Journal of the Royal Statistical Society Series B: Statistical Methodology, 2026, **88**, 330–351
https://doi.org/10.1093/jrsssb/qkaf053
Advance access publication 25 August 2025
Original Article

ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS
Journal of the Statistics Society
Series B
Statistical Methodology
B

Wasserstein generative regression

Shanshan Song[1,†], Tong Wang[2,†], Guohao Shen[3], Yuanyuan Lin[2] and Jian Huang[4]

[1]School of Mathematical Sciences, School of Economics and Management, and Key Laboratory of Intelligent Computing and Applications (Ministry of Education), Tongji University, Shanghai 200092, China
[2]Department of Statistics, The Chinese University of Hong Kong, Hong Kong SAR 999077, China
[3]Department of Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR 999077, China
[4]Departments of Data Science and AI, and Applied Mathematics, The Hong Kong Polytechnic University, Hong Kong SAR 999077, China

Address for correspondence: Yuanyuan Lin, Department of Statistics, The Chinese University of Hong Kong, Hong Kong SAR 999077, China. Email: ylin@sta.cuhk.edu.hk

Abstract

In this paper, we propose a new and unified approach for nonparametric regression and conditional distribution learning. Our approach simultaneously estimates a regression function and a conditional generator using a generative learning framework, where a conditional generator is a function that can generate samples from a conditional distribution. The main idea is to estimate a conditional generator satisfying the constraint that it produces a good regression function estimator. We use deep neural networks to model the conditional generator. Our approach can handle problems with multivariate outcomes and covariates, and can be used to construct prediction intervals. We provide theoretical guarantees by deriving nonasymptotic error bounds and the distributional consistency of our approach under suitable assumptions. We perform numerical experiments to demonstrate the effectiveness and superiority of our approach over some existing approaches in various scenarios.

Keywords: conditional distribution, deep neural networks, generative learning, nonparametric regression

1 Introduction

Regression models and conditional distributions play a key role in a variety of prediction and inference problems in statistics. There is a vast literature on nonparametric methods for regression analysis, as well as the estimation of conditional density, distribution, and quantiles. Most existing methods use smoothing and basis expansion techniques, including kernel smoothing, local polynomials, and splines (Cai, 2002; Fan & Gijbels, 1996; Györfi et al., 2002; Hall & Müller, 2003; Hall et al., 1999; Racine & Li, 2017; Scott, 1992; Silverman, 1986; Tsybakov, 2008; Veraverbeke et al., 2014; Wasserman, 2006; Yu & Jones, 1998). However, the existing nonparametric regression and conditional density or distribution estimation methods suffer from the 'curse of dimensionality', that is, their performance deteriorates dramatically as the dimensionality of data increases. Indeed, most existing methods can only effectively handle up to a few predictors. Moreover, most existing methods only consider the case when the response is a scalar, but are not applicable to the settings with a high-dimensional response vector.

To circumvent the curse of dimensionality, many researchers have proposed and studied non- and semi-parametric models that impose certain structural constraints that reduce the model dimensionality. Some notable examples include the single-index model (Hardle et al., 1993;

† Shanshan Song and Tong Wang are co-first authors.

Received: June 20, 2023. **Revised:** May 12, 2025. **Accepted:** July 25, 2025
© The Royal Statistical Society 2025.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Ichimura, 1993), the generalized additive model (Hastie & Tibshirani, 1986; Stone, 1986), and the projection pursuit model (Friedman & Stuetzle, 1981), among others. These methods, though make strong assumptions about the model structure, are popular due to their interpretability. They focus on estimating the regression function but not the conditional distribution, thus they are effective to provide point prediction. In the Bayesian paradigm, there is a vast amount of work that can provide interval predictions and quantify the uncertainty of model parameters (Antoniadis et al., 2004; Fahrmeir & Lang, 2001; Hilton et al., 2019; Klein et al., 2015; Reich et al., 2011; West et al., 1985).

In recent years, there have been significant advancements in *deep generative learning* (Salakhutdinov, 2015), in which deep neural networks are used to approximate high-dimensional functions, such as generator and discriminator functions. In particular, for learning distributions of high-dimensional data arising in image analysis and natural language processing, the generative adversarial networks (GANs; Arjovsky et al., 2017; Goodfellow et al., 2014) have proven to be effective and achieved impressive success (Reed et al., 2016; Zhu et al., 2017). Instead of estimating the functional form of a density or distribution function, GANs start from a fixed noise distribution and learn a map that pushes the noise distribution to the data distribution. GANs have also been extended to learn conditional distributions (Zhou et al., 2022).

One of the main challenges in nonparametric regression is to estimate a function that can accurately capture the relationship between covariate and response variables. Generative adversarial networks can learn complex distributions. However, to the best of our knowledge, there have not been systematic studies on how GANs can be used for nonparametric regression, despite their successes in distribution learning. Furthermore, it is still unclear whether conditional GANs, as a natural extension of GANs for learning conditional distributions, can directly provide satisfactory estimation of a regression function. Our work will study how to use conditional GANs for regression tasks.

We propose a new and unified approach for nonparametric regression and conditional distribution learning. Our approach estimates the regression function and a conditional generator at the same time using a generative learning method. A conditional generator is a function that transforms a random vector from a fixed noise distribution to the response variable space, which can be used to sample from a conditional distribution. Thus, when a conditional generator is estimated, it can be used to explore the target conditional distribution. Theoretically, the regression function is the expectation of the conditional generator with respect to the noise distribution. However, empirically such an expectation may not produce a good estimator of the regression function.

Our main idea is to constrain the conditional generator to produce samples that minimize the quadratic loss of the regression function, which is computed as the expectation of the conditional generator. Specifically, in the objective function for estimating the conditional generator based on distribution matching using the Wasserstein distance, we incorporate a quadratic loss term to control the error of the estimated regression function. We use deep neural networks to approximate the conditional generator, which can capture the complex structure of the data distribution. In principle, other approximation methods, such as splines, can also be used. However, deep neural network approximation has the important advantage of being able to adapting to the latent structure of the data distribution. For simplicity, we call our method Wasserstein generative regression (WGR).

The proposed method has several attractive properties. First, it is applicable to problems with a high-dimensional response variable, while the existing methods typically only consider the case of a scalar response. Second, the proposed method allows continuous, discrete and mixed types of predictors and responses, while the smoothing and basis expansion methods are mainly applicable to continuous-type variables. Third, since the proposed method learns a conditional distribution generator, it can be used for constructing prediction intervals. In comparison, the existing nonparametric regression can only give point prediction. Finally, the proposed method is able to adapt to the latent data structure in a data-driven manner and thus can mitigate the curse of dimensionality, under the assumption that the data distribution is supported on an approximate low-dimensional set.

The rest of the paper is organized as follows. In Section 2, we describe the proposed WGR method. We present the implementation details in Section 3. In Section 4, we establish nonasymptotic

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

error bounds for the proposed estimator and show that it is consistent. In Section 5, we conduct numerical experiments, including simulation studies and real data analysis, to evaluate the performance of the proposed method. Some concluding remarks are presented in Section 6. Technical proofs and additional numerical experiments are given in the online supplementary material.

2 Method

Consider a pair of random vectors $(X, Y) \in \mathcal{X} \times \mathcal{Y}$, where X is a vector of predictors and Y is a vector of response variables. Suppose $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^q$ with $d, q \geq 1$. We allow either or both of X and Y to be high-dimensional. The predictor X or the response Y can contain both continuous and categorical components. Our goal is to learn the conditional distribution of Y given $X = x$ and estimate the regression function $\mathbb{E}(Y \mid X = x)$ in a unified framework.

We describe the proposed WGR method in detail below, which has three main ingredients, a conditional distribution generator, a quadratic loss for regression, and the Wasserstein metric for distribution matching.

2.1 Conditional generator

The theoretical foundation of WGR is the noise-outsourcing lemma in probability theory. The original version of the lemma (Kallenberg, 2002, Theorem 6.10; Austin, 2015, Lemma 3.1) states that, if $\mathcal{Y}$ is a standard Borel space, there exists a Borel-measurable function $g^* : \mathcal{X} \times [0, 1] \to \mathcal{Y}$ and a random variable $\eta \sim \text{Uniform}[0, 1]$ such that η is independent of X and

$$(X, Y) = (X, g^*(X, \eta)) \quad \text{almost surely.} \tag{1}$$

The condition that $\mathcal{Y}$ being a standard Borel space is mild and satisfied in various applications. A more general version of the lemma allows η to be an m-dimensional random vector ($m \geq 1$) following a fixed continuous distribution P_η, e.g. $\eta \sim N(\mathbf{0}, \mathbf{I}_m)$, the m-dimensional multivariate standard normal distribution as in Zhou et al. (2022) and Sharma et al. (2023). Note that the dimension of η does not necessarily be the same as the dimension of Y.

Remark 1 The noise-outsourcing lemma ensures the existence of η and g^*_η, in the sense that for $\eta \sim P_\eta$, one can always construct a Borel-measurable function $g^*_\eta : \mathcal{X} \times \mathbb{R}^m \to \mathcal{Y}$ satisfying $(X, Y) = (X, g^*_\eta(X, \eta))$ almost surely. To see this, we consider $P_\eta \sim N(\mathbf{0}, \mathbf{I}_m)$. In this setting, there exists a measurable function $h : \mathbb{R}^m \to [0, 1]$ such that $h(\eta) \sim \text{Uniform}[0, 1]$. For example, let $h(\eta) := \Phi(\eta_{(1)})$, where $\eta_{(1)}$ is the first element of η, and Φ is the cumulative distribution function of the univariate standard normal distribution. Then, the original noise-outsourcing lemma (Kallenberg, 2002, Theorem 6.10; Austin, 2015, Lemma 3.1) ensures that there exists a function $g^*_\eta(x, \eta) := g^*(x, h(\eta))$ such that $(X, Y) = (X, g^*_\eta(X, \eta))$ almost surely. This fact indicates that g^*_η can be constructed accordingly for η following any fixed continuous distribution P_η. For notational simplicity, for $\eta \sim P_\eta$, we still write its corresponding g^*_η as g^* to suppress its dependence on η for the rest of this paper.

The function g^* is not unique, even when P_η is specified. Our theoretical analysis of the proposed method in Section 4 require only the existence of g^*, and do not rest on its uniqueness or identifiability.

We call the function g^* in (1) a conditional generator, since satisfying (1) implies that it also satisfies

$$g^*(x, \eta) \sim P_{Y \mid X = x}, \ \eta \sim P_\eta, \ x \in \mathcal{X}. \tag{2}$$

So for a given x, to sample from the conditional distribution $P_{Y \mid X = x}$, we can first generate $\eta \sim P_\eta$, then calculate $g^*(x, \eta)$, which gives a sample from $P_{Y \mid X = x}$. In addition, we can calculate any

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

moments of $P_{Y|X=x}$ via $g^*(x, \cdot)$. In particular,

$$\mathbb{E}(Y \mid X = x) = \mathbb{E}_\eta[g^*(x, \eta)], x \in \mathcal{X},$$

where $\mathbb{E}_\eta$ is the expectation with respect to η.

In summary, we can determine the usual regression function (the conditional mean) and sample from the conditional distribution as follows:

- Regression function or conditional mean: $\mathbb{E}(Y \mid X = x) = \mathbb{E}_\eta g^*(x; \eta), x \in \mathcal{X}$,
- Conditional distribution: $g^*(x; \eta) \sim P_{Y|X=x}, x \in \mathcal{X}, \eta \sim P_\eta$.

Hence, the conditional generator provides a basis for a unified framework for nonparametric regression and conditional distribution learning.

