CARD: Classification and Regression Diffusion Models

Xizewen Han[*] **Huangjie Zheng**[*]
Department of Statistics and Data Sciences
The University of Texas at Austin
Austin, TX 78712
{xizewen.han, huangjie.zheng}@utexas.edu

Mingyuan Zhou
McCombs School of Business
The University of Texas at Austin
Austin, TX 78712
mingyuan.zhou@mccombs.utexas.edu

Abstract

Learning the distribution of a continuous or categorical response variable $\boldsymbol{y}$ given its covariates $\boldsymbol{x}$ is a fundamental problem in statistics and machine learning. Deep neural network-based supervised learning algorithms have made great progress in predicting the mean of $\boldsymbol{y}$ given $\boldsymbol{x}$, but they are often criticized for their ability to accurately capture the uncertainty of their predictions. In this paper, we introduce classification and regression diffusion (CARD) models, which combine a denoising diffusion-based conditional generative model and a pre-trained conditional mean estimator, to accurately predict the distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$. We demonstrate the outstanding ability of CARD in conditional distribution prediction with both toy examples and real-world datasets, the experimental results on which show that CARD in general outperforms state-of-the-art methods, including Bayesian neural network-based ones that are designed for uncertainty estimation, especially when the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ is multi-modal. In addition, we utilize the stochastic nature of the generative model outputs to obtain a finer granularity in model confidence assessment at the instance level for classification tasks. Our implementation is publicly available at https://github.com/XzwHan/CARD.

1 Introduction

A fundamental problem in statistics and machine learning is to predict the response variable $\boldsymbol{y}$ given a set of covariates $\boldsymbol{x}$. Generally speaking, $\boldsymbol{y}$ is a continuous variable for regression analysis and a categorical variable for classification. Denote $f(\boldsymbol{x}) \in \mathbb{R}^C$ as a deterministic function that transforms $\boldsymbol{x}$ into a C dimensional output. Denote $f_c(\boldsymbol{x})$ as the c-th dimension of $f(\boldsymbol{x})$. Existing methods typically assume an additive noise model: for regression analysis with $\boldsymbol{y} \in \mathbb{R}^C$, one often assumes $\boldsymbol{y} = f(\boldsymbol{x}) + \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, while for classification with $y \in \{1, \ldots, C\}$, one often assumes $y = \arg\max \big(f_1(\boldsymbol{x}) + \boldsymbol{\epsilon}_1, \ldots, f_C(\boldsymbol{x}) + \boldsymbol{\epsilon}_C\big)$, where $\boldsymbol{\epsilon}_c \overset{iid}{\sim} \mathrm{EV}_1(0, 1)$, a standard type-1 extreme value distribution. Thus we have the expected value of $\boldsymbol{y}$ given $\boldsymbol{x}$ as $\mathbb{E}[\boldsymbol{y} \,|\, \boldsymbol{x}] = f(\boldsymbol{x})$ in regression and $P(y = c \,|\, \boldsymbol{x}) = \mathbb{E}[y = c \,|\, \boldsymbol{x}] = \mathrm{softmax}_c\big(f(\boldsymbol{x})\big) = \frac{\exp(f_c(\boldsymbol{x}))}{\sum_{c'=1}^{C} \exp(f_{c'}(\boldsymbol{x}))}$ in classification.

These additive-noise models are primarily focusing on accurately estimating the conditional mean $\mathbb{E}[\boldsymbol{y} \,|\, \boldsymbol{x}]$, while paying less attention to whether the noise distribution can accurately capture the uncertainty of $\boldsymbol{y}$ given $\boldsymbol{x}$. For this reason, they may not work well if the distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ clearly deviates from the additive-noise assumption. For example, if $p(\boldsymbol{y} \,|\, \boldsymbol{x})$ is multi-modal, which commonly happens when there are missing categorical covariates in $\boldsymbol{x}$, then $\mathbb{E}[\boldsymbol{y} \,|\, \boldsymbol{x}]$ may not be close to any possible true values of $\boldsymbol{y}$ given that specific $\boldsymbol{x}$. More specifically, consider a person whose weight, height, blood pressure, and age are known but gender is unknown, then the testosterone or estrogen level of this person is likely to follow a bi-modal distribution and the chance

[*] Equal contribution.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-1316

of developing breast cancer is also likely to follow a bi-modal distribution. Therefore, these widely used additive-noise models, which use a deterministic function $f(\boldsymbol{x})$ to characterize the conditional mean of $\boldsymbol{y}$, are inherently restrictive in their ability for uncertainty estimation.

In this paper, our goal is to accurately recover the full distribution of $\boldsymbol{y}$ conditioning on $\boldsymbol{x}$ given a set of N training data points, denoted as $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{1,N}$. To realize this goal, we consider the diffusion-based (a.k.a. score-based) generative models (Sohl-Dickstein et al., 2015; Song and Ermon, 2019; Ho et al., 2020; Song and Ermon, 2020; Song et al., 2021c) and inject covariate-dependence into both the forward and reverse diffusion chains. Our method can model the conditional distribution of both continuous and categorical $\boldsymbol{y}$ variables, and the algorithms developed under this method will be collectively referred to as **C**lassification **A**nd **R**egression **D**iffusion (CARD) models.

