consideration. Furthermore, the ε_i in (4.1) does not behave "regularly". The bandwidth tends to 0 as the sample size n increases resulting in the unbounded $K_b(Y_i - y) = K(Y_i - y)/b$. Thus, the challenges in bounding $\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2$ arise in two aspects. Firstly, ε_i is not centered, i.e., $\mathbb{E}(\varepsilon_i \mid \mathbf{X}_i) \neq 0$, due to the kernel approximation as shown in Lemma 1. Secondly, the infinity norm of the response $K_b(Y_i - y) = K\{(Y_i - y)/b\}/b$ increases at the same order of $1/b$, where b is expected to converge to 0 as the sample size n

increases. Additionally, if $1/b$ increases at the polynomial order of n, the empirical process bounds for a bounded response (Györfi et al., 2002) or a sub-exponential response (Jiao et al., 2023) are no longer effective. We try to tackle these difficulties in Section 4.2. In addition, we discuss the approximation error in Lemma 5 and we recommend Jiao et al. (2023) for comprehensive studies for the analysis of approximation error with deep neural networks.

4.2 Stochastic Error

To bound the stochastic error terms T_1 and T_2, we need to use techniques from the empirical process theory. The following Lemma 3 gives an error bound of the term T_1.

Lemma 3. *Assume that Assumptions 1, 2 are satisfied. Given* $\|\hat{f}_n\|_\infty \leq \mathcal{B}$, $\|f_*\|_\infty \leq \mathcal{B}$, $\mathcal{B} > 1$, *and* $\sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n) \geq 3$, *the term* T_1 *in Lemma 2 can be bounded as*

$$\mathbb{E}_{\mathcal{D}_n}\left(\|\hat{f}_n - f_*\|_2^2 - 2\|\hat{f}_n - f_*\|_{2,n}^2\right) \lesssim \frac{1}{n}\log \sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n).$$

Lemma 3 is a standard result in the empirical process theory. We omit some constants related with $\mathcal{B}$ for simplicity. The detailed proof of Lemma 3 is deferred to Supplementary Material S4. By Lemma 3, we can infer that the term T_1 is bounded by a term determined by the covering numbers. Subsequently, we deal with the stochastic error T_2. We truncate ε_i defined in (4.1) and derive a small probability of extreme values, which leads us to Lemma 4.

Lemma 4. *Assume that Assumptions 1, 2 are satisfied. The term* T_2 *in Lemma 2 can be*

bounded by

$$\begin{aligned}\mathbb{E}_{\mathcal{D}_n}\left[\mathbb{E}_n\{\varepsilon_i(\hat{f}_n - f_n)\}\right] \lesssim\ & b^{2\beta} + \log(1/b)b + \|f_n - f_*\|^2/4 \\ & + \mathbb{E}_{\mathcal{D}_n}\left\{\mathbb{E}_n(\hat{f}_n - f_*)^2\right\}/8 + \frac{\log \sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n)}{n} \\ & + \left(\frac{\log \sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n)}{n}\right)^{1/2} \left(\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2\right)^{1/2}.\end{aligned}$$

The proof of Lemma 4 is deferred to Supplementary Material S5. In Lemma 4, there are additional terms $b^{2\beta}$ and $b\log(1/b)$ induced by the approximation of kernel functions. Combining the error bounds of these two stochastic errors, we can conclude the following Theorem 1.

Theorem 1. *Assume that Assumptions 1–2 are satisfied. Given $y \in \mathbb{S}_Y$, and the kernel bandwidth b, the excess risk $\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2$ can be bounded by*

$$\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2 \lesssim \frac{1}{n}\log \sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n) + b^{2\beta} + b\log(1/b) + \|f_n - f_*\|_2^2,$$

where the last term is the approximation error of the neural network function class.

According to Bartlett et al. (2019), the covering number $\sup_{\mathcal{D}_n} \mathcal{N}_\infty(1/n, \mathcal{F}_n, \mathcal{D}_n)$ in Theorem 1 can be further bounded by the size and the depth of the neural network function class with

$$\log\left\{\sup \mathcal{N}_\infty\left(1/n, \mathcal{F}_n, \mathcal{D}_n\right)\right\} \lesssim \mathcal{S}\mathcal{T}\log(\mathcal{S})\log n,$$

which implies Corollary 1.

Corollary 1. *Assume that Assumptions 1–2 are satisfied. Choose the bandwidth as* $b = n^{-1/(2\beta+d)} \wedge n^{-2\beta/(2\beta+d)}$. *The excess risk can be bounded by*

$$\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2 \lesssim \frac{1}{n}\mathcal{S}\mathcal{T}\log(\mathcal{S})\log(n) + n^{-2\beta/(2\beta+d)}\log n + \|f_n - f_*\|_2^2, \tag{4.3}$$

where the last term denotes the approximation error by the neural network function class.

Corollary 1 is derived from Theorem 1 by substituting the covering number bound with the structure parameters of neural network function class. Please refer to Supplementary Material S7 for a detailed discussion.

4.3 Overall Error Bounds

In this section, we first bound the approximation error term $\|f_n - f_*\|_2^2$ and make a balance between the approximation error and the stochastic error. Our approximation error bound is built on Theorem 3.3 in Jiao et al. (2023).

Lemma 5. *(Jiao et al., 2023, Theorem 3.3) Suppose that Assumption 2 is satisfied. For any* $M, N \in \mathbb{N}^+$, *there exists a function* ϕ_0 *implemented by a deep ReLU network with width* $\mathcal{W} = 38\lceil\beta\rceil^2 d^{\lceil\beta\rceil} N\lceil\log_2(8N)\rceil$ *and depth* $\mathcal{T} = 21\lceil\beta\rceil^2 M\lceil\log_2(8M)\rceil$ *such that*

$$|f(\mathbf{x}) - \phi_0(\mathbf{x})| \leq 18\mathcal{B}\lceil\beta\rceil^2 d^{\lceil\beta\rceil - 1 + (\beta\vee 1)/2)}(NM)^{-2\beta/d},$$

for all $\mathbf{x} \in [0,1]^d \backslash \Omega([0,1]^d, K, \delta)$, *and*

$$\Omega([0,1]^d, K, \delta) = \bigcup_{i=1}^{d}\left\{x = [x_1, \ldots, x_d]^\top : x_i \in \cup_{k=1}^{K-1}(k/K - \delta, k/K)\right\},$$

where $K = \lceil (MN)^{2/d} \rceil$ *and* δ *is an arbitrary number in* $(0, 1/(3K)]$.

Based on Lemma 5, for a deep ReLU network with the width $\mathcal{W} = 38\lceil\beta\rceil^2 d^{\lceil\beta\rceil} N\lceil\log_2(8N)\rceil$ and the depth $\mathcal{T} = 21\lceil\beta\rceil^2 M\lceil\log_2(8M)\rceil$, the approximation error $\|f_n - f_*\|_2^2$ converges to zero at the order $(NM)^{-4\beta/d}$. Consequently, the excess risk satisfies

$$\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n - f_*\|_2^2 \lesssim \frac{1}{n}\mathcal{S}\mathcal{T}\log(\mathcal{S})\log(n) + n^{-2\beta/(2\beta+d)}\log n + (NM)^{-4\beta/d}.$$

We can derive the following Corollary 2 regarding the consistency and convergence rate of the conditional density estimator.