2.2 Objective function

Let $P_{X,g}$ denote the joint distribution of $(X, g(X, \eta))$, which is the generated distribution based on a conditional generator $g(x, \eta)$, $\eta \sim P_\eta$, $x \in \mathcal{X}$. One possible way to measure the quality of a conditional generator g is to compare the generated distribution $P_{X,g}$ with the data distribution $P_{X,Y}$. A good conditional generator g should ensure that the generated distribution is close to the data distribution in some sense. For example, one can use a distance metric such as the Wasserstein distance or a divergence measure such as the Kullback–Leibler divergence to quantify the discrepancy between the two distributions.

Let $\mathbb{D}$ be a divergence measure for the difference between $P_{X,g}$ and $P_{X,Y}$. Then, we formulate an objective function that combines this divergence measure with the least squares loss to minimize the distribution mismatch and the prediction error simultaneously. Throughout the paper, for a vector $x \in \mathbb{R}^d$, its $\ell_1, \ell_2, \ell_\infty$-norm is defined as $\|x\|_1 = \sum_{k=1}^d |x_k|$, $\|x\| = (\sum_{k=1}^d |x_k|^2)^{1/2}$, and $\|x\|_\infty = \max_{1 \le k \le d} |x_k|$, respectively. The objective function is

$$\lambda_w \mathbb{D}(P_{X,g} \| P_{X,Y}) + \lambda_\ell \mathbb{E}_{(X,Y)} \| Y - \mathbb{E}_\eta g(X, \eta) \|^2, \tag{3}$$

where the expectation $\mathbb{E}_{(X,Y)}$ is taken with respect to (X, Y). Here, both λ_ℓ and λ_w are tuning parameters weighing two losses, which are assumed to be nonnegative and $\lambda_\ell + \lambda_w = 1$. The objective function (3) combines two types of losses: the first one evaluates how closely the generated distribution $P_{X,g}$ resembles the data distribution $P_{X,Y}$; the second one is a criterion quantifying how well the regression function fits the data. Intuitively, the objective function (3) tries to learn the conditional distribution of Y given X with the regularization that the conditional mean is well estimated.

We take $\mathbb{D}$ to be the *1-Wasserstein* distance, using the Monge–Rubinstein dual (Villani, 2009) for computational convenience:

$$\mathbb{D}_{\mathrm{W}}(P_{X,g}, P_{X,Y}) = \sup_{f \in \mathcal{F}^1_{\mathrm{Lip}}} \left\{ \mathbb{E}_{(X,\eta)} f(X, g(X, \eta)) - \mathbb{E}_{(X,Y)} f(X, Y) \right\}, \tag{4}$$

where $\mathcal{F}^1_{\mathrm{Lip}} = \{f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}, |f(u) - f(v)| \le \|u - v\|, \forall u, v \in \mathcal{X} \times \mathcal{Y}\}$ is a 1-Lipschitz class of functions on $\mathcal{X} \times \mathcal{Y}$, and $\mathbb{E}_{(X,\eta)}$ is the expectation with respect to (X, η). The Lipschitz function f in (4) is often called a critic or a discriminator. The 1-Wasserstein distance is an intuitively meaningful measure that quantifies the minimal cost of transporting mass from one probability distribution to another in optimal transport theory (Villani, 2009). Also, convergence of 1-Wasserstein distance implies the weak convergence of probability measures on bounded domains.

Based on (3), the population objective function for the proposed WGR is:

$$L(g, f) = \lambda_w L_{\mathrm{W}}(g, f) + \lambda_\ell L_{\mathrm{LS}}(g),$$

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

where

$$L_{\mathrm{W}}(g, f) = \mathbb{E}_{(X,\eta)} f(X, g(X, \eta)) - \mathbb{E}_{(X,Y)} f(X, Y), \quad L_{\mathrm{LS}}(g) = \mathbb{E}_{(X,Y)} \| Y - \mathbb{E}_{\eta} g(X, \eta) \|^2.$$

At the population level, the target conditional generator g^* and discriminator f^* are characterized by the minimax problem: $(g^*, f^*) \in \arg\min_g \max_{f \in \mathcal{F}^1_{\mathrm{Lip}}} L(g, f)$.

Suppose we have a random sample $\{(X_i, Y_i), i = 1, 2, \ldots, n\}$ from $P_{X,Y}$, where $n \geq 1$ is the sample size. Let $\{\eta_i, i = 1, 2, \ldots, n\}$ and $\{\eta_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, J\}$ with $J \geq 1$ be random vectors generated independently from P_η. Define $\mathcal{S} = \{(X_i, Y_i, \eta_i), i = 1, 2, \ldots, n\} \cup \{\eta_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, J\}$. We parameterize the generator function g and the discriminator f by neural network functions $g_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{\phi}}$ with parameters (weights and biases) $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. That is, we use neural network functions to approximate the generator and critic functions and optimize the objective function given below over the neural networks to obtain an estimator of g. In addition, since $\mathbb{E}_\eta g_{\boldsymbol{\theta}}(X_i, \eta)$ generally does not have a close-form expression, we approximate it by the sample average $J^{-1} \sum_{j=1}^{J} g_{\boldsymbol{\theta}}(X_i, \eta_{ij})$. Then, the empirical objective function for estimating $(\boldsymbol{\theta}, \boldsymbol{\phi})$ is

$$\widehat{L}(g_{\boldsymbol{\theta}}, f_{\boldsymbol{\phi}}) = \lambda_w \widehat{L}_{\mathrm{W}}(g_{\boldsymbol{\theta}}, f_{\boldsymbol{\phi}}) + \lambda_\ell \widehat{L}_{\mathrm{LS}}(g_{\boldsymbol{\theta}}), \tag{5}$$

where

$$\widehat{L}_{\mathrm{W}}(g_{\boldsymbol{\theta}}, f_{\boldsymbol{\phi}}) = \frac{1}{n} \sum_{i=1}^{n} \left\{ f_{\boldsymbol{\phi}}(X_i, g_{\boldsymbol{\theta}}(X_i, \eta_i)) - f_{\boldsymbol{\phi}}(X_i, Y_i) \right\},$$

$$\widehat{L}_{\mathrm{LS}}(g_{\boldsymbol{\theta}}) = \frac{1}{n} \sum_{i=1}^{n} \left\| Y_i - \frac{1}{J} \sum_{j=1}^{J} g_{\boldsymbol{\theta}}(X_i, \eta_{ij}) \right\|^2.$$

Let $(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}})$ be a solution to the minimax problem

$$(\hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\phi}}) = \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} \widehat{L}(g_{\boldsymbol{\theta}}, f_{\boldsymbol{\phi}}). \tag{6}$$

Then, the estimated conditional generator is $\hat{g}(x, \eta) = g_{\hat{\boldsymbol{\theta}}}(x, \eta)$ and the estimated regression function is obtained by taking the expectation of $\hat{g}(x, \eta)$ with respect to η, that is, $\hat{g}(x) = \mathbb{E}_\eta \hat{g}(x, \eta)$. Since there is no analytical expression for the expectation $\mathbb{E}_\eta \hat{g}(x, \eta)$, we approximate it using an empirical average based on a random sample $\{\eta'_1, \ldots, \eta'_{J'}\}$ from P_η with $J' \geq 1$, which is $\hat{g}(x) \approx (1/J') \sum_{j=1}^{J'} \hat{g}(x, \eta'_j)$. This gives the estimated regression function.

Note that for a given $x \in \mathcal{X}$, $\{\hat{g}(x, \eta'_j), j = 1, \ldots, J'\}$ are approximately distributed as $P_{Y|X=x}$. We can use $\{\hat{g}(x, \eta'_j), j = 1, \ldots, J'\}$ to explore any aspects of $P_{Y|X=x}$ that we are interested in such as its higher moments and quantiles.

2.3 Limitations of conditional Wasserstein generative adversarial network for regression tasks

In this subsection, we discuss the necessity to impose the L_2-regularization term, i.e. the least-square loss in (5), in the estimation of the regression function.

In the absence of the L_2-regularization term $\lambda_\ell \widehat{L}_{\mathrm{LS}}(g_{\boldsymbol{\theta}})$ in (5), one can estimate the conditional generator by minimizing $\max_{\boldsymbol{\phi}} \widehat{L}_{\mathrm{W}}(g_{\boldsymbol{\theta}}, f_{\boldsymbol{\phi}})$ over $\boldsymbol{\theta}$ [i.e. conditional Wasserstein generative adversarial network (cWGAN)], and estimate the regression function based on the estimated conditional generator. However, the original design of the cWGAN focuses on generating data that 'look like' the real data, rather than exploring the underlying relationships between covariate

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

and response variables, the goal of regression tasks. To shed more light on this, we provide an illustration below.

For simplicity, we assume that $(X, Y) \in [0, 1] \times [0, 1]$ and rewrite the minimax problem of the standard cWGAN as

$$(\bar{g}, \bar{f}) = \arg\min_{g \in \mathcal{G}} \max_{f \in \mathcal{F}^1_{\text{Lip}}} \frac{1}{n} \sum_{i=1}^{n} \{ f(X_i, g(X_i, \eta_i)) - f(X_i, Y_i) \},$$

where $\mathcal{G}$ is a predetermined function class. That is, $\bar{g}$ makes the worst-case empirical objective function over all 1-Lipschitz functions as small as possible. We will consider some special 1-Lipschitz functions in this illustration. Observe that for any nonnegative integers a and b, the functions $f(x, y) = y$, $f(x, y) = xy/2$, $f(x, y) = y^2/2$, $f(x, y) = x^a y^b/(a + b)$ are 1-Lipschitz functions. Then, under mild conditions,

(i) (*marginal first moment matching*) when $f(x, y) = y$, we have $(1/n)\sum_{i=1}^{n} \bar{g}(X_i, \eta_i) - (1/n)\sum_{i=1}^{n} Y_i = o_p(1)$ for large n, indicating that solving the minimax problem of the cWGAN is essentially matching the first moment of Y;

(ii) (*marginal or joint moment matching*) likewise, when $f(x, y) = y^2/2$, $f(x, y) = xy/2$ or $f(x, y) = x^a y^b/(a + b)$, we have $(1/n)\sum_{i=1}^{n} \bar{g}^2(X_i, \eta_i) - (1/n)\sum_{i=1}^{n} Y_i^2 = o_p(1)$, $(1/n)\sum_{i=1}^{n} X_i \bar{g}(X_i, \eta_i) - (1/n)\sum_{i=1}^{n} X_i Y_i = o_p(1)$ or $(1/n)\sum_{i=1}^{n} X_i^a \bar{g}^b(X_i, \eta_i) - (1/n)\sum_{i=1}^{n} X_i^a Y_i^b = o_p(1)$ for large n, respectively. This implies that solving the minimax problem of the cWGAN is basically matching the second moment of Y, the first joint moments or other relevant joint moment of Y and X.

Here, $o_p(1)$ denotes the convergence in probability. The above illustration tells that, when $f(x, y)$ is any polynomial function, the cWGAN only enforces marginal or joint moment matching, but not conditional moment matching. Meanwhile, the Stone–Weierstrass theorem guarantees that every continuous function on a compact set can be well approximated by a polynomial function. In other words, there exists a polynomial function $\tilde{f}$ that can approximate $\bar{f}$ arbitrarily well. With this view, the standard cWGAN cannot cater conditional moment matching, making it improper for regression tasks. Other variants of conditional GANs suffer from similar problems (Zhou et al., 2022).