Diffusion-based generative models have received significant recent attention due to not only their ability to generate high-dimensional data, such as high-resolution photo-realistic images, but also their training stability. They can be understood from the perspective of score matching (Hyvärinen and Dayan, 2005; Vincent, 2011; Kadkhodaie and Simoncelli, 2021) and Langevin dynamics (Neal, 2011; Welling and Teh, 2011), as pioneered by Song and Ermon (2019). They can also be understood from the perspective of diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020), which first define a forward diffusion to transform the data into noise and then a reverse diffusion to regenerate the data from noise.

These previous methods mainly focus on unconditional generative modeling. While there exist guided-diffusion models (Song and Ermon, 2019; Song et al., 2021c; Dhariwal and Nichol, 2021; Nichol et al., 2022; Ramesh et al., 2022) that target on generating high-resolution photo-realistic images that match the semantic meanings or content of the label, text, or corrupted-images, we focus on studying diffusion-based conditional generative modeling at a more fundamental level. In particular, our goal is to thoroughly investigate whether CARD can help accurately recover $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathcal{D})$, the predictive distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ after observing data $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1,N}$. In other words, our focus is on regression analysis of continuous or categorical response variables given their corresponding covariates.

We summarize our main contributions as follows: 1) We show CARD, which injects covariate-dependence and a pre-trained conditional mean estimator into both the forward and reverse diffusion chains to construct a denoising diffusion probabilistic model, provides an accurate estimation of $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathcal{D})$. 2) We provide a new metric to better evaluate how well a regression model captures the full distribution $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \mathcal{D})$. 3) Experiments on standard benchmarks for regression analysis show that CARD achieves state-of-the-art results, using both existing metrics and the new one. 4) For classification tasks, we push the assessment of model confidence in its predictions towards the level of individual instances, a finer granularity than the previous methods.

2 Methods and Algorithms for CARD

Given the ground-truth response variable $\boldsymbol{y}_0$ and its covariates $\boldsymbol{x}$, and assuming a sequence of intermediate predictions $\boldsymbol{y}_{1:T}$ made by the diffusion model, the goal of supervised learning is to learn a model such that the log-likelihood is maximized by optimizing the following ELBO:

$$\log p_\theta(\boldsymbol{y}_0 \,|\, \boldsymbol{x}) = \log \int p_\theta(\boldsymbol{y}_{0:T} \,|\, \boldsymbol{x}) d\boldsymbol{y}_{1;T} \geq \mathbb{E}_{q(\boldsymbol{y}_{1:T} \,|\, \boldsymbol{y}_0, \boldsymbol{x})} \left[\log \frac{p_\theta(\boldsymbol{y}_{0:T} \,|\, \boldsymbol{x})}{q(\boldsymbol{y}_{1:T} \,|\, \boldsymbol{y}_0, \boldsymbol{x})} \right], \tag{1}$$

where $q(\boldsymbol{y}_{1:T} \,|\, \boldsymbol{y}_0, \boldsymbol{x})$ is called the forward process or diffusion process in the concept of diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020). Denote $D_{\mathrm{KL}}(q \,||\, p)$ as the Kullback–Leibler (KL) divergence from distribution p to distribution q. The above objective can be rewritten as

$$\mathcal{L}_{\mathrm{ELBO}}(\boldsymbol{y}_0, \boldsymbol{x}) \coloneqq \mathcal{L}_0(\boldsymbol{y}_0, \boldsymbol{x}) + \sum_{t=2}^{T} \mathcal{L}_{t-1}(\boldsymbol{y}_0, \boldsymbol{x}) + \mathcal{L}_T(\boldsymbol{y}_0, \boldsymbol{x}), \tag{2}$$

$$\mathcal{L}_0(\boldsymbol{y}_0, \boldsymbol{x}) \coloneqq \mathbb{E}_q \left[-\log p_\theta(\boldsymbol{y}_0 \,|\, \boldsymbol{y}_1, \boldsymbol{x}) \right], \tag{3}$$

$$\mathcal{L}_{t-1}(\boldsymbol{y}_0, \boldsymbol{x}) \coloneqq \mathbb{E}_q \left[D_{\mathrm{KL}}\big(q(\boldsymbol{y}_{t-1} \,|\, \boldsymbol{y}_t, \boldsymbol{y}_0, \boldsymbol{x}) \,\big\|\, p_\theta(\boldsymbol{y}_{t-1} \,|\, \boldsymbol{y}_t, \boldsymbol{x})\big) \right], \tag{4}$$

$$\mathcal{L}_T(\boldsymbol{y}_0, \boldsymbol{x}) \coloneqq \mathbb{E}_q \left[D_{\mathrm{KL}}\big(q(\boldsymbol{y}_T \,|\, \boldsymbol{y}_0, \boldsymbol{x}) \,\big\|\, p(\boldsymbol{y}_T \,|\, \boldsymbol{x})\big) \right]. \tag{5}$$

Here we follow the convention to assume $\mathcal{L}_T$ does not depend on any parameter and it will be close to zero by carefully diffusing the observed response variable $\boldsymbol{y}_0$ towards