Corollary 2. *(Consistency and convergence rates)*

1. *For a deep ReLU network class* $\mathcal{F}_n$*, if*

$$\mathcal{S} \to \infty, \text{ and } \frac{1}{n}\mathcal{S}\mathcal{T}\log(\mathcal{S})\log(n) \to 0 \text{ as } n \to \infty,$$

the estimator $\hat{f}_n$ *in* (3.5) *is consistent with its excess risk converging to zero. Here* $\mathcal{S}$ *and* $\mathcal{T}$ *represent the size and the depth of the neural network function class* $\mathcal{F}_n$.

2. *The estimator achieves the minimax optimal rate up to some* $\log n$ *factors if* M *and* N *are properly specified. For example, we set* $N = O(1)$ *and* $M \asymp n^{d/(4\beta+2d)}$*, which can be viewed as a depth-increasing regime (Vardi et al., 2022). Then we obtain*

$$\mathbb{E}_{\mathcal{D}_n}\|\hat{f}_n(y \mid \mathbf{x}) - f_*(y \mid \mathbf{x})\|_2^2 \lesssim n^{-2\beta/(2\beta+d)}\log^4(n). \tag{4.4}$$

The proof of Corollary 2 is given in Supplementary Material S8.

Remark 3. The minimax optimal convergence rate for nonparametric regression and density estimation within the Hölder class is known to be $n^{-2\beta/(2\beta+d)}$ (Stone, 1982). For conditional density estimation involving a scalar response Y and d-dimensional covariate $\mathbf{X}$, the minimax optimal rate is $n^{-2\beta/(2\beta+d+1)}$ (Li et al., 2022). In our approach, we fix a point y and estimate the conditional density over the covariates $\mathbf{X}$. Consequently, the minimax optimal rate under our setting reverts to $n^{-2\beta/(2\beta+d)}$. We establish that our estimator attains this rate, up to logarithmic factors.

4.4 Adaptation into Low-dimensional Manifold Assumptions

In our approach, the key distinction from standard regression lies in the treatment of stochastic error. Consequently, our approach can also utilize the remarkable approximation ability of deep neural networks in the low-dimensional manifold assumption. To illustrate this point, we adopt the low-dimensional manifold assumption as discussed in Jiao et al. (2023).

Assumption 3. The predictor $\mathbf{X}$ for $f_*(y \mid \mathbf{x})$ is supported on $\mathcal{M} \subset [0,1]^d$, where the $\mathcal{M}$ is a compact $d_{\mathcal{M}}$-dimensional Riemannian manifold isometrically embedded in $\mathbb{S}_{\mathbf{X}}$. Typically, $d_{\mathcal{M}}$ is significantly smaller than d.

Corollary 3. *(Error bounds in low-dimensional manifold settings) Assume that Assumptions 1–3 are satisfied. Let the bandwidth be chosen as* $b = n^{-1/(2\beta+d_{\mathcal{M}})} \wedge n^{-2\beta/(2\beta+d_{\mathcal{M}})}$. *There exists a neural network function class* $\mathcal{F}_n$, *such that the excess risk of the empirical risk minimizer satisfies*

$$\mathbb{E}_{\mathcal{D}_n} \|\hat{f}_n - f_*\|_2^2 \lesssim n^{-2\beta/(2\beta+d_{\mathcal{M}})} \log^4(n).$$

Remark 4. In Corollary 2, we show that the convergence rate is of the order $n^{-2\beta/(2\beta+d)}$, up to some logarithmic factors. By imposing Assumption 3, we further demonstrate that the rate can be improved to $n^{-2\beta/(2\beta+d_{\mathcal{M}})}$, where the dimensionality is reduced from d to the intrinsic dimension of the manifold $d_{\mathcal{M}}$. This result provides insights into the effectiveness of our approach in mitigating the curse of dimensionality.

5. Tackling the Varying Response Case

In the previous sections, we focused on estimating the conditional density function by fixing y and varying $\mathbf{x}$. While this approach is theoretically rigorous, it can be computationally burdensome in practice, as it necessitates a separate training process for each query value of y. In addition, the property that the conditional density integrates to one is not guaranteed. To address this limitation, we introduce an extension that enables simultaneous estimation for all values of y. Specifically, we expand the input space of the neural network to include the response variable y, thereby learning a mapping from $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$ to the conditional density values $f(y \mid \mathbf{x})$. To train this global model, we employ a data augmentation strategy. For each observation $\mathbf{Z}_i = (\mathbf{X}_i, Y_i)$ in the sample, we pair it with every observed response Y_j with $j = 1, \ldots, n$ from the dataset, yielding a total of n^2 augmented data points $\mathcal{D}_{n,\text{aug}} = \{(\mathbf{Z}_i, Y_j) : i = 1, \ldots, n, j = 1, \ldots, n\}$. The loss function is defined as

$$L_{n,\text{aug}}(f) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \{K_b(Y_i - Y_j) - f(Y_j \mid \mathbf{X}_i)\}^2.$$

Consequently, the estimated conditional density function is obtained by

$$\hat{f}_{n,\text{aug}} = \arg\min_{f \in \mathcal{F}_n} L_{n,\text{aug}}(f). \tag{5.1}$$

By minimizing the aggregated squared error loss over the n^2 augmented data points, $\hat{f}_{n,\text{aug}}$ provides a global approximation of the conditional density function. However, a potential drawback is that $\hat{f}_{n,\text{aug}}(y \mid \mathbf{x})$ does not necessarily integrate to one for a given $\mathbf{x}$. To address this issue, we employ a normalization procedure that is utilized by Han et al. (2025). Specifically, we construct a normalized estimator

$$\tilde{f}_{n,\text{aug}}(y \mid \mathbf{x}) = \frac{\hat{f}_{n,\text{aug}}(y \mid \mathbf{x})}{\int \hat{f}_{n,\text{aug}}(y \mid \mathbf{x})\, \mathrm{d}y}. \tag{5.2}$$

In practice, the integral in the denominator can be approximated using numerical integration, and $\tilde{f}_{n,\text{aug}}(y \mid \mathbf{x})$ can then be evaluated as a function of y for a fixed value of $\mathbf{x}$. The theoretical results developed earlier for the fixed-y estimator do not directly extend to $\tilde{f}_{n,\text{aug}}$ in (5.2). The sample augmentation contains dependence because the same observations are reused across the n^2 pairs, and the normalization step introduces a random denominator that depends on the estimated function over the response domain. For these reasons, extending the theory to the normalized varying-response estimator is nontrivial and is left for future work.

We provide simulations for the fixed $\mathbf{x}$ and varying y case in Section 6.1 (see Equation (6.2) and Table 1), as well as comparative simulation results between the unnormalized and normalized estimators in Supplementary Material S1 (see Table S1).

6. Numerical Studies

In this section, we first present some simulated data in Section 6.1. We then apply our method to analyze a photometric redshift dataset and a typhoon center pressure dataset in Section 6.2.

6.1 Simulated Data

We evaluate the performance of our proposed method using simulated data. We compare our method with three competitors: the Nadaraya–Watson (NW) estimator (Rosenblatt, 1956), the local linear (LL) estimator (Fan et al., 1996), and the kernel mixture network (KMN) (Ambrogioni et al., 2017). The KMN is a modern neural network-based method for conditional density estimation that represents the conditional density as a mixture of kernel functions with parameters learned through deep neural networks. We compare these four methods in terms of estimated excess risk under two evaluation protocols: the fixed-y protocol using estimator (3.5) and the fixed-$\mathbf{x}$ protocol using the global estimator (5.2).