Next, we will provide the insight into how the regularization term enforces conditional mean matching. To see this, define $\epsilon_i = Y_i - \mathbb{E}(Y_i \mid X_i)$ for $i = 1, \ldots, n$, where ϵ_i satisfies $\mathbb{E}(\epsilon_i \mid X_i) = 0$. For the sake of simplicity, we assume that $\mathbb{E}\epsilon_i^2 < \infty$. Notice that the regularization term

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} \left\| Y_i - \frac{1}{J}\sum_{j=1}^{J} g(X_i, \eta_{ij}) \right\|^2 \\
&\quad = \frac{1}{n}\sum_{i=1}^{n} \left\| \mathbb{E}(Y_i \mid X_i) + \epsilon_i - \frac{1}{J}\sum_{j=1}^{J} g(X_i, \eta_{ij}) \right\|^2 \\
&\quad = \frac{1}{n}\sum_{i=1}^{n} \left\| \mathbb{E}(Y_i \mid X_i) - \frac{1}{J}\sum_{j=1}^{J} g(X_i, \eta_{ij}) \right\|^2 + \frac{1}{n}\sum_{i=1}^{n} \epsilon_i^2 \\
&\qquad - \frac{2}{n}\sum_{i=1}^{n} \epsilon_i \left\{ \mathbb{E}(Y_i \mid X_i) - \frac{1}{J}\sum_{j=1}^{J} g(X_i, \eta_{ij}) \right\} \\
&=: \Pi_1 + \Pi_2 + \Pi_3,
\end{aligned}$$

where $\Pi_3 = o_p(1)$ for large n by law of large numbers. Heuristically, minimizing the regularization term over g is approximately equivalent to minimizing Π_1 over g, which will force $J^{-1}\sum_{j=1}^{J} g(X_i, \eta_{ij})$ to be close to $\mathbb{E}(Y_i \mid X_i)$ for each X_i. The effectiveness of this L_2 regularization

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

term for regression tasks depends on the weights $(\lambda_\ell, \lambda_w)$. A detailed discussion on the influence of $(\lambda_\ell, \lambda_w)$ is given in Section 4.

Numerically, the L_2 regularization term also plays a role in alleviating the instability often encountered during the training of WGAN (Gulrajani et al., 2017). It is known that training WGAN is to find a Nash equilibrium to a two-player game, with each player trying to minimize its cost function. The associated optimization problem is often solved by the (stochastic) gradient descent on the cost function of each player simultaneously. Nevertheless, there is no guarantee of convergence. Feature matching (Salimans et al., 2016) was proposed to address the instability of GANs by specifying a new objective for the generator that prevents overtraining of the current discriminator. Specifically, the objective function in Salimans et al. (2016) enforces the generator to generate data that match the first-order feature statistics of real data. In our work, imposing the L_2 regularization term in (5) bears the same rationale as feature matching, i.e. encouraging the convergence of cWGAN's training process and meanwhile enforcing the conditional mean matching for regression tasks.

3 Implementation

In this section, we present the details for implementing WGR. We first describe the neural networks used in the approximation of g and f. We then present the computational algorithm we implemented in detail.

3.1 Rectified linear unit feedforward neural networks

We first give a brief description of feedforward neural networks (FNN) with the rectified linear unit (ReLU) activation function. The ReLU function is denoted by $\sigma(x) := \max(x, 0)$ for each component of x if x is a vector. A neural network can be expressed as a composite function $\zeta(x) = \mathcal{L}_H \circ \sigma \circ \mathcal{L}_{H-1} \circ \sigma \circ \cdots \circ \sigma \circ \mathcal{L}_1 \circ \sigma \circ \mathcal{L}_0(x)$, $x \in \mathbb{R}^{p_0}$, where $\mathcal{L}_i(x) = W_i x + b_i$ with a weight matrix $W_i \in \mathbb{R}^{p_{i+1} \times p_i}$ and bias vector $b_i \in \mathbb{R}^{p_{i+1}}$ in the ith linear transformation, and p_i is the width of the ith layer, $i = 0, 1, \ldots, H$. The width and depth of the network are described by $W = \max\{p_1, \ldots, p_H\}$ and H, respectively. For simplicity, we use $\mathcal{NN}(p_0, p_{H+1}, W, H)$ to denote the neural networks with input dimension p_0, output dimension p_{H+1}, width at most W and depth at most H. We now specify the function classes below:

- The generator network class $\mathcal{G}$: Let $\mathcal{G} \equiv \mathcal{NN}(d + m, q, W_\mathcal{G}, H_\mathcal{G})$ be a class of ReLU-activated FNNs $g_{\boldsymbol{\theta}} : \mathbb{R}^{d+m} \to \mathbb{R}^q$, with parameter $\boldsymbol{\theta}$, width $W_\mathcal{G}$, depth $H_\mathcal{G}$.
- The discriminator network class $\mathcal{D}$: Let $\mathcal{D} \equiv \mathcal{NN}(d + q, 1, W_\mathcal{D}, H_\mathcal{D}) \cap \mathrm{Lip}(\mathcal{X} \times \mathcal{Y}; K_\mathcal{D})$ be a class of ReLU-activated FNNs, $f_{\boldsymbol{\phi}} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, with parameter $\boldsymbol{\phi}$, width $W_\mathcal{D}$, and depth $H_\mathcal{D}$, where for some $K_\mathcal{D} > 0$, $\mathrm{Lip}(\mathcal{X} \times \mathcal{Y}; K_\mathcal{D})$ is a class of Lipschitz functions defined below.

  For any function $f : \Omega \to \mathbb{R}$, the Lipschitz constant of f is denoted by

$$\mathrm{Lip}(f) = \sup_{x,y \in \Omega, x \neq y} \frac{|f(x) - f(y)|}{\|x - y\|}.$$

  For a given $0 < C < \infty$, denote $\mathrm{Lip}(\Omega; C)$ as the set of all functions $f : \Omega \mapsto \mathbb{R}$ with $\mathrm{Lip}(f) \leq C$. Hence, $\mathcal{F}^1_{\mathrm{Lip}}$ defined in (4) is $\mathrm{Lip}(\mathcal{X} \times \mathcal{Y}; 1)$.

3.2 Computation

For training the conditional distribution generator $g_{\boldsymbol{\theta}}$ and the discriminator $f_{\boldsymbol{\phi}}$, we use the leaky rectified linear unit (leaky ReLU) as the activation function in $g_{\boldsymbol{\theta}}$ and $f_{\boldsymbol{\phi}}$. The training algorithm is presented in Algorithm 1.

To constrain the discriminator $f_{\boldsymbol{\phi}}$ to the class of 1-Lipschitz functions, a gradient penalty is used in (7), which is a modified version of the algorithm proposed by Gulrajani et al. (2017). The difference is that we evaluate the gradients at the sample points in the penalty, instead of using generated intermediate points. Another approach that we have tried to enforce the Lipschitz constraint is the clipping method (Arjovsky et al., 2017), which also produces acceptable results, but seems to be less stable than the penalty method described in Algorithm 1. We suggest using traversal to select the tuning parameters λ_ℓ and λ_w that control the trade-off

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Algorithm 1 WGR algorithm

Require: (a) dataset $\{(X_i, Y_i)\}_{i=1}^n$; (b) minibatch size $v \le n$; (c) an m-dimensional continuous distribution P_η; (d) J, the size of noise vector η; (e) tuning parameters λ_ℓ and λ_w; (f) penalty parameter λ.

1: **for** number of training iterations **do**

2: Sample v pairs $\{(X_{bi}, Y_{bi})\}_{i=1}^v$ from $\{(X_i, Y_i)\}_{i=1}^n$.

3: Sample i.i.d. $\{\eta_{ij}, i = 1, 2, \ldots, n, j = 0, 1, \ldots, J\}$ from P_η.

4: Update the discriminator f_ϕ by ascending its stochastic gradient:

$$\nabla_\phi \left\{ \frac{1}{v} \sum_{i=1}^{v} f_{\boldsymbol{\phi}}\left(X_{bi}, g_{\boldsymbol{\theta}}\left(X_{bi}, \eta_{i0}\right)\right) - f_{\boldsymbol{\phi}}\left(X_{bi}, Y_{bi}\right) - \lambda\left(\left\|\nabla_{(x,y)} f_{\boldsymbol{\phi}}\left(X_{bi}, Y_{bi}\right)\right\| - 1\right)^2 \right\}. \tag{7}$$

5: Update the generator $g_{\boldsymbol{\theta}}$ by descending its stochastic gradient:

$$\nabla_\theta \left[\frac{\lambda_\ell}{v} \sum_{i=1}^{v} \left\{ Y_{bi} - \frac{1}{J} \sum_{j=1}^{J} g_{\boldsymbol{\theta}}(X_{bi}, \eta_{ij}) \right\}^2 + \frac{\lambda_w}{v} \sum_{i=1}^{v} f_{\boldsymbol{\phi}}\left(X_{bi}, g_{\boldsymbol{\theta}}\left(X_{bi}, \eta_{i0}\right)\right) \right].$$

6: **end for**

between the distribution discrepancy and the prediction error. In Section 5, we demonstrate the effectiveness of this algorithm in various numerical experiments. However, we do not have a theoretical analysis of its convergence behaviour and we leave this as an open problem for future research.

Remark 2 We provide a simple data-driven method to determine m, the dimension of η, in practice. Specifically, we propose to minimize the following criterion over m within a candidate set:

$$\sum_{i=1}^{n} \left\| Y_i - \frac{1}{J} \sum_{j=1}^{J} \hat{g}(X_i, \eta_{ij}) \right\|^2 + (d + m) \log n,$$

where the first term measures the goodness of fit for nonparametric mean regression, and the term $(d + m) \log n$ is a penalty on model complexity. Intuitively, this is a Bayesian Information Criterion (BI)-type criterion (Schwarz, 1978) for model selection. Our numerical experiments in the online supplementary material provide supporting evidence that the proposed selector works reasonably well. A rigorous justification of its validity, however, would be challenging and merits further study.

4 Error analysis and convergence

To evaluate the statistical performance of WGR, we will establish the bounds for the prediction error $\mathbb{E}_{\mathcal{S}}\{\mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2\}$ and the 1-Wasserstein distance $\mathbb{E}_{\mathcal{S}}\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\}$, respectively. Here, $\mathbb{E}_X$ and $\mathbb{E}_{\mathcal{S}}$ are the expectation with respect to X and $\mathcal{S} = \{(X_i, Y_i, \eta_i), i = 1, 2, \ldots, n\} \cup \{\eta_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, J\}$, respectively. In this section, we first develop an error decomposition, which decomposes these two estimation errors into approximation errors and stochastic errors of the generator and discriminator. We then derive nonasymptotic error bounds for WGR based on this error decomposition.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

4.1 Error decomposition

We present a high-level description of the error decomposition for WGR. A function class $\mathcal{F}$ is called symmetric if $f \in \mathcal{F}$ implies $-f \in \mathcal{F}$. The integral probability metric (Müller, 1997) between $P_{X,g}$ and $P_{X,Y}$ with respect to a symmetric class $\mathcal{F}$ of real-valued functions on $\mathcal{X} \times \mathcal{Y}$ is defined as

$$d_{\mathcal{F}}(P_{X,g}, P_{X,Y}) = \sup_{f \in \mathcal{F}} \{\mathbb{E}_{(X,\eta)} f(X, g(X, \eta)) - \mathbb{E}_{(X,Y)} f(X, Y)\}.$$

Clearly, $d_{\mathcal{F}^1_{\mathrm{Lip}}}(P_{X,g}, P_{X,Y}) = \mathbb{D}_W(P_{X,g}, P_{X,Y})$. Let $\mathcal{F}^1_B = \{f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}, |f(u) - f(v)| \le \|u - v\|, \forall u, v \in \mathcal{X} \times \mathcal{Y}, \|f\|_\infty \le B\}$ for a constant $0 < B < \infty$ be the class of uniformly bounded 1-Lipschitz functions. If the space $\mathcal{X} \times \mathcal{Y}$ is compact and $B \ge \max_{u,v \in \mathcal{X} \times \mathcal{Y}} \|u - v\|$, then $d_{\mathcal{F}^1_B}(P_{X,g}, P_{X,Y}) = \mathbb{D}_W(P_{X,g}, P_{X,Y})$.

We introduce a new error decomposition method in Lemma 1, which decomposes the estimation errors into approximation errors and stochastic errors of the generator and discriminator.