The fixed-y protocol evaluates performance at a fixed response value y across varying covariate values, using estimator $\hat{f}_n$ from (3.5). We train $\hat{f}_n$ on the sample $\mathcal{D}_n$ and evaluate it on an independent test set $\{\mathbf{X}'_j,\, j = 1, \ldots, m\}$ drawn from the marginal distribution of $\mathbf{X}$. The estimated excess risk is

$$\frac{1}{m}\sum_{j=1}^{m}\left\{\hat{f}_n(y \mid \mathbf{X}'_j) - f_*(y \mid \mathbf{X}'_j)\right\}^2 f_*(y \mid \mathbf{X}'_j). \tag{6.1}$$

The fixed-$\mathbf{x}$ protocol evaluates performance at a fixed covariate value $\mathbf{x}$ across varying response values, using the global estimator in (5.2). For a fixed $\mathbf{x}$, we draw i.i.d. test

responses $\{Y'_j : j = 1, \ldots, m\}$ according to $f_*(\cdot \mid \mathbf{x})$ and compute

$$\frac{1}{m}\sum_{j=1}^{m}\left\{\hat{f}(Y'_j \mid \mathbf{x}) - f_*(Y'_j \mid \mathbf{x})\right\}^2 f_*(Y'_j \mid \mathbf{x}), \tag{6.2}$$

where $\hat{f}$ is the normalized estimator $\tilde{f}_{n,\text{aug}}$ from (5.2). We also compare the excess risk of (5.2) with that of its unnormalized counterpart $\hat{f}_{n,\text{aug}}$ in (5.1). See Supplementary Material S1.

We report estimated excess risk and standard deviation for both metrics with sample sizes $n = 500, 1000, 2000$. For the fixed-y protocol (6.1), we evaluate at $y = a$ for $a = -1, 0, 1, 2$. For the fixed-$\mathbf{x}$ protocol (6.2), we evaluate at $\mathbf{x} = (a, a, \ldots, a)^\top$ for $a = -0.3, 0, 0.3$. The test set size is $m = 200$ for both protocols.

For bandwidth selection, we follow the recommendations in the original papers for the NW and LL estimators (Rosenblatt, 1956; Fan et al., 1996). For our NN method, we set the bandwidth proportional to $n^{-2/(2+d)}$ following the guidance in Corollary 1. For the neural network-based methods (our NN method and KMN), we use feedforward neural networks with hidden-layer widths $(128, 64, 32, 16)$ and train them in PyTorch using the Adam optimizer. For KMN, we build on the implementation provided by Rothfuss et al. (2019).

We consider two types of errors when generating the data: additive error (see Model 1) and multiplicative error (see Model 2). Within each error model, we evaluate performance in both low-dimensional and high-dimensional settings.

Model 1 (Additive error)**.** We generate $\mathbf{X} = (X_1, \ldots, X_d)^\top \in \mathbb{R}^d$, where each $X_i \sim U(-1, 1)$, and error $\varepsilon \sim N_{[-5,5]}(0, 1)$.

Setting 1.1 (Low-dimensional $\mathbf{X}$). We set $d = 3$ and generate the response as $Y = X_1^2 + \sin(X_2) + \varepsilon$.

Setting 1.2 (High-dimensional $\mathbf{X}$). We set $d = 300$ and generate the response as $Y = X_1^2 + \sin(X_2) + \exp(X_3) + \varepsilon$.

Model 2 (Multiplicative error)**.** We generate $\mathbf{X} = (X_1, \ldots, X_d)^\top \in \mathbb{R}^d$, where each $X_i \sim U(-1, 1)$, and error $\varepsilon \sim N_{[-5,5]}(0, 1)$.

Setting 2.1 (Low-dimensional $\mathbf{X}$). We set $d = 3$ and generate the response as $Y = \exp(X_1/2) \cdot (X_2 + 2) \cdot \cos(X_3) \cdot \varepsilon$.

Setting 2.2 (High-dimensional $\mathbf{X}$). We set $d = 300$ and generate the response as $Y = \exp(X_1/2) \cdot (|X_2| + 2) \cdot \{|\sin(X_3)| + 1\} \cdot \varepsilon$.

The estimated excess risk and standard deviation for Model 1 and Model 2 are reported in Table 1, which contains four sub-tables corresponding to each combination of error type and dimensionality level. Computation times for all methods are reported in Table 2. An interesting pattern in Table 2 is that KMN is relatively insensitive to n over the sample sizes considered, whereas our NN method exhibits a clearer increase. This is because KMN fits a single network with a fixed mixture structure, making the marginal effect of increasing n relatively small in this range. By contrast, the varying-response NN estimator in (5.2) is trained on the augmented sample and subsequently normalized over y, so its computational burden grows with the sample size.

We first examine how the performance of all methods varies with the dimensionality of the covariate space. In the low-dimensional settings (Table 1a and Table 1c), all four methods perform comparably well in terms of estimated excess risk. For the fixed-y evaluation protocol, the methods alternate in achieving the best performance depending on the specific y value and sample size. KMN exhibits larger excess risk when the sample size is small (e.g.,

Table 1: Simulation results for Model 1 and Model 2. "NN" refers to our approach, "NW" denotes the Nadaraya–Watson estimator, "LL" denotes the local linear estimator, and "KMN" denotes the kernel mixture network. Standard deviations are shown as subscripts. All values are multiplied by 1000 for *Setting* 1.1 and *Setting* 2.1, and by 100 for *Setting* 1.2 and *Setting* 2.2, except for a and n. All results are based on 200 replications.

(a) *Setting* 1.1

		Fixed y			
a	n	NN	NW	LL	KMN
-1	500	$0.65_{(0.33)}$	$0.67_{(0.20)}$	$\mathbf{0.64_{(0.18)}}$	$2.79_{(1.92)}$
	1000	$\mathbf{0.42_{(0.23)}}$	$0.52_{(0.15)}$	$0.59_{(0.14)}$	$0.97_{(0.60)}$
	2000	$\mathbf{0.29_{(0.15)}}$	$0.39_{(0.12)}$	$0.58_{(0.14)}$	$0.33_{(0.22)}$
0	500	$1.75_{(0.68)}$	$\mathbf{1.66_{(0.37)}}$	$2.40_{(0.28)}$	$7.32_{(5.68)}$
	1000	$\mathbf{1.22_{(0.43)}}$	$1.23_{(0.27)}$	$2.30_{(0.28)}$	$2.28_{(1.61)}$
	2000	$0.87_{(0.31)}$	$0.91_{(0.22)}$	$2.24_{(0.34)}$	$\mathbf{0.73_{(0.40)}}$
1	500	$\mathbf{1.51_{(0.74)}}$	$1.52_{(0.40)}$	$1.64_{(0.21)}$	$6.30_{(4.04)}$
	1000	$\mathbf{1.03_{(0.40)}}$	$1.12_{(0.28)}$	$1.52_{(0.26)}$	$1.86_{(1.01)}$
	2000	$0.72_{(0.32)}$	$0.81_{(0.20)}$	$1.51_{(0.35)}$	$\mathbf{0.61_{(0.30)}}$
2	500	$0.48_{(0.29)}$	$\mathbf{0.47_{(0.18)}}$	$0.68_{(0.24)}$	$2.03_{(1.52)}$
	1000	$\mathbf{0.29_{(0.15)}}$	$0.36_{(0.14)}$	$0.53_{(0.20)}$	$0.66_{(0.42)}$
	2000	$\mathbf{0.21_{(0.12)}}$	$0.30_{(0.11)}$	$0.42_{(0.14)}$	$0.26_{(0.17)}$
		Fixed $\mathbf{x}$			
a	n	NN	NW	LL	KMN
-0.3	500	$\mathbf{0.53_{(0.65)}}$	$1.17_{(0.60)}$	$2.28_{(0.56)}$	$3.68_{(3.59)}$
	1000	$\mathbf{0.40_{(0.57)}}$	$0.86_{(0.37)}$	$2.29_{(0.54)}$	$1.28_{(1.40)}$
	2000	$\mathbf{0.35_{(0.36)}}$	$0.56_{(0.39)}$	$2.22_{(0.49)}$	$0.36_{(0.34)}$
0	500	$\mathbf{0.47_{(0.56)}}$	$1.12_{(0.35)}$	$2.40_{(0.45)}$	$4.61_{(7.26)}$
	1000	$\mathbf{0.38_{(0.49)}}$	$0.85_{(0.31)}$	$2.34_{(0.44)}$	$1.19_{(1.17)}$
	2000	$\mathbf{0.30_{(0.34)}}$	$0.57_{(0.27)}$	$2.27_{(0.46)}$	$0.38_{(0.36)}$
0.3	500	$\mathbf{0.45_{(0.58)}}$	$1.08_{(0.37)}$	$2.25_{(0.60)}$	$3.95_{(4.38)}$
	1000	$\mathbf{0.36_{(0.47)}}$	$0.79_{(0.35)}$	$2.19_{(0.55)}$	$1.08_{(1.42)}$
	2000	$\mathbf{0.32_{(0.42)}}$	$0.61_{(0.32)}$	$2.13_{(0.55)}$	$0.35_{(0.40)}$

(b) *Setting* 1.2

Fixed y			
NN	NW	LL	KMN
$\mathbf{0.08_{(0.02)}}$	$0.14_{(0.02)}$	$0.11_{(0.02)}$	$0.26_{(0.16)}$
$\mathbf{0.07_{(0.01)}}$	$0.14_{(0.02)}$	$0.11_{(0.02)}$	$0.26_{(0.17)}$
$\mathbf{0.06_{(0.01)}}$	$0.14_{(0.02)}$	$0.11_{(0.02)}$	$0.24_{(0.17)}$
$\mathbf{0.42_{(0.06)}}$	$0.74_{(0.07)}$	$1.20_{(0.10)}$	$1.99_{(0.86)}$
$\mathbf{0.37_{(0.05)}}$	$0.77_{(0.08)}$	$1.18_{(0.08)}$	$1.92_{(0.88)}$
$\mathbf{0.31_{(0.04)}}$	$0.68_{(0.07)}$	$1.04_{(0.09)}$	$1.74_{(0.82)}$
$\mathbf{0.55_{(0.07)}}$	$1.22_{(0.11)}$	$2.71_{(0.11)}$	$5.14_{(2.13)}$
$\mathbf{0.54_{(0.06)}}$	$1.16_{(0.09)}$	$2.65_{(0.11)}$	$5.12_{(2.45)}$
$\mathbf{0.51_{(0.06)}}$	$1.07_{(0.09)}$	$2.60_{(0.07)}$	$4.97_{(2.47)}$
$\mathbf{0.55_{(0.07)}}$	$1.19_{(0.10)}$	$2.47_{(0.11)}$	$4.92_{(1.84)}$
$\mathbf{0.54_{(0.07)}}$	$1.14_{(0.10)}$	$2.43_{(0.10)}$	$4.82_{(2.03)}$
$\mathbf{0.50_{(0.05)}}$	$1.08_{(0.09)}$	$2.41_{(0.07)}$	$4.91_{(2.43)}$
Fixed x			
NN	NW	LL	KMN
$\mathbf{0.76_{(0.74)}}$	$0.97_{(0.80)}$	$2.80_{(0.27)}$	$2.75_{(0.88)}$
$\mathbf{0.73_{(0.68)}}$	$0.85_{(0.77)}$	$2.79_{(0.30)}$	$2.33_{(1.01)}$
$\mathbf{0.71_{(0.61)}}$	$0.80_{(0.58)}$	$2.78_{(0.26)}$	$1.66_{(1.05)}$
$\mathbf{0.79_{(0.73)}}$	$0.90_{(0.71)}$	$2.80_{(0.29)}$	$2.68_{(0.99)}$
$\mathbf{0.79_{(0.69)}}$	$0.82_{(0.69)}$	$2.82_{(0.24)}$	$2.43_{(1.07)}$
$\mathbf{0.72_{(0.61)}}$	$0.78_{(0.57)}$	$2.71_{(0.22)}$	$1.58_{(0.85)}$
$\mathbf{0.78_{(0.71)}}$	$0.97_{(0.84)}$	$2.80_{(0.28)}$	$2.70_{(1.05)}$
$\mathbf{0.78_{(0.68)}}$	$0.95_{(1.03)}$	$2.72_{(0.25)}$	$2.47_{(1.03)}$
$\mathbf{0.67_{(0.60)}}$	$0.86_{(0.56)}$	$2.70_{(0.29)}$	$1.61_{(0.89)}$

(c) *Setting* 2.1

		Fixed y			
a	n	NN	NW	LL	KMN
-1	500	$0.38_{(0.24)}$	$\mathbf{0.28_{(0.09)}}$	$0.90_{(0.22)}$	$1.01_{(0.57)}$
	1000	$0.27_{(0.12)}$	$\mathbf{0.26_{(0.07)}}$	$0.79_{(0.17)}$	$0.38_{(0.21)}$
	2000	$0.20_{(0.08)}$	$0.22_{(0.05)}$	$0.67_{(0.15)}$	$\mathbf{0.17_{(0.07)}}$
0	500	$\mathbf{3.60_{(1.73)}}$	$4.73_{(2.00)}$	$4.09_{(2.05)}$	$4.26_{(2.19)}$
	1000	$\mathbf{2.90_{(1.31)}}$	$3.92_{(1.66)}$	$4.64_{(2.04)}$	$3.73_{(2.32)}$
	2000	$\mathbf{2.13_{(1.07)}}$	$2.83_{(1.15)}$	$4.42_{(1.73)}$	$3.64_{(2.39)}$
1	500	$0.41_{(0.23)}$	$\mathbf{0.29_{(0.08)}}$	$0.90_{(0.23)}$	$1.14_{(0.78)}$
	1000	$0.29_{(0.14)}$	$\mathbf{0.26_{(0.07)}}$	$0.79_{(0.20)}$	$0.37_{(0.22)}$
	2000	$0.19_{(0.07)}$	$0.22_{(0.06)}$	$0.66_{(0.15)}$	$\mathbf{0.18_{(0.09)}}$
2	500	$0.11_{(0.07)}$	$\mathbf{0.09_{(0.03)}}$	$0.17_{(0.06)}$	$0.35_{(0.21)}$
	1000	$\mathbf{0.07_{(0.04)}}$	$0.08_{(0.02)}$	$0.14_{(0.05)}$	$0.13_{(0.06)}$
	2000	$\mathbf{0.05_{(0.03)}}$	$0.06_{(0.02)}$	$0.13_{(0.05)}$	$0.06_{(0.03)}$
		Fixed x			
a	n	NN	NW	LL	KMN
-0.3	500	$\mathbf{0.83_{(0.72)}}$	$3.23_{(2.19)}$	$3.23_{(3.28)}$	$3.29_{(2.19)}$
	1000	$\mathbf{0.48_{(0.40)}}$	$2.82_{(1.79)}$	$2.72_{(2.70)}$	$1.48_{(1.28)}$
	2000	$\mathbf{0.33_{(0.26)}}$	$2.51_{(1.30)}$	$2.29_{(1.76)}$	$0.54_{(0.52)}$
0	500	$\mathbf{0.56_{(0.64)}}$	$2.26_{(1.36)}$	$2.20_{(1.89)}$	$2.86_{(1.91)}$
	1000	$\mathbf{0.27_{(0.25)}}$	$2.01_{(1.04)}$	$1.81_{(1.21)}$	$1.04_{(0.92)}$
	2000	$\mathbf{0.20_{(0.17)}}$	$1.77_{(0.83)}$	$1.57_{(0.99)}$	$0.39_{(0.35)}$
0.3	500	$\mathbf{0.37_{(0.36)}}$	$1.90_{(1.24)}$	$1.87_{(1.77)}$	$1.88_{(1.27)}$
	1000	$\mathbf{0.16_{(0.13)}}$	$1.52_{(0.66)}$	$1.40_{(0.65)}$	$0.91_{(0.84)}$
	2000	$\mathbf{0.12_{(0.10)}}$	$1.38_{(0.59)}$	$1.23_{(0.53)}$	$0.26_{(0.24)}$