Lemma 1 Assume that the discriminator network class $\mathcal{D}$ is symmetric and the probability measures of (X, Y) and $(X, g(X, \eta))$ are supported on a compact set $\mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^{d+q}$ for any $g \in \mathcal{G}$. Then, for the WGR estimator defined in (6),

$$\begin{aligned} &\mathbb{E}_{\mathcal{S}}\left\{\lambda_\ell \mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2 + \lambda_w \mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\right\} \\ &\quad \le \lambda_\ell \mathcal{E}_1 + 4\lambda_\ell \mathcal{E}_2 + 2\mathcal{E}_3 + 2\lambda_w \mathcal{E}_4 + 3\lambda_w \mathcal{E}_5 + 3\lambda_w \mathcal{E}_6, \end{aligned} \tag{8}$$

where $\mathcal{S} = \{(X_i, Y_i, \eta_i), i = 1, 2, \ldots, n\} \cup \{\eta_{ij}, i = 1, 2, \ldots, n, j = 1, 2, \ldots, J\}$ and

$$\begin{aligned} \mathcal{E}_1 &:= \mathbb{E}_{\mathcal{S}}\left\{\mathbb{E}_{(X,Y)} \|Y - \mathbb{E}_\eta g^*(X, \eta)\|^2 + \mathbb{E}_{(X,Y)} \|Y - \mathbb{E}_\eta \hat{g}(X, \eta)\|^2 \right. \\ &\quad \left. - \frac{2}{n} \sum_{i=1}^n \|Y_i - \mathbb{E}_\eta \hat{g}(X_i, \eta)\|^2 \right\}, \\ \mathcal{E}_2 &:= \mathbb{E}_{\mathcal{S}}\left\{\sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n \left| \|Y_i - \frac{1}{J} \sum_{j=1}^J g(X_i, \eta_{ij})\|^2 - \|Y_i - \mathbb{E}_\eta g(X_i, \eta)\|^2 \right| \right\}, \\ \mathcal{E}_3 &:= \inf_{g \in \mathcal{G}} \left[\lambda_\ell \mathbb{E}_X \|\mathbb{E}_\eta g(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2 \right. \\ &\quad \left. + \lambda_w \sup_{f \in \mathcal{D}} \{\mathbb{E}_{(X,\eta)} f(X, g(X, \eta)) - \mathbb{E}_{(X,Y)} f(X, Y)\} \right], \\ \mathcal{E}_4 &:= \sup_{h \in \mathcal{F}^1_{B_0}} \inf_{f \in \mathcal{D}} \|h - f\|_\infty \text{ with } B_0 = \max_{u,v \in \mathcal{X} \times \mathcal{Y}} \|u - v\|, \\ \mathcal{E}_5 &:= \mathbb{E}_{\mathcal{S}}\left[\sup_{f \in \mathcal{D}} \left\{\mathbb{E}_{(X,Y)} f(X, Y) - \frac{1}{n} \sum_{i=1}^n f(X_i, Y_i)\right\}\right], \\ \mathcal{E}_6 &:= \mathbb{E}_{\mathcal{S}}\left[\sup_{f \in \mathcal{D}, g \in \mathcal{G}} \left\{\mathbb{E}_{(X,\eta)} f(X, g(X, \eta)) - \frac{1}{n} \sum_{i=1}^n f(X_i, g(X_i, \eta_i))\right\}\right]. \end{aligned}$$

Here, $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_5$, and $\mathcal{E}_6$ are stochastic errors; $\mathcal{E}_3$ and $\mathcal{E}_4$ are approximation errors. Lemma 1 provides a general error decomposition method, which covers the error decomposition inequality in Jiao et al. (2023) for the traditional nonparametric regression as a special case

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

(corresponding to the case that $\lambda_\ell = 1$, $\lambda_w = 0$ and $J = \infty$). It can also be utilized for the error analysis for the cWGAN (corresponding to the case that $\lambda_\ell = 0$, $\lambda_w = 1$). Moreover, when $\lambda_\ell = 0$ and $\lambda_w = 1$, our error decomposition result in (8) is in line with that in Lemma 9 in Huang et al. (2022) for the general GANs. The main difference is that the upper bound of $\mathcal{E}_6$ in (8) depends on the sample size n, while it is determined by the sample size of the generated noise η in Huang et al. (2022). This is the key difference in the theoretical analysis between cWGAN and general WGAN.

4.2 Nonasymptotic error bounds

More notations are needed. Let $\mathbb{N}$ be the set of positive integers and $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. The maximum and minimum of A and B are denoted by $A \vee B$ and $A \wedge B$. Let $\lfloor A \rfloor$ be the largest integer strictly smaller than A and $\lceil A \rceil$ be the smallest integer strictly larger than A. For any $\beta > 0$ and a set $\Omega \subseteq \mathbb{R}^{m+d}$, the Hölder class of functions $\mathcal{H}^\beta(\Omega, B)$ with a constant $0 < B < \infty$ is defined as

$$\mathcal{H}^\beta(\Omega, B) = \left\{ f : \Omega \to \mathbb{R}, \max_{\|\alpha\|_1 \le \lfloor\beta\rfloor} \|D^\alpha f\|_\infty \le B, \max_{\|\alpha\|_1 = \lfloor\beta\rfloor} \sup_{x,y\in\Omega, x\neq y} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{\|x - y\|^r} \le B \right\},$$

where $\partial^\alpha = \partial^{\alpha_1} \cdots \partial^{\alpha_{m+d}}$ with $\alpha = (\alpha_1, \ldots, \alpha_{m+d})^\top \in \mathbb{N}_0^{m+d}$.

The following conditions are needed.

Condition 1 The probability measures of (X, Y) and $(X, g(X, \eta))$ are supported on a compact set $\mathcal{X} \times \mathcal{Y} \subseteq [-B_1, B_1]^{d+q}$ for any $g \in \mathcal{G}$, with a constant $0 < B_1 < \infty$.

Condition 2 The probability measure of η is supported on $\Omega_\eta \subseteq [-B_1, B_1]^m$.

Condition 3 For $g^* = (g_1^*, \ldots, g_q^*)^\top$, $g_k^* \in \mathcal{H}^\beta(\mathcal{X} \times \Omega_\eta, B_1)$, $k = 1, \ldots, q$, where $\beta > 0$.

Condition 4 For any $x \in \mathcal{X}$, there exists a vector $\eta_x \in \Omega_\eta$ such that

$$\|\mathbb{E}_\eta g(x, \eta) - \mathbb{E}_\eta \tilde{g}(x, \eta)\|_1 \le \|g(x, \eta_x) - \tilde{g}(x, \eta_x)\|_1 \quad \forall g, \tilde{g} \in \mathcal{G}.$$

Let W, $\bar{W}$, $\bar{H} \in \mathbb{N}$, which may depend on n. We also make the following assumptions on the network classes $\mathcal{D}$ and $\mathcal{G}$.

ND 1 The discriminator ReLU network class $\mathcal{D} = \mathcal{NN}(d + q, 1, W_\mathcal{D}, H_\mathcal{D}) \cap \mathrm{Lip}(\mathcal{X} \times \mathcal{Y}; K_\mathcal{D})$ has width $W_\mathcal{D} = W^{d+q}\{9(W + 1) + 5(d + q) - 1\}$, depth $H_\mathcal{D} = 3 + 14(d + q)(d + q - 1)$ and Lipschitz constant $K_\mathcal{D} \le 54 B_0 B_1^{d+q} 4^{d+q} (d + q)^{1/2} W^2$.

NG 1 The generator ReLU network class $\mathcal{G} = \mathcal{NN}(m + d, q, W_\mathcal{G}, H_\mathcal{G})$ has width $W_\mathcal{G} = 38q(\lfloor\beta\rfloor + 1)^2 3^{(m+d)} (m + d)^{\lfloor\beta\rfloor+1} \bar{W} \lceil \log_2(8\bar{W}) \rceil$ and depth $H_\mathcal{G} = 21(\lfloor\beta\rfloor + 1)^2 \bar{H} \lceil \log_2(8\bar{H}) \rceil + 2(m + d)$.

Conditions 1–3 require $P_{X,Y}$, $P_{X,g}$, and P_η have a bounded support. Condition 3 is a smoothness condition for g^* in (1). Condition 4 is a technical condition. The upper bound of the Lipschitz constant in ND1 is needed to achieve a small approximation error. More details can be found in Lemma S5 in the online supplementary material.

We will study the nonasymptotic error bounds for the prediction error $\mathbb{E}_\mathcal{S}\{\mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2\}$ and the Wasserstein metric $\mathbb{E}_\mathcal{S}\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\}$, respectively, by bounding the stochastic errors and approximation errors introduced in Section 4.1. To lighten the notations, we

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

define the following two quantities:

$$a := \frac{\beta}{2\beta + \{3(m+d)\} \vee \{2\beta(d+q+1)\}}, \quad b := \frac{3(m+d)}{2[2\beta + \{3(m+d)\} \vee \{2\beta(d+q+1)\}]}.$$

Theorem 1 Suppose that Conditions 1–4 hold and the network parameters of $\mathcal{D}$ and $\mathcal{G}$ satisfy **ND**1 and **NG**1 with $W = \lceil n^a \rceil$, $\bar{W} = \lceil n^b / \log^2 n \rceil$, and $\bar{H} = \lceil \log n \rceil$. Then, for $J \gtrsim n$ and given weights λ_ℓ and λ_w satisfying $0 < \lambda_\ell, \lambda_w < 1$ and $\lambda_\ell + \lambda_w = 1$, we have

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2\right\} \leq C_1 n^{-a} (\log n)^{\frac{2\beta}{m+d} \vee 1},$$

where C_1 is a positive constant independent of n and J.

Theorem 1 establishes a nonasymptotic upper bound for the prediction error of WGR. It implies the consistency of WGR for nonparametric mean regression in the sense that $\mathbb{E}_{\mathcal{S}}\{\mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2\}$ converges to zero as $n \to \infty$ (Jiao et al., 2023; Lugosi & Zeger, 1995; Mielniczuk & Tyrcha, 1993). The convergence rates in Theorem 1 is slightly slower than $O(n^{-2\beta/(2\beta+d)})$ (up to a logarithmic factor), the optimal-minimax rate achieved by deep least squares nonparametric regression (Bauer & Kohler, 2019; Jiao et al., 2023; Schmidt-Hieber, 2020). This is because in nonparametric regression, there is no distributional matching constraint, thus the noise vector η is not involved and a faster convergence rate can be achieved. In our proposed framework, we are not only interested in estimating the mean regression function, but also the conditional generator, which involves the noise vector from a noise distribution. This increases the dimensionality of the problem and results in a slower convergence rate. We next establish a nonasymptotic error bound for the Wasserstein distance $\mathbb{E}_{\mathcal{S}}\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\}$.

Theorem 2 Suppose that the conditions of Theorem 1 are satisfied. Then, for $J \gtrsim n$ and given weights λ_ℓ and λ_w satisfying $0 \leq \lambda_\ell < 1$, $0 < \lambda_w \leq 1$ and $\lambda_\ell + \lambda_w = 1$, we have

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\right\} \leq C_2 n^{-a} (\log n)^{\frac{2\beta}{m+d} \vee 1},$$

where C_2 is a positive constant independent of n and J.

The nonasymptotic error bounds in Theorems 1 and 2 are established for fixed positive weights λ_ℓ and λ_w. In this case, we obtain the same convergence rate of the prediction error $\mathbb{E}_{\mathcal{S}}\{\mathbb{E}_X \|\mathbb{E}_\eta \hat{g}(X, \eta) - \mathbb{E}_\eta g^*(X, \eta)\|^2\}$ and the Wasserstein metric $\mathbb{E}_{\mathcal{S}}\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\}$. When $\lambda_\ell = 0$ and $\lambda_w = 1$, Theorem 2 can also imply a nonasymptotic error bound for the standard cWGAN under the Wasserstein metric. Note that the cWGAN is involved in our proposed procedure, thus the joint stochastic error of the discriminator and generator is affected by the dimension of the noise vector η, resulting in a slower convergence rate compared with the one in Theorem 5 in Huang et al. (2022) for GAN estimators. The result in Theorem 2 implies the weak convergence of $(X, \hat{g}(X, \eta))$ to (X, Y), which leads to the next corollary regarding the Wasserstein metric. Next, we give nonasymptotic error bounds for the Wasserstein distance $\mathbb{E}_{\mathcal{S}}\mathbb{E}_X\{\mathbb{D}_W(P_{\hat{g}|X}, P_{Y|X})\}$ in Corollary 1, indicating that WGR can accurately learn the target conditional distribution.