(d) *Setting* 2.2

Fixed y			
NN	NW	LL	KMN
$\mathbf{0.11_{(0.01)}}$	$0.29_{(0.04)}$	$0.13_{(0.01)}$	$0.23_{(0.08)}$
$\mathbf{0.10_{(0.02)}}$	$0.27_{(0.04)}$	$0.12_{(0.01)}$	$0.23_{(0.09)}$
$\mathbf{0.10_{(0.01)}}$	$0.27_{(0.04)}$	$0.12_{(0.01)}$	$0.21_{(0.09)}$
$\mathbf{0.13_{(0.02)}}$	$0.33_{(0.02)}$	$0.17_{(0.01)}$	$0.26_{(0.09)}$
$\mathbf{0.13_{(0.02)}}$	$0.31_{(0.03)}$	$0.17_{(0.02)}$	$0.25_{(0.09)}$
$\mathbf{0.12_{(0.02)}}$	$0.28_{(0.04)}$	$0.17_{(0.01)}$	$0.25_{(0.10)}$
$\mathbf{0.11_{(0.01)}}$	$0.29_{(0.03)}$	$0.12_{(0.01)}$	$0.24_{(0.08)}$
$\mathbf{0.11_{(0.01)}}$	$0.27_{(0.03)}$	$0.13_{(0.01)}$	$0.23_{(0.08)}$
$\mathbf{0.10_{(0.01)}}$	$0.24_{(0.03)}$	$0.12_{(0.01)}$	$0.23_{(0.10)}$
$\mathbf{0.07_{(0.01)}}$	$0.22_{(0.04)}$	$0.08_{(0.01)}$	$0.17_{(0.07)}$
$\mathbf{0.06_{(0.01)}}$	$0.21_{(0.03)}$	$0.08_{(0.01)}$	$0.17_{(0.07)}$
$\mathbf{0.06_{(0.01)}}$	$0.19_{(0.02)}$	$0.07_{(0.01)}$	$0.16_{(0.08)}$
Fixed x			
NN	NW	LL	KMN
$\mathbf{0.12_{(0.11)}}$	$0.33_{(0.14)}$	$0.27_{(0.12)}$	$0.31_{(0.19)}$
$\mathbf{0.09_{(0.08)}}$	$0.32_{(0.17)}$	$0.23_{(0.09)}$	$0.22_{(0.15)}$
$\mathbf{0.06_{(0.08)}}$	$0.33_{(0.18)}$	$0.23_{(0.09)}$	$0.11_{(0.08)}$
$\mathbf{0.13_{(0.09)}}$	$0.23_{(0.09)}$	$0.18_{(0.09)}$	$0.25_{(0.12)}$
$\mathbf{0.10_{(0.07)}}$	$0.22_{(0.10)}$	$0.14_{(0.05)}$	$0.19_{(0.12)}$
$\mathbf{0.07_{(0.07)}}$	$0.21_{(0.11)}$	$0.14_{(0.05)}$	$0.11_{(0.09)}$
$\mathbf{0.13_{(0.08)}}$	$0.17_{(0.07)}$	$0.26_{(0.08)}$	$0.20_{(0.08)}$
$\mathbf{0.11_{(0.07)}}$	$0.17_{(0.08)}$	$0.22_{(0.03)}$	$0.19_{(0.08)}$
$\mathbf{0.08_{(0.06)}}$	$0.14_{(0.10)}$	$0.22_{(0.03)}$	$0.14_{(0.07)}$

Table 2: Computation times (in seconds) averaged over different settings and sample sizes. All results are based on 200 replications.

Setting	n	NN	NW	LL	KMN
Setting 1.1	500	0.27	14.21	15.93	0.54
	1000	0.68	25.36	26.85	0.54
	2000	0.89	49.18	50.71	0.54
Setting 2.1	500	1.30	36.16	53.59	1.07
	1000	1.38	51.86	63.28	1.03
	2000	1.78	80.02	82.21	1.08
Setting 1.2	500	2.62	71.76	102.38	1.53
	1000	2.91	104.04	124.13	1.52
	2000	3.98	158.58	162.53	1.59
Setting 2.2	500	3.90	153.06	225.38	2.52
	1000	5.68	231.09	275.10	2.48
	2000	9.91	332.82	340.49	2.39

Computations were performed on an Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with an NVIDIA GeForce RTX 2080 Ti GPU.

$n = 500$), but its performance improves as the sample size increases to $n = 2000$. In contrast, when the dimension increases to $d = 300$ (Table 1b and Table 1d), our proposed NN method demonstrates a clear advantage over competing approaches, as deep neural networks can more effectively capture complex patterns and interactions among high-dimensional covariates.

Regarding the comparison between additive and multiplicative error structures, we observe that the relative performance of the methods remains consistent across both error types. All four methods achieve similar estimated excess risk under both additive (Model 1) and multiplicative (Model 2) error structures, suggesting that the methods are robust to different noise mechanisms. However, the multiplicative error structure leads to longer computation times for all methods, as shown in Table 2. This is because multiplicative errors induce more complex and heterogeneous conditional distributions, requiring additional computational effort to estimate accurately.

An important trade-off emerges when comparing the neural network-based methods (NN and KMN) with classical kernel methods (NW and LL), particularly evident in the fixed-$\mathbf{x}$ evaluation protocol. While the NN method achieves lower estimated excess risk (better bias) compared to NW and LL, it exhibits higher standard deviations. This reflects a characteristic bias–variance trade-off in neural network-based estimators: their flexibility enables better fit to complex conditional densities but may introduce greater variability across random samples. The NN method is particularly advantageous when reducing bias is critical in complex, high-dimensional settings, while classical methods may be preferred when stability across samples is paramount.