Corollary 1 Suppose that the conditions of Theorem 1 are satisfied. For $J \gtrsim n$ and given weights λ_ℓ and λ_w satisfying $0 \leq \lambda_\ell < 1$, $0 < \lambda_w \leq 1$ and $\lambda_\ell + \lambda_w = 1$, we have

$$\mathbb{E}_{\mathcal{S}}\mathbb{E}_X\left\{\mathbb{D}_W(P_{\hat{g}|X}, P_{Y|X})\right\} \leq C_2 n^{-a} (\log n)^{\frac{2\beta}{m+d} \vee 1},$$

where C_2 is a positive constant independent of n and J.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Remark 3 The results in Corollary 1 enable us to construct prediction intervals based on the learned conditional generator directly. In existing literature, effective methods for constructing prediction intervals include conformal prediction (Candès et al., 2023; Lei et al., 2013; Romano et al., 2019; Vovk et al., 1999), and Bayesian methods (Fahrmeir & Lang, 2001; Hilton et al., 2019; Klein et al., 2015; West et al., 1985). Conformal prediction is a distribution-free framework that provides prediction intervals with guaranteed nonasymptotic coverage. Bayesian methods construct prediction intervals based on the posterior distribution. In the numerical studies, we will compare the performance of WGR to these methods in terms of the length of prediction interval and the coverage probability (CP) in Section 5.

Corollary 2 Suppose the conditions of Theorem 1 are satisfied. When $\lambda_\ell = 0$ and $\lambda_w = 1$,

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{E}_X\|\mathbb{E}_\eta \hat{g}(X,\eta) - \mathbb{E}_\eta g^*(X,\eta)\|^2\right\} \le C_3 n^{-a}(\log n)^{\frac{2\beta}{m+d}\vee 1},$$

where C_3 is a positive constant independent of n and J.

Corollary 2 establishes a nonasymptotic upper bound for the prediction error of the standard cWGAN. Theorem 1 and Corollary 2 tell that, with fixed positive weights λ_ℓ and λ_w, the prediction error of our proposed WGR and the cWGAN have the same convergence rate.

Theorem 3 Suppose that Conditions 1–4 hold and the network parameters of $\mathcal{D}$ and $\mathcal{G}$ satisfy **ND**1 and **NG**1 with $W = \lceil n^a \rceil$, $\bar{W} = \lceil n^b/\log^2 n \rceil$ and $\bar{H} = \lceil \log n \rceil$. Then, for $\lambda_\ell > 0$, $\lambda_w > 0$ satisfying $\lambda_\ell + \lambda_w = 1$ and $\lambda_w = O(n^{-1/(d+q+2)})$, when $2\beta(d+q+1) \ge 3(m+d)+\beta$ and $J \gtrsim n^{\{3(m+d)+6\beta\}/\{4\beta(d+q+2)\}}$, we have

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{E}_X\|\mathbb{E}_\eta \hat{g}(X,\eta) - \mathbb{E}_\eta g^*(X,\eta)\|^2\right\} \le C_4 n^{-\frac{3}{2(d+q+2)}}(\log n)^{\frac{2\beta}{m+d}\vee 2},$$

where C_4 is a positive constant independent of n and J. Moreover, as $n \to \infty$,

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{D}_W(P_{X,\hat{g}}, P_{X,Y})\right\} \to 0.$$

Theorem 3 establishes a nonasymptotic error bound for varying weights λ_ℓ and λ_w that can diverge with the sample size n. Theorem 3 tells that, when $2\beta(d+q+1) \ge 3(m+d)+\beta$ and $\lambda_w = O(n^{-1/(d+q+2)})$, the convergence rate $n^{-3/\{2(d+q+2)\}}$ (up to a logarithmic factor) of the prediction error is faster than the rate n^{-a} in Theorem 1 with fixed positive λ_w, since $3/\{2(d+q+2)\}$ is greater than $a = 1/\{2(d+q+2)\}$. The faster convergence rate in Theorem 3 shows the superiority of our proposed WGR over the standard cWGAN in estimating the conditional mean function. It also provides some theoretical guidance on the selection of $(\lambda_\ell, \lambda_w)$: for more accurate estimation of the regression function, the value of $(\lambda_\ell, \lambda_w)$ should be determined by the sample size n and the dimension of (X, Y). Meanwhile, Theorem 3 also implies that $(X, \hat{g}(X,\eta))$ weakly converges to (X, Y) as $n \to \infty$.

As stated in Section 2, in practice, we approximate $\mathbb{E}_\eta \hat{g}(X,\eta)$ by the empirical average $J'^{-1}\sum_{j=1}^{J'} \hat{g}(x,\eta'_j)$ based on a random sample $\{\eta'_1, \ldots, \eta'_{J'}\}$ sampled from P_η. For completion, we present a nonasymptotic error bound regarding $J'^{-1}\sum_{j=1}^{J'} \hat{g}(x,\eta'_j)$ below, a direct consequence of Theorem 3. Notably, since J' is user-determined, setting a large J' would be beneficial in practical scenarios.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Corollary 3 Suppose that the conditions of Theorem 3 are satisfied. Then, for $\lambda_\ell > 0$, $\lambda_w > 0$ satisfying $\lambda_\ell + \lambda_w = 1$ and $\lambda_w = O(n^{-1/(d+q+2)})$, when $2\beta(d+q+1) \geq 3(m+d) + \beta$ and $J \gtrsim n^{\{3(m+d)+6\beta\}/\{4\beta(d+q+2)\}}$, we have

$$\mathbb{E}_{\mathcal{S}}\left\{\mathbb{E}_{X\cup\{\eta'_j\}_{j=1}^{J'}}\left\|\frac{1}{J'}\sum_{j=1}^{J'}\hat{g}(X,\eta'_j) - \mathbb{E}_{\eta}g^*(X,\eta)\right\|^2\right\} \leq C_4 n^{-\frac{3}{2(d+q+2)}}(\log n)^{\frac{2\beta}{m+d}\vee 1} + C_5 J'^{-1},$$

where $\mathbb{E}_{X\cup\{\eta'_j\}_{j=1}^{J'}}$ is the expectation with respect to X and $\{\eta'_1, \ldots, \eta'_{J'}\}$, C_4, C_5 are positive constants independent of n, J and J'.

5 Numerical studies

In this section, numerical experiments including simulation studies and real data examples are conducted to assess the performance of the proposed method.

For comparison, we compute the deep nonparametric least squares regression (DNLS), Bayesian Neural Networks-based method (BNN) (Jospin et al., 2022) and cWGAN (Arjovsky et al., 2017). For the numerical studies in this section, we take the distribution of η to be $N(\mathbf{0}, \mathbf{I}_m)$ for simplicity, and the dimension m is determined by the data-driven selector introduced in Remark 2. Additional results are presented in the online supplementary material, including the details of tuning parameters selection and neural network architecture, four additional simulation models and investigation into the influence of sample size, different choice of noise distribution, its dimensionality m, and its size J, J'.

5.1 Simulation studies

We conduct simulations on four models to evaluate the performance of WGR. For models M1–M2, we generate data with a univariate response Y, and for models M3–M4, with a multidimensional response Y. In all models, covariates $X = (x_1, \ldots, x_d)^\top$ follow the multivariate standard normal distribution. For models M1–M2, we consider both low-dimensional ($d = 5$) and high-dimensional ($d = 100$) cases.

M1 A nonlinear regression model with additive heteroscedastic error: $Y = x_1^2 + \exp(x_2 + x_3/3) + x_4 - x_5 + (0.5 + x_2^2/2 + x_5^2/2)\varepsilon$, $\varepsilon \sim N(0, 1)$, $X \perp \varepsilon$.

M2 A single-index model with an additive error: $Y = (X^\top\beta)^2 + \sin(|X^\top\beta|) + 2\cos(\varepsilon)$, where $\varepsilon \sim N(0, 1)$, $X \perp \varepsilon$, $\beta = (1, 1, -1, -1, 1)^\top$ for $d = 5$, or $\beta = (1, 1, -1, -1, 1, 0_{95}^\top)^\top$ for $d = 100$.

M3 Involute model: $Y_1 = 2X + U\sin(2U) + \varepsilon_1$, $Y_2 = 2X + U\cos(2U) + \varepsilon_2$, where $X \sim N(0, 1)$, $U \sim \text{Uniform}(0, 2\pi)$, $\varepsilon_1 \sim N(0, 0.4^2)$, $\varepsilon_2 \sim N(0, 0.4^2)$ and X, U, ε_1, ε_2 are mutually independent.

M4 A Gaussian mixture model: $Y_1 = X + \varepsilon_1$, $Y_2 = X + \varepsilon_2$, where $X \perp (\varepsilon_1, \varepsilon_2)$, $X \sim N(0, 1)$, $\varepsilon_i \sim \frac{1}{3}N(-2, 0.25^2) + \frac{1}{3}N(0, 0.25^2) + \frac{1}{3}N(2, 0.25^2)$, $i = 1, 2$.

For evaluation purpose, for any estimator for the conditional generator $\hat{g}$, under a validation or testing data set $\{(Y'_k, X'_k, \eta'_{k,j}) : j = 1, \ldots, J'\}_{k=1}^K$, we define $\hat{E}(Y \mid X = X'_k) := (1/J')\sum_{j=1}^{J'}\hat{g}(X'_k, \eta'_{kj})$ as the estimator for the conditional mean $E(Y \mid X = X'_k)$, $\hat{SD}(Y \mid X = X'_k) := [(1/J')\sum_{j=1}^{J'}\{\hat{g}(X'_k, \eta'_{kj}) - \hat{E}(Y \mid X = X'_k)\}^2]^{1/2}$ as the estimator for the conditional standard deviation $SD(Y \mid X = X'_k)$, and $\hat{F}^{-1}_{Y\mid X}(\tau \mid X = X'_k)$ as the sample τth conditional quantile calculated via Monte Carlo methods for estimating the τth conditional quantile $F^{-1}_{Y\mid X}(\tau \mid X = X'_k)$. We also

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

compute the L_1 and L_2 errors and the mean squared error (MSE) for the conditional mean, the conditional standard deviation and the τth conditional quantile:

$$L_1 = \frac{1}{K}\sum_{k=1}^{K} \|Y'_k - \hat{E}(Y \mid X = X'_k)\|,$$

$$L_2 = \frac{1}{K}\sum_{k=1}^{K} \|Y'_k - \hat{E}(Y \mid X = X'_k)\|^2,$$

$$\text{MSE(mean)} = \frac{1}{K}\sum_{k=1}^{K} \{\hat{E}(Y \mid X = X'_k) - E(Y \mid X = X'_k)\}^2,$$

$$\text{MSE(sd)} = \frac{1}{K}\sum_{k=1}^{K} \{\hat{SD}(Y \mid X = X'_k) - SD(Y \mid X = X'_k)\}^2,$$

$$\text{MSE}(\tau) = \frac{1}{K}\sum_{k=1}^{K} \{\hat{F}^{-1}_{Y|X}(\tau \mid X = X'_k) - F^{-1}_{Y|X}(\tau \mid X = X'_k)\}^2,$$

where the quantile level τ is taken as 0.05, 0.25, 0.50, 0.75, 0.95. For models with multi-dimensional responses (models M3–M4), we evaluate the performance of each dimension separately. For comparison, we compute the estimates for $E(Y \mid X = X'_k)$, $SD(Y \mid X = X'_k)$, $F^{-1}_{Y|X}(\tau \mid X = X'_k)$ via Monte Carlo methods for BNN; but $SD(Y \mid X = X'_k)$, $F^{-1}_{Y|X}(\tau \mid X = X'_k)$ are not computable for DNLS, as DNLS only estimates the conditional mean $E(Y \mid X)$.