Finally, our proposed method offers computational advantages, particularly as the sample size and dimensionality increase. As reported in Table 2, the NN method achieves computation times similar to the KMN method and is faster than the classical kernel methods NW and LL across all settings. Moreover, Table 2 shows that, for all methods, computation time increases when moving from low to high dimension and is typically larger under multiplicative errors than under additive errors. This advantage arises because NW and LL require repeated local estimation at each evaluation point, while our NN method (and KMN) produces a global estimator that can be evaluated efficiently.

6.2 Real Data Analysis

We illustrate the performance of our method using two real datasets: the Photometric Redshift Dataset in astronomy (Izbicki and Lee, 2016) and the Typhoon Center Best Track Dataset in meteorology (Cloud et al., 2019). To compare the performance with the other two methods, we use the empirical likelihood metric inspired by Hinder et al. (2021). Specifically,

given a real dataset $\mathcal{D}_n$, we leave out one data point $(\mathbf{X}_j, Y_j)$ and estimate the conditional density $f(Y_j \mid \mathbf{X}_j)$ based on the dataset $\mathcal{D}_{-j} = \{(\mathbf{X}_i, Y_i), i \neq j\}$. If we draw a set of indices $\mathcal{J}$ with $|\mathcal{J}| = J$, the empirical likelihood is calculated by

$$\hat{l} = \frac{1}{J} \sum_{j \in \mathcal{J}} \hat{f}(Y_j \mid \mathbf{X}_j). \tag{6.3}$$

Photometric Redshift Dataset The interest in conditional density estimation for photometric redshift is well-documented in previous studies (Sheldon et al. 2012, Izbicki et al. 2017, Izbicki and Lee 2017). Conditional density estimation enables researchers to estimate the distance to galaxies based on their observed magnitudes, which are measured across five different wavelength bands: ultraviolet (u), green (g), red (r), infrared (i), and near-infrared (z), along with their associated errors, and the response is the redshift. In addition to providing the conditional mean, density estimation offers a more comprehensive quantification of the redshift, which is the primary focus of our research. We report the means and standard deviations of the metric (6.3) under different methods in Table 3a. Our approach has demonstrated significant improvements compared to other methods.

To further illustrate our findings, we estimate the conditional density by fixing other variables at their means and varying the type of light. The results in Figure 1a reveal that the ultraviolet light (solid line) is multi-modal distributed, while the other types of light are uni-modal distributed corresponding to the conditional density of redshift. This highlights the potential for more nuanced and accurate estimations of redshift based on the characteristics of the observed light.

Table 3: Results of real data analyses. The means and standard deviations of the empirical likelihood are reported. NN denotes our method, NW denotes the Nadaraya–Watson estimator, LL denotes the local linear estimator, and KMN denotes the kernel mixture network.

(a) Photometric redshift dataset.

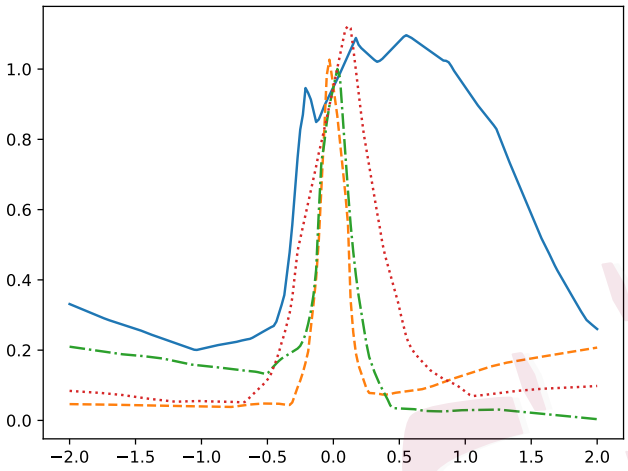
	NN	NW	LL	KMN
Mean	1.019	0.746	0.764	0.828
S. D.	0.204	0.235	0.265	0.360

(b) Typhoon center pressure dataset.

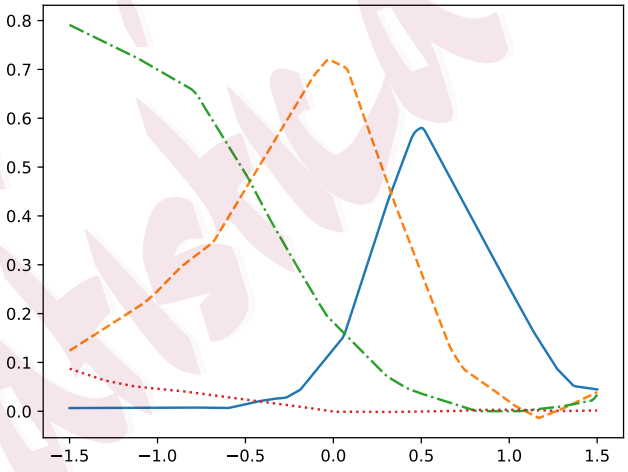
	NN	NW	LL	KMN
Mean	0.581	0.522	0.538	0.564
S. D.	0.092	0.098	0.112	0.308

Typhoon Center Pressure Dataset Tropical typhoons are powerful and destructive weather phenomena that can cause significant damage to coastal areas and communities. Understanding the behavior and characteristics of typhoons is crucial for predicting their intensities given location and other conditions, which can help in mitigating their impact on human lives and infrastructure (Cloud et al. 2019, Huberman et al. 2022). The dataset includes the location (i.e., latitude and longitude), the typhoon grade, the wind speed, and the central pressure. The particular interest is the estimation of the density of central pressure given other characteristics. The corresponding results are presented in Table 3b, where our approach also demonstrates a significant improvement compared with other methods.

Furthermore, we plot the conditional density under different central pressure y. In Figure 1b, the solid, dashed, dash-dotted, and dotted lines indicate increasing levels of central pressure. From the figure, it is evident that as the central pressure increases, the density reaches its mode at a lower grade, which is consistent with findings from (Rosendal and Shaw, 1982). Their research found that the intensity of a hurricane is closely related to its central pressure, with lower pressure values indicating a more intense storm.



(a) Visualization of the conditional density of the photometric redshift dataset with y fixed at its mean. The x-axis represents the intensity of light, while the y-axis represents the density of redshift. In this representation, the blue solid, orange dashed, green dash-dotted and red dotted lines correspond to the estimated density distributions of ultraviolet light, green light, red light, and infrared light, respectively.



(b) Visualization of the conditional density of the typhoon center pressure dataset. The x-axis represents the grade of the typhoon, while the y-axis represents the density of a fixed central pressure under different grades. In this representation, the blue solid, orange dashed, green dash-dotted, and red dotted lines indicate increasing levels of central pressure.

Figure 1: Real datasets visualizations.

7. Related Work

In this section, we discuss the related literature by focusing on three key aspects: nonparametric regression using neural networks, conditional density estimation using kernel functions, and (conditional) density estimation using deep learning.

7.1 Nonparametric Regression with Neural Networks

Nonparametric regression based on ReLU-activated deep neural networks has been extensively studied in the literature (Schmidt-Hieber, 2020; Farrell et al., 2021; Jiao et al., 2023). By formulating conditional density estimation as a nonparametric regression problem, our approach leverages a few well-established results, such as the approximation error bounds of deep neural networks given by Jiao et al. (2023).

However, our primary focus is on conditional density estimation, and our work differs from previous studies in two main respects, the transformation of the response using a kernel function and the irregular behavior of the error term. To address the approximation error introduced by the kernel function, we derive a modified oracle inequality and control this error by selecting the appropriate kernel bandwidth. For the irregular behavior of the error term, we employ a truncation technique and similarly manage the error through bandwidth selection.

7.2 Kernel Methods for Conditional Density Estimation

In the literature on conditional density estimation using kernel functions, classical approaches rely on kernel functions to approximate the conditional density locally (Rosenblatt 1956, Parzen 1962, Hyndman et al. 1996). Fan et al. (1996) and Hyndman and Yao (2002) intro-

duced a local linear estimator for estimating the conditional density at a given point based on linear approximation.

We briefly summarize the convergence rates of the classical NW-type kernel conditional density estimator and the local linear estimator, and compare them with our approach.

For the NW estimator, Hyndman et al. (1996) established the convergence rate for conditional density estimation. When the density function is twice differentiable, the NW estimator achieves the optimal rate $n^{-2/3}$ for the univariate (X, Y) case. In Supplementary Material S10, we show that by employing higher-order kernels for both $\mathbf{x}$ and y, the NW estimator can attain the minimax optimal rate $n^{-2\beta/(2\beta+d+1)}$ for the integrated mean squared error over $(\mathbf{X}, Y)$.

For the local linear conditional density estimator, based on the theoretical analysis in Equations (3.1)–(3.5) of Hyndman and Yao (2002), the convergence rate of the IMSE is of order $n^{-4\beta/(5\beta+2)}$, which is minimax optimal when $\beta = 2$, while it becomes suboptimal when $\beta > 2$. This scenario arises because their approximation $\mathbb{E}_Y\{K_b(Y - y) \mid \mathbf{X} = \mathbf{x}\} = f_*(y \mid \mathbf{x}) + O(b^2)$ introduces a bias term of order b^2, which prevents the method from fully exploiting smoothness beyond second order.

Our work differs in both the estimator and the risk formulation. We recast conditional density estimation as a regression problem by fixing y and regressing the transformed response $K_b(Y - y)$ on $\mathbf{X}$, and we analyze the excess risk for this fixed y integrated over $\mathbf{x}$. Consequently, the effective dimension is d instead of $d + 1$, leading to the minimax rate $n^{-2\beta/(2\beta+d)}$ (up to logarithmic factors). Our approach advances the field in two aspects. First, we estimate the conditional density globally with respect to $\mathbf{x}$, eliminating the need to compute local weights for each query point as required by kernel-based estimators, which

reduces computational complexity. Second, our estimator adapts to the intrinsic geometry of the data. When the data lies on a manifold of dimension $d_{\mathcal{M}} < d$, our estimator achieves a faster rate depending on $d_{\mathcal{M}}$.

7.3 Deep Learning for (Conditional) Density Estimation

Beyond classical kernel-based estimators, deep learning has also been widely used for (conditional) density estimation.

One line of work combines neural networks with kernel mixture models. For example, Ambrogioni et al. (2017) and Rothfuss et al. (2019) employ neural networks to predict the parameters of a mixture distribution, enabling flexible conditional density estimation.

Another prominent line of work focuses on generative models, which learn a transformation that maps a reference distribution to the target distribution and thereby represent the (conditional) density. This category includes generative adversarial networks (GANs) (Goodfellow et al., 2014; Mirza and Osindero, 2014), normalizing flows (Rezende and Mohamed, 2015; Dinh et al., 2017; Chen et al., 2018), and conditional variational autoencoders (Kingma and Welling, 2014; Sohn et al., 2015). More recently, diffusion models (Song and Ermon, 2019; Ho et al., 2020; Song et al., 2021) and flow models (Liu et al., 2023; Albergo and Vanden-Eijnden, 2022; Lipman et al., 2023) have gained popularity. Theoretical analysis of these generative models has also been conducted for GANs (Huang et al., 2022; Stéphanovitch et al., 2024; Zhou et al., 2023), diffusion models (Dai et al., 2025), and continuous normalizing flows (Gao et al., 2024; Cai and Li, 2025; Fukumizu et al., 2025). These theoretical analyses typically investigate the convergence of the Wasserstein distance or the total variation distance between the estimated distribution and the true distribution, where

the distribution is approximated by sampling the synthetic data. In contrast, our approach provides direct access to the conditional density after training and allows for the analysis of the L_2-risk for any given y and over $\mathbf{x}$.

Beyond generative models, there is growing interest in establishing theoretical foundations for deep learning-based density estimators from a statistical perspective. Han et al. (2025) developed a deep mutual density ratio estimation method with applications to conditional density estimation. Most closely related to our work is Bos and Schmidt-Hieber (2024), who proposed a supervised deep learning method for density estimation that reformulates the problem as a regression using kernel transformation to construct the response from additional data points. They established that their estimator achieves the minimax optimal rate under compositional structure assumptions. While our approach also employs kernel transformation to transform the response, our primary focus lies in conditional density estimation, which inherently adopts an auto-supervised framework and does not require additional data points. Our work contributes to this growing literature by establishing the minimax optimal convergence rate for conditional density estimation using deep neural networks.

8. Conclusion and Discussion

In this paper, we have introduced a novel conditional density estimator based on deep ReLU networks. We have proved that the estimator achieves the minimax optimal rate under the Hölder class and is adaptable to low-dimensional manifold settings. Our nonparametric approach does not assume any underlying parametric structure and offers global estimation over $\mathbf{x}$ for fixed y, outperforming classical kernel estimators in simulations and real data applications. These findings suggest the robustness and practical potential of our method

in conditional density estimation. Future work may focus on establishing theoretical convergence guarantees for the extended estimator and on exploring more efficient training strategies for deep neural networks.

Supplementary Material

The Supplementary Material contains additional simulation results and provides proofs for each result stated in the paper.

Acknowledgements

We are grateful to the Editor, Associate Editor, and three anonymous reviewers for their valuable comments and suggestions, which significantly improved the quality of this article. Liping Zhu's work was supported by the National Key R&D Program of China (2023YFA1008702), the National Natural Science Foundation of China (12225113), and the Public Computing Cloud, Renmin University of China. Jian Huang's work was supported by the National Natural Science Foundation of China (72331005) and the research grants from The Hong Kong Polytechnic University (P0046811, P0042888, P0045417, P0045931).

References

Albergo, M. S. and E. Vanden-Eijnden (2022). Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*.

Ambrogioni, L., U. Güçlü, M. A. J. van Gerven, and E. Maris (2017). The kernel mixture network: A nonparametric method for conditional density estimation of continuous random variables. arXiv:1705.07111.

Bartlett, P. L., N. Harvey, C. Liaw, and A. Mehrabian (2019). Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research 20*(63), 1–17.

Bishop, C. M. (1994). Mixture density networks. Technical report, Aston University.

Bos, T. and J. Schmidt-Hieber (2024). A supervised deep learning method for nonparametric density estimation. *Electronic Journal of Statistics 18*(2), 5601–5658.

Cai, C. and G. Li (2025). Minimax optimality of the probability flow ODE for diffusion models. arXiv:2503.09583.

Chen, R. T. Q., Y. Rubanova, J. Bettencourt, and D. K. Duvenaud (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, Volume 31. Curran Associates, Inc.

Cloud, K. A., B. J. Reich, C. M. Rozoff, S. Alessandrini, W. E. Lewis, and L. D. Monache (2019). A feed forward neural network based on model output statistics for short-term hurricane intensity prediction. *Weather and Forecasting 34*(4), 985–997.