The training, validation and testing data size are 5,000, 1,000, and 1,000, respectively. We repeat the simulations 100 times. Table 1 summaries the average L_1 error, L_2 error, MSE(mean), and MSE(sd). Table 2 reports the average MSE(τ) for different τ. For each criterion, the empirical standard errors across 100 replications are in parentheses. In most settings, WGR performs comparably to DNLS and better than cWGAN and BNN in terms of L_1 error, L_2 error and MSE(mean), indicating the competitiveness of our method on conditional mean estimation. Furthermore, WGR gives smaller MSE(sd) and MSE(τ) compared to cWGAN and BNN in most models, demonstrating better performance in distribution matching. Note that WGR enjoys a smaller standard deviation for most criteria than cWGAN and BNN, which indicates that our method is more stable.

We believe that the improvement in distribution matching is because the L_2-regularization term is helpful in mitigating the instability in the training of WGAN, as discussed in Section 2.3. To see this, in Figure 1, we visualize the quality of the conditional samples and the conditional density estimation given a random realization of X as the number of epochs increases. It shows that, unlike cWGAN, the conditional distribution generated by WGR starts to approach the target distribution after only 5 or 10 epochs of training, that is, WGR can effectively capture the underlying conditional distributions. Moreover, the last column of Figure 1 shows that WGR provides more accurate conditional distribution estimation compared to cWGAN after 200 epochs.

5.2 Real data examples

We apply the proposed WGR to five real datasets: CT slices (Graf et al., 2011), UJIndoorLoc (Torres-Sospedra et al., 2014), MNIST (Modified National Institute of Standards and Technology) (LeCun et al., 2010), CIFAR10 (Krizhevsky, 2009), and STL10 (Coates et al., 2011). Due to space limitations, the results for CIFAR10 and STL10 are provided in the online supplementary material.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Table 1. Comparison of WGR with DNLS, BNN and cWGAN for M1–M4

	d	Y	Method	L_1	L_2	MSE(mean)	MSE(sd)
M1	5	Y	DNLS	**1.24**(0.04)	3.61(0.52)	0.28(0.14)	–
			BNN	1.27(0.05)	3.63(0.48)	0.40(0.15)	0.68(0.15)
			cWGAN	1.28(0.05)	3.72(0.43)	0.48(0.15)	0.38(0.22)
			WGR	**1.24**(0.05)	**3.46**(0.34)	**0.26**(0.06)	**0.33**(0.09)
	100	Y	DNLS	1.64(0.07)	5.64(0.79)	2.30(0.38)	–
			BNN	1.69(0.12)	6.00(0.88)	2.69(0.65)	1.59(0.29)
			cWGAN	1.80(0.08)	6.37(0.70)	3.15(0.56)	0.94(0.11)
			WGR	**1.60**(0.07)	**5.46**(0.54)	**2.26**(0.51)	**0.38**(0.12)
M2	5	Y	DNLS	0.73(0.02)	0.89(0.07)	0.10(0.06)	–
			BNN	**0.72**(0.07)	1.00(0.36)	0.35(0.21)	0.10(0.01)
			cWGAN	0.75(0.03)	0.97(0.10)	0.16(0.08)	0.08(0.03)
			WGR	**0.72**(0.02)	**0.86**(0.04)	**0.06**(0.02)	**0.07**(0.04)
	100	Y	DNLS	1.25(0.10)	3.07(0.71)	2.12(0.52)	–
			BNN	1.25(0.09)	3.17(0.79)	2.37(0.79)	0.35(0.25)
			cWGAN	1.29(0.14)	3.26(0.84)	2.46(0.80)	0.43(0.18)
			WGR	**1.17**(0.09)	**2.61**(0.46)	**1.81**(0.47)	**0.15**(0.06)
M3	1	Y_1	DNLS	2.04(0.04)	**6.40**(0.22)	0.07(0.04)	-
			BNN	2.05(0.03)	6.47(0.16)	0.11(0.05)	2.93(0.24)
			cWGAN	2.05(0.05)	6.49(0.25)	0.14(0.09)	0.12(0.06)
			WGR	**2.03**(0.01)	6.47(0.13)	**0.06**(0.04)	**0.10**(0.08)
		Y_2	DNLS	**2.03**(0.03)	**6.77**(0.19)	0.06(0.04)	–
			BNN	2.06(0.04)	6.95(0.19)	0.11(0.05)	3.12(0.01)
			cWGAN	2.04(0.03)	6.80(0.19)	0.19(0.21)	0.06(0.06)
			WGR	2.05(0.01)	6.86(0.07)	**0.05**(0.03)	**0.05**(0.04)
M4	1	Y_1	DNLS	**1.40**(0.02)	2.74(0.04)	**0.03**(0.01)	-
			BNN	1.41(0.02)	2.76(0.05)	0.04(0.00)	0.64(0.21)
			cWGAN	**1.40**(0.01)	**2.73**(0.03)	**0.03**(0.02)	0.09(0.04)
			WGR	**1.40**(0.01)	2.76(0.02)	0.05(0.01)	**0.02**(0.01)
		Y_2	DNLS	1.41(0.01)	**2.74**(0.03)	**0.02**(0.01)	–
			BNN	1.42(0.02)	2.78(0.05)	0.04(0.01)	0.58(0.11)
			cWGAN	1.41(0.01)	**2.74**(0.04)	0.03(0.01)	0.11(0.05)
			WGR	**1.40**(0.01)	**2.74**(0.03)	0.05(0.02)	**0.02**(0.01)

Note. The corresponding standard errors are given in parentheses. The smallest MSEs are in bold font. WGR = Wasserstein generative regression; DNLS = deep nonparametric least squares regression; BNN = Bayesian Neural Networks; cWGAN = conditional Wasserstein generative adversarial network; MSE = mean squared error.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Table 2. Comparison of WGR with BNN and cWGAN for M1–M4: MSE at quantile levels $\tau \in \{0.05, 0.25, 0.50, 0.75, 0.95\}$

	d	Y	Method	MSE(0.05)	MSE(0.25)	MSE(0.50)	MSE(0.75)	MSE(0.95)
M1	5	Y	BNN	1.99(0.40)	0.70(0.18)	0.44(0.15)	0.76(0.17)	2.13(0.44)
			cWGAN	1.86(0.21)	0.94(0.26)	0.85(0.26)	1.00(0.21)	2.59(0.52)
			WGR	**1.14**(0.34)	**0.41**(0.10)	**0.27**(0.06)	**0.43**(0.10)	**1.12**(0.37)
	100	Y	BNN	6.32(1.24)	3.35(0.74)	2.75(0.65)	3.56(0.69)	6.77(1.09)
			cWGAN	4.99(0.51)	2.63(0.24)	2.21(0.22)	2.79(0.36)	5.41(0.65)
			WGR	**2.76**(0.61)	**2.27**(0.30)	**2.19**(0.21)	**2.61**(0.24)	**3.64**(0.53)
M2	5	Y	BNN	**1.04**(0.36)	0.35(0.28)	0.35(0.24)	0.35(0.23)	0.68(0.32)
			cWGAN	0.91(0.23)	0.24(0.09)	0.21(0.09)	0.22(0.11)	0.30(0.17)
			WGR	**1.04**(0.29)	**0.11**(0.06)	**0.09**(0.03)	**0.16**(0.04)	**0.26**(0.10)
	100	Y	BNN	3.65(1.04)	2.70(0.92)	2.51(0.81)	2.48(0.82)	3.62(1.41)
			cWGAN	4.71(1.19)	2.59(0.84)	2.61(0.79)	2.74(0.77)	2.81(0.82)
			WGR	**3.24**(0.42)	**1.81**(0.47)	**1.94**(0.49)	**1.94**(0.44)	**2.07**(0.50)
M3	1	Y_1	BNN	11.62(0.85)	1.39(0.11)	0.21(0.06)	1.01(0.02)	8.50(0.45)
			cWGAN	0.41(0.24)	0.24(0.16)	0.16(0.09)	0.21(0.11)	**0.09**(0.06)
			WGR	**0.16**(0.12)	**0.09**(0.07)	**0.08**(0.07)	**0.09**(0.05)	**0.09**(0.16)
		Y_2	BNN	9.71(0.29)	1.10(0.13)	0.13(0.06)	1.67(0.03)	11.83(0.15)
			cWGAN	0.20(0.13)	0.30(0.24)	0.22(0.14)	0.34(0.32)	0.59(0.27)
			WGR	**0.14**(0.10)	**0.09**(0.06)	**0.05**(0.04)	**0.20**(0.05)	**0.50**(0.12)
M4	1	Y_1	BNN	0.93(0.38)	1.62(0.24)	0.05(0.01)	1.67(0.20)	0.96(0.36)
			cWGAN	0.05(0.05)	0.05(0.06)	**0.02**(0.02)	0.04(0.03)	**0.05**(0.04)
			WGR	**0.04**(0.03)	**0.04**(0.02)	**0.02**(0.01)	**0.03**(0.04)	**0.05**(0.03)
		Y_2	BNN	0.83(0.22)	1.60(0.16)	0.04(0.01)	1.58(0.08)	0.82(0.15)
			cWGAN	0.05(0.04)	0.04(0.03)	0.03(0.02)	0.04(0.02)	**0.05**(0.04)
			WGR	**0.03**(0.03)	**0.04**(0.04)	**0.02**(0.01)	**0.03**(0.03)	**0.05**(0.05)

Note. The corresponding standard errors are given in parentheses. The smallest MSEs are in bold font. WGR = Wasserstein generative regression; BNN = Bayesian Neural Networks; cWGAN = conditional Wasserstein generative adversarial network; MSE = mean squared error.

For evaluation, aside from the L_1 and L_2 errors given in Section 5.1, we consider the average length of the estimated 95% prediction interval (LPI), the CP, the standard deviation of upper bound error (SD-UBE) and lower bound error (SD-LBE), defined as

$$\text{LPI} = \frac{1}{K}\sum_{k=1}^{K}\left\{\hat{F}^{-1}_{Y|X}(0.975 \mid X = X'_k) - \hat{F}^{-1}_{Y|X}(0.025 \mid X = X'_k)\right\},$$

$$\text{CP} = \frac{1}{K}\sum_{k=1}^{K} I\left\{Y'_k \in [\hat{F}^{-1}_{Y|X}(0.025 \mid X = X'_k), \hat{F}^{-1}_{Y|X}(0.975 \mid X = X'_k)]\right\},$$

$$\text{SD-UBE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left|\hat{F}^{-1}_{Y|X}(0.975 \mid X = X'_k) - Y'_k\right|^2},$$

$$\text{SD-LBE} = \sqrt{\frac{1}{K}\sum_{k=1}^{K}\left|\hat{F}^{-1}_{Y|X}(0.025 \mid X = X'_k) - Y'_k\right|^2},$$

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

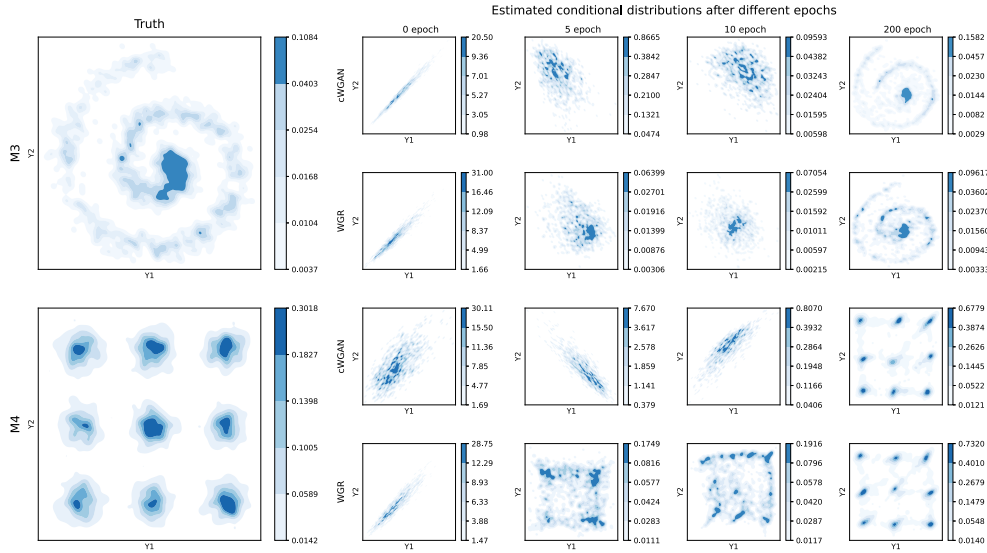

Figure 1. Comparison of conditional density estimation. The conditional distributions are estimated using 5,000 samples, generated by the conditional samplers for a randomly selected value of X. Bayesian Neural Networks (BNN) results are not shown, because its network parameter is collected at the end of the training process, making distribution estimates inaccessible during training.

Table 3. Summary statistics for CT test data

Method	L_1	L_2	LPI	CP	SD-UBE	SD-LBE
DNLS	**0.40**	0.51	**2.31**	**0.95**	0.33	**0.31**
BNN	0.80	1.27	3.14	0.88	0.86	0.90
cWGAN	0.95	2.30	1.54	0.48	0.46	0.50
WGR	0.45	**0.42**	2.44	0.96	**0.32**	**0.31**

Note. For DNLS, we use CQR to compute LPI, CP, SD-UBE, and SD-LBE. The best performances under each criterion are in bold font. LPI = length of the estimated 95% prediction interval; CP = coverage probability; SD-UBE = standard deviation of upper bound error; SD-LBE = standard deviation of lower bound error; DNLS = deep nonparametric least squares regression; BNN = Bayesian Neural Networks; cWGAN = conditional Wasserstein generative adversarial network; WGR = Wasserstein generative regression; CQR = conformal quantile regression.

where $I(\cdot)$ is the indicator function. Here, SD-UBE and SD-LBE are used to examine the stability of each method. The smaller SD-UBE and SD-LBE are, the greater the stability is. Since the τth conditional quantile cannot be computed for DNLS, we apply the conformal quantile regression (CQR, Romano et al., 2019) method to compute LPI, CP, SD-UBE, and SD-LBE for comparison.

5.2.1 The CT slices dataset

This dataset contains 53,500 CT images from 74 patients (43 male, 31 female) with anatomical landmarks annotated on the axial axis of the human body. Each CT image is represented by two histograms in polar space: 239 variables for the bone histogram and 145 for the air histogram, totalling 383 covariates. The response variable is the image's relative location on the axial axis, ranging from 0 (top of the head) to 180 (soles of the feet). The sample size of this dataset is 53,500. We use 40,000 samples for training, 3,500 for validation, and 10,000 for testing.

We summarize the numerical results in Table 3. Though WGR and DNLS perform comparably, the proposed WGR gives PI and CP directly, whereas DNLS computes the prediction intervals with the help of CQR. Furthermore, compared to cWGAN and BNN, WGR performs significantly better across all criteria.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Table 4. Analysis results of the UJIndoorLoc testing dataset

		L_1	L_2	LPI	CP	SD-UBE	SD-LBE
Longitude	DNLS	**0.07**	0.06	0.25	**0.88**	**0.11**	**0.10**
	BNN	0.30	0.17	3.13	1.00	0.48	0.47
	cWGAN	0.14	0.04	**0.24**	0.55	0.18	0.17
	WGR	0.08	**0.02**	0.29	0.82	0.12	0.12
Latitude	DNLS	**0.09**	0.10	0.34	**0.86**	1.81	1.82
	BNN	0.33	0.21	3.12	1.00	0.49	0.51
	cWGAN	0.17	0.07	**0.26**	0.50	0.20	0.20
	WGR	**0.09**	**0.02**	0.28	0.79	**0.13**	**0.14**
Floor	DNLS	**0.12**	0.05	**0.30**	**0.87**	1.38	1.40
	BNN	0.45	0.35	3.12	0.98	0.67	0.68
	cWGAN	0.23	0.11	0.31	0.44	0.34	0.35
	WGR	0.13	**0.04**	0.43	0.80	**0.20**	**0.20**
Building ID	DNLS	**0.05**	0.12	**0.16**	**0.94**	0.33	0.31
	BNN	0.30	0.17	3.11	1.00	0.44	0.48
	cWGAN	0.12	0.04	0.22	0.53	0.16	0.16
	WGR	0.06	**0.03**	0.28	0.90	**0.14**	**0.14**
Space ID	DNLS	**0.16**	0.21	0.44	**0.85**	1.46	1.43
	BNN	0.47	0.48	3.13	0.96	0.77	0.76
	cWGAN	0.29	0.36	0.61	0.68	0.47	0.58
	WGR	**0.16**	**0.12**	**0.35**	0.70	**0.33**	**0.31**

Note. The best performances under each criterion are in bold font. For DNLS, we use CQR to compute PI, CP, SD-UBE, and SD-LBE. LPI = length of the estimated 95% prediction interval; CP = coverage probability; SD-UBE = standard deviation of upper bound error; SD-LBE = standard deviation of lower bound error; DNLS = deep nonparametric least squares regression; BNN = Bayesian Neural Networks; cWGAN = conditional Wasserstein generative adversarial network; WGR = Wasserstein generative regression.

5.2.2 The UJIndoorLoc dataset

We analyse the UJIndoor dataset (Torres-Sospedra et al., 2014), a multi-building multi-floor indoor localization database that relies on WLAN/WiFi fingerprinting. This dataset contains 21,048 observations, divided into three parts: 14,948 for training, 1,100 for validation, and 5,000 for testing. Each observation has 529 attributes: 520 WiFi intensity values (from −104 to 0 dBm) and 100 for nondetected WAPs, serving as covariates, along with five location responses: *longitude, latitude, floor, building ID, and space ID*. The first two variables are continuous, while the others are categorical with at least two levels. We apply standardization before training. Our goal is to predict the location information from the WiFi intensity values using different methods and compare their performance.

Table 4 presents the analysis results. For all five response variables, WGR performs comparably to DNLS and significantly better than BNN and cWGAN in terms of L_1 and L_2. Compared to cWGAN, WGR provides prediction intervals with higher CP while maintaining a comparable length. Although DNLS gives prediction interval with higher CP and shorter length, its predictions are less stable, as the standard deviations of the upper and lower bound errors are larger compared to WGR. Additionally, BNN performs poorly on this dataset, producing very long prediction intervals with coverage probabilities close to 1.

5.2.3 MNIST handwritten digits dataset

We now demonstrate the performance of WGR on a problem, where both X and Y are high-dimensional. We use the MNIST dataset (LeCun et al., 2010), which consists of 28×28 grey-scale images with labels in $\{0, 1, \ldots, 9\}$. We apply WGR to reconstruct the central masked part of each

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

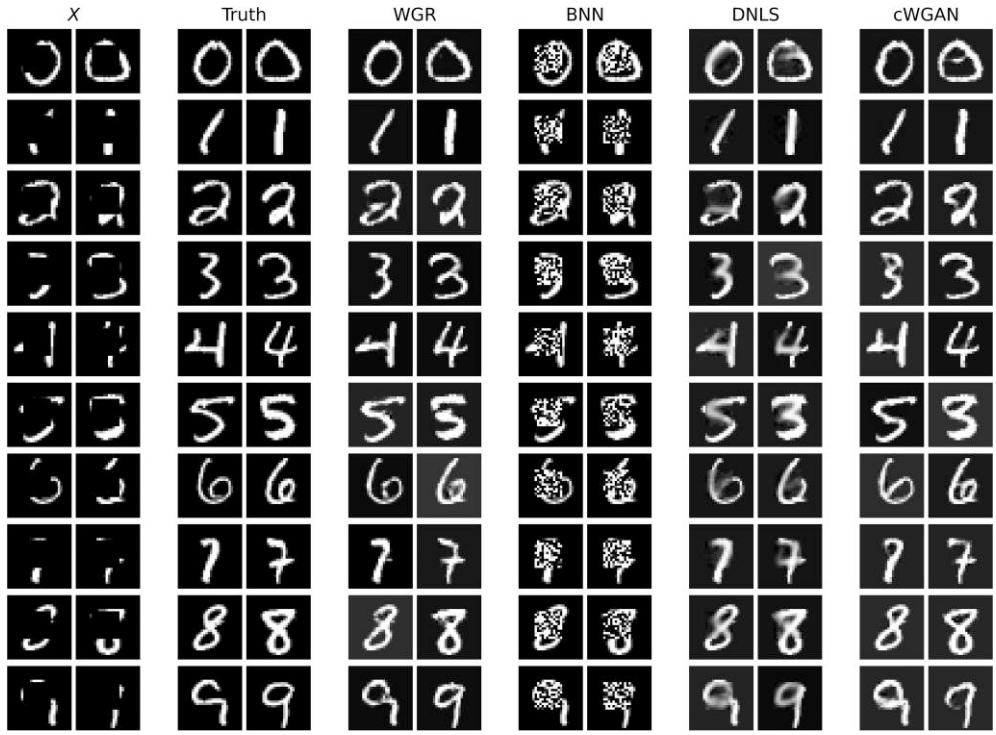

Figure 2. Reconstructed images in MNIST test data.

image, which is treated as the response $Y \in \mathbb{R}^{14\times 14}$, with the remaining part as the covariate X of dimension $28 \times 28 - 14 \times 14 = 588$. We randomly selected 20,000 images for training, 1,000 for validation, and 10,000 for testing. To evaluate reconstruction quality, we randomly sampled two images per digit from the test set and compared four different methods in Figure 2. The figure shows that WGR produces sharper and more faithful images than the other methods, as it preserves more details and reduces artifacts.

6 Conclusions

In this paper, we have proposed a generative regression approach, WGR, for simultaneously estimating a regression function and a conditional generator with theoretical guarantee. Our numerical experiments demonstrate that it works well in various situations from the standard generalized nonparametric regression problems to more complex image reconstruction tasks.

WGR can be viewed as a way of estimating a conditional generator with a data-dependent regularization on the first conditional moment, thereby providing a flexible and model-free way for constructing prediction intervals. Our framework can be adapted to other tasks by choosing different loss functions. For instance, we can estimate the conditional median function or the conditional quantile function by using other losses that are more suitable for these objectives. We can also impose regularization on higher conditional moments or other properties of the conditional distribution, depending on the research question.

Although we have established nonasymptotic error bounds and convergence properties of WGR, our analysis is only a first attempt to deal with a challenging technical problem that involves empirical processes on complex functional spaces and approximation properties of deep neural networks. Further work is needed to better understand the properties of generative regression methods, including the proposed WGR. For instance, it would be interesting to know if the error bounds we derived are optimal or if they can be improved. WGR is a nonparametric method. For statistical inference and model interpretation, it is desirable to incorporate a semiparametric

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

structure (Bickel et al., 1998) or a variable selection and dimension reduction component in WGR (Chen et al., 2022; Huang et al., 2012).

Generative regression leverages the power of deep neural networks to model complex and high-dimensional conditional distributions. Unlike traditional regression methods that only output point estimates, generative regression can capture the uncertainty and variability of the data by generating samples from the learned distribution. This allows for more interpretable results in various applications. For example, in financial applications, our method enables portfolio optimization through balancing expected returns against extreme risks such as conditional value at risk. It can also be applies to pathology for enhancing both detailed visualization (such as stain transfer) and high-level diagnostic assessment. We expect generative learning to be a useful addition to the existing methods for prediction and inference in statistics.