Dai, D., J. Fan, Y. Gu, and D. Mukherjee (2025). CINDES: Classification induced neural density estimator and simulator. arXiv:2510.00367.

De Gooijer, J. G. and D. Zerom (2003). On conditional density estimation. *Statistica Neerlandica 57*(2), 159–176.

Dinh, L., J. Sohl-Dickstein, and S. Bengio (2017). Density estimation using Real NVP. In *International Conference on Learning Representations*.

Efromovich, S. (2007). Conditional density estimation in a regression setting. *The Annals of Statistics 35*(6), 2504–2535.

Fan, J., Q. Yao, and H. Tong (1996). Estimation of conditional densities and sensitivity measures in nonlinear dynamical systems. *Biometrika 83*(1), 189–206.

Farrell, M. H., T. Liang, and S. Misra (2021). Deep neural networks for estimation and inference. *Econometrica 89*(1), 181–213.

Fukumizu, K., T. Suzuki, N. Isobe, K. Oko, and M. Koyama (2025). Flow matching achieves almost minimax optimal convergence. In *The Thirteenth International Conference on Learning Representations*.

Gao, Y., J. Huang, Y. Jiao, and S. Zheng (2024). Convergence of continuous normalizing flows for learning probability distributions. arXiv:2404.00551.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Volume 27. Curran Associates, Inc.

Györfi, L., M. Kohler, A. Krzyżak, and H. Walk (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer Series in Statistics. New York, NY: Springer.

Han, D., S. Zheng, G. Shen, X. Song, L. Sun, and J. Huang (2025). Deep mutual density ratio estimation with bregman divergence and its applications. *Journal of the American Statistical Association 120*(551), 1990–2001.

Hinder, F., V. Vaquet, J. Brinkrolf, and B. Hammer (2021). Fast non-parametric conditional density estimation using moment trees. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–7.

Ho, J., A. Jain, and P. Abbeel (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 6840–6851. Curran Associates, Inc.

Huang, J., Y. Jiao, Z. Li, S. Liu, Y. Wang, and Y. Yang (2022). An error analysis of generative adversarial networks for learning distributions. *Journal of Machine Learning Research 23*(116), 1–43.

Huberman, D. B., B. J. Reich, and H. D. Bondell (2022). Nonparametric conditional density estimation in a deep learning framework for short-term forecasting. *Environmental and Ecological Statistics 29*(4), 677–704.

Hyndman, R. J., D. M. Bashtannyk, and G. K. Grunwald (1996). Estimating and visualizing conditional densities. *Journal of Computational and Graphical Statistics 5*(4), 315–336.

Hyndman, R. J. and Q. Yao (2002). Nonparametric estimation and symmetry tests for conditional density functions. *Journal of Nonparametric Statistics 14*(3), 259–278.

Izbicki, R. and A. B. Lee (2016). Nonparametric conditional density estimation in a high-dimensional regression setting. *Journal of Computational and Graphical Statistics 25*(4), 1297–1316.

Izbicki, R. and A. B. Lee (2017). Converting high-dimensional regression to high-dimensional conditional density estimation. *Electronic Journal of Statistics 11*(2), 2800–2831.

Izbicki, R., A. B. Lee, and P. E. Freeman (2017). Photo-z estimation: An example of nonparametric conditional density estimation under selection bias. *The Annals of Applied Statistics 11*(2), 698–724.

Jiao, Y., G. Shen, Y. Lin, and J. Huang (2023). Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics 51*(2), 691–716.

Kingma, D. P. and M. Welling (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*.

Li, M., M. Neykov, and S. Balakrishnan (2022). Minimax optimal conditional density estimation under total variation smoothness. *Electronic Journal of Statistics 16*(2), 3937–3972.

Li, Q. and J. S. Racine (2006). *Nonparametric Econometrics: Theory and Practice*. Princeton University Press.

Lindgren, G. (2012). *Stationary Stochastic Processes: Theory and Applications*. CRC Press.

Lipman, Y., R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le (2023). Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*.

Liu, X., C. Gong, and Q. Liu (2023). Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*.

Mirza, M. and S. Osindero (2014). Conditional generative adversarial nets. arXiv:1411.1784.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics 33*(3), 1065–1076.

Rezende, D. J. and S. Mohamed (2015). Variational inference with normalizing flows. In *Proceedings of the 32 Nd*

International Conference on Machine Learning, Lille, France.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics 27*(3), 832–837.

Rosendal, H. E. and S. L. Shaw (1982). Relationship of maximum sustained winds to minimum sea level pressure in central North Pacific tropical cyclones. Technicacl memorandum, United States, National Weather Service., Pacific Region.

Rothfuss, J., F. Ferreira, S. Walther, and M. Ulrich (2019). Conditional density estimation with neural networks: Best practices and benchmarks. arXiv:1903.00954.

Ruzgas, T., M. Lukauskas, and G. **v**Cepkauskas (2021). Nonparametric multivariate density estimation: Case study of Cauchy mixture model. *Mathematics 9*(21), 2717.

Samani, F. S., R. Stadler, C. Flinta, and A. Johnsson (2021). Conditional density estimation of service metrics for networked services. *IEEE Transactions on Network and Service Management 18*(2), 2350–2364.

Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics 48*(4), 1875–1897.

Scott, D. W. (2015). *Multivariate Density Estimation: Theory, Practice, and Visualization* (2nd edition ed.). Hoboken, New Jersey: Wiley.

Sheldon, E. S., C. E. Cunha, R. Mandelbaum, J. Brinkmann, and B. A. Weaver (2012). Photometric redshift probability distributions for galaxies in the SDSS DR8. *The Astrophysical Journal Supplement Series 201*(2), 32.

Sohn, K., H. Lee, and X. Yan (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, Volume 28. Curran Associates, Inc.

Song, Y. and S. Ermon (2019). Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

Song, Y., J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

Stéphanovitch, A., E. Aamari, and C. Levrard (2024). Wasserstein generative adversarial networks are minimax optimal distribution estimators. *The Annals of Statistics 52*(5), 2167–2193.

Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics 10*(4), 1040–1053.

Vardi, G., G. Yehudai, and O. Shamir (2022). Width is less important than depth in ReLU neural networks. In *Proceedings of Thirty Fifth Conference on Learning Theory*, pp. 1249–1281. PMLR.

Yan, X., Y. Su, and W. Ma (2023). Ensemble multi-quantiles: Adaptively flexible distribution prediction for uncertainty quantification. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*(11), 13068–13082.

Zhou, X., Y. Jiao, J. Liu, and J. Huang (2023). A deep generative approach to conditional sampling. *Journal of the American Statistical Association 118*(543), 1837–1848.

Chenxuan He

Institute of Statistics and Big Data, Renmin University of China, Beijing, China

E-mail: hechenxuan@ruc.edu.cn

Yuan Gao

LPMC & KLMDASR, School of Statistics and Data Science, Nankai University, Tianjin, China

E-mail: yuangao@nankai.edu.cn

Liping Zhu

Center for Applied Statistics and Institute of Statistics and Big Data, Renmin University of China, Beijing, China

E-mail: zhu.liping@ruc.edu.cn

Jian Huang

Department of Data Science and AI and Department of Applied Mathematics, The Hong Kong Polytechnic University,

Hong Kong, China

E-mail: j.huang@polyu.edu.hk

Chenxuan He and Yuan Gao are co-first authors.