Acknowledgments

The authors thank the Editor, the Associate Editor, and three anonymous reviewers for their insightful comments and constructive suggestions that helped improve the paper significantly. This work was conducted when S. Song was a postdoctoral fellow in the Department of Statistics, The Chinese University of Hong Kong.

Conflicts of interest: All authors declare that they have no conflicts of interest.

Funding

S.S. research was partially supported by the National Natural Science Foundation of China grant (No. 12401362) and the Shanghai Rising-star Program grant (No. 24YF2748600). G.S. research was partially supported by the Hong Kong Research Grants Council (No. 15305523) and the research grant from The Hong Kong Polytechnic University (No. P0048718). Y.L. research was partially supported by the Hong Kong Research Grants Council (No. 14306620 and 14304523), and Direct Grants for Research, The Chinese University of Hong Kong. J.H. research was supported by the National Natural Science Foundation of China grant (No. 72331005) and the research grants from The Hong Kong Polytechnic University (No. P0046811, P0042888, P0045417 and P0045931).

Data availability

The code and the data supporting the findings of this work are openly available at https://github.com/Tong273/WGR.

Supplementary material

Supplementary material is available online at *Journal of the Royal Statistical Society: Series B*.

References

Antoniadis A., Grégoire G., & McKeague I. W. (2004). Bayesian estimation in single-index models. *Statistica Sinica*, *14*(4), 1147–1164.

Arjovsky M., Chintala S., & Bottou L. (2017). Wasserstein generative adversarial networks. In D. Precup, & Y. W. Teh (Eds.), *Proceedings of the 34th international conference on machine learning* (pp. 214–223). Proceedings of Machine Learning Research (PMLR).

Austin T. (2015). Exchangeable random measures. *Annales de l'IHP Probabilités et Statistiques*, *51*(3), 842–861. https://doi.org/10.1214/13-AIHP584

Bauer B., & Kohler M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics*, *47*(4), 2261–2285. https://doi.org/10.1214/18-AOS1747

Bickel P. J., Klaassen C. A. J., Yaácov R., & Wellner J. A. (1998). *Efficient and adaptive estimation for semiparametric models*. Springer.

Cai Z. (2002). Regression quantiles for time series. *Econometric Theory*, *18*(1), 169–192. https://doi.org/10.1017/S0266466602181096

Candès E., Lei L., & Ren Z. (2023). Conformalized survival analysis. *Journal of the Royal Statistical Society: Series B, Statistical Methodology*, *85*(1), 24–45. https://doi.org/10.1093/jrsssb/qkac004

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Chen Y., Gao Q., & Wang X. (2022). Inferential Wasserstein generative adversarial networks. *Journal of the Royal Statistical Society Series B*, *84*(1), 83–113. https://doi.org/10.1111/rssb.12476
Coates A., Ng A., & Lee H. (2011). An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th international conference on artificial intelligence and statistics* (pp. 215–223). Proceedings of Machine Learning Research (PMLR). https://proceedings.mlr.press/v15/coates11a.html
Fahrmeir L., & Lang S. (2001). Bayesian inference for generalized additive mixed models based on Markov random field priors. *Journal of the Royal Statistical Society: Series C, Applied Statistics*, *50*(2), 201–220. https://doi.org/10.1111/1467-9876.00229
Fan J., & Gijbels I. (1996). *Local polynomial modelling and its applications*. Monographs on statistics and applied probability series. Chapman & Hall.
Friedman J. H., & Stuetzle W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, *76*(376), 817–823. https://doi.org/10.1080/01621459.1981.10477729
Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., & Bengio Y. (2014). Generative adversarial nets. In *Proceedings of the 27th international conference on neural information processing systems* (pp. 2672–2680). Proceedings of Machine Learning Research (PMLR).
Graf F., Kriegel H.-P., Ólsterl S., & Schubert M. (2011). Position prediction in CT volume scans. In *Proceedings of the 28th international conference on machine learning. Workshop on learning for global challenges*. Omnipress.
Gulrajani I., Ahmed F., Arjovsky M., Dumoulin V., & Courville A. C. (2017). Improved training of Wasserstein GANs. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 5769–5779). Proceedings of Machine Learning Research (PMLR).
Györfi L., Kohler M., Krzyżak A., & Walk H. (2002). *A distribution-free theory of nonparametric regression*. Springer-Verlaghttps://doi.org/10.1007/b97848
Hall P., & Müller H.-G. (2003). Order-preserving nonparametric regression, with applications to conditional distribution and quantile function estimation. *Journal of the American Statistical Association*, *98*(463), 598–608. https://doi.org/10.1198/016214503000000512
Hall P., Wolff R. C., & Yao Q. (1999). Methods for estimating a conditional distribution function. *Journal of the American Statistical Association*, *94*(445), 154–163. https://doi.org/10.1080/01621459.1999.10473832
Hardle W., Hall P., & Ichimura H. (1993). Optimal smoothing in single-index models. *Annals of Statistics*, *21*(1), 157–178. https://doi.org/10.1214/aos/1176349020
Hastie T., & Tibshirani R. (1986). Generalized additive models. *Statistical Science*, *1*(3), 297–310. https://doi.org/10.1214/ss/1177013604
Hilton J., Dodd E., Forster J. J., & Smith P. W. (2019). Projecting UK mortality by using Bayesian generalized additive models. *Journal of the Royal Statistical Society: Series C, Applied Statistics*, *68*(1), 29–49. https://doi.org/10.1111/rssc.12299
Huang J., Breheny P., & Ma S. (2012). A selective review of group selection in high-dimensional models. *Statistical Science*, *27*(4), 481–499. https://doi.org/10.1214/12-STS392
Huang J., Jiao Y., Li Z., Liu S., Wang Y., & Yang Y. (2022). An error analysis of generative adversarial networks for learning distributions. *Journal of Machine Learning Research*, *23*(116), 1–43.
Ichimura H. (1993). Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. *Journal of Econometrics*, *58*(1–2), 71–120. https://www.sciencedirect.com/science/article/pii/030440769390114K. https://doi.org/10.1016/0304-4076(93)90114-K
Jiao Y., Shen G., Lin Y., & Huang J. (2023). Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds and polynomial prefactors. *Annals of Statistics*, *51*(2), 691–716. https://doi.org/10.1214/23-AOS2266
Jospin L. V., Laga H., Boussaid F., Buntine W., & Bennamoun M. (2022). Hands-on Bayesian neural networks—a tutorial for deep learning users. *IEEE Computational Intelligence Magazine*, *17*(2), 29–48. https://doi.org/10.1109/MCI.2022.3155327
Kallenberg O. (2002). *Foundations of modern probability*. Springer.
Klein N., Kneib T., & Lang S. (2015). Bayesian generalized additive models for location, scale, and shape for zero-inflated and overdispersed count data. *Journal of the American Statistical Association*, *110*(509), 405–419. https://doi.org/10.1080/01621459.2014.912955
Krizhevsky A. (2009). *Learning multiple layers of features from tiny images*. Toronto, ON, Canada. https://api.semanticscholar.org/CorpusID:18268744
LeCun Y., Cortes C., & Burges C. (2010). MNIST handwritten digit database. *AT&T Labs [Online]*. http://yann.lecun.com/exdb/mnist 2.
Lei J., Robins J., & Wasserman L. (2013). Distribution-free prediction sets. *Journal of the American Statistical Association*, *108*(501), 278–287. https://doi.org/10.1080/01621459.2012.751873
Lugosi G., & Zeger K. (1995). Nonparametric estimation via empirical risk minimization. *IEEE Transactions on Information Theory*, *41*(3), 677–687. https://doi.org/10.1109/18.382014

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026

Mielniczuk J., & Tyrcha J. (1993). Consistency of multilayer perceptron regression estimators. *Neural Networks*, *6*(7), 1019–1022. https://doi.org/10.1016/S0893-6080(09)80011-7
Müller A. (1997). Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, *29*(2), 429–443. https://doi.org/10.2307/1428011
Racine J. S., & Li K. (2017). Nonparametric conditional quantile estimation: A locally weighted quantile kernel approach. *Journal of Econometrics*, *201*(1), 72–94. https://doi.org/10.1016/j.jeconom.2017.06.020
Reed S., Akata Z., Yan X., Logeswaran L., Schiele B., & Lee H. (2016). Generative adversarial text to image synthesis. In *Proceedings of The 33rd international conference on machine learning* (pp. 1060–1069). Proceedings of Machine Learning Research (PMLR).
Reich B. J., Bondell H. D., & Li L. (2011). Sufficient dimension reduction via Bayesian mixture modeling. *Biometrics*, *67*(3), 886–895. https://doi.org/10.1111/biom.2011.67.issue-3
Romano Y., Patterson E., & Candes E. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems*, *32*, 3543–3553.
Salakhutdinov R. (2015). Learning deep generative models. *Annual Review of Statistics and Its Application*, *2*(1), 361–385. https://doi.org/10.1146/statistics.2015.2.issue-1
Salimans T., Goodfellow I., Zaremba W., Cheung V., Radford A., & Chen X. (2016). Improved techniques for training GANs. In *Advances in neural information processing systems* (Vol. *29*). Curran Associates, Inc..
Schmidt-Hieber J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, *48*(4), 1875–1897. 10.1214/19-AOS1931
Schwarz G. (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464. https://doi.org/10.1214/aos/1176344136
Scott D. W. (1992). *Multivariate density estimation: Theory, practice and visualization*. Wiley.
Sharma M., Farquhar S., Nalisnick E., & Rainforth T. (2023). Do Bayesian neural networks need to be fully stochastic? In *International conference on artificial intelligence and statistics* (pp. 7694–7722). Proceedings of Machine Learning Research (PMLR).
Silverman B. W. (1986). *Density estimation for statistics and data analysis*. Chapman & Hall.
Stone C. J. (1986). The dimensionality reduction principle for generalized additive models. *Annals of Statistics*, *14*(2), 590–606. https://doi.org/10.1214/aos/1176349940.
Torres-Sospedra J., Montoliu R., Martínez-Usó A., Avariento J. P., Arnau T. J., Benedito-Bordonau M., & Huerta J. (2014). UJIindoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems. In *International conference on indoor positioning and indoor navigation* (pp. 261–270). IEEE.
Tsybakov A. (2008). *Introduction to nonparametric estimation*. Springer Science & Business Media.
Veraverbeke N., Gijbels I., & Omelka M. (2014). Preadjusted non-parametric estimation of a conditional distribution function. *Journal of the Royal Statistical Society: Series B, Statistical Methodology*, *76*(2), 399–438. https://doi.org/10.1111/rssb.12041
Villani C. (2009). *Optimal transport: Old and new*. Springer.
Vovk V., Gammerman A., & Saunders C. (1999). Machine-learning applications of algorithmic randomness. In *International conference on machine learning* (pp. 444–453). Morgan Kaufmann Publishers Inc.
Wasserman L. (2006). *All of nonparametric statistics*. Springer texts in statistics. Springer-Verlag.
West M., Harrison P. J., & Migon H. S. (1985). Dynamic generalized linear models and Bayesian forecasting. *Journal of the American Statistical Association*, *80*(389), 73–83. https://doi.org/10.1080/01621459.1985.10477131
Yu K., & Jones M. C. (1998). Local linear quantile regression. *Journal of the American Statistical Association*, *93*(441), 228–237. https://doi.org/10.1080/01621459.1998.10474104
Zhou X., Jiao Y., Liu J., & Huang J. (2022). A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, *118*, 1837–1848. https://doi.org/10.1080/01621459.2021.2016424
Zhu J.-Y., Park T., Isola P., & Efros A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision* (pp. 2223–2232). IEEE Computer Society.

Downloaded from https://academic.oup.com/jrsssb/article/88/1/330/8240631 by The Hong Kong Polytechnic University user on 10 March 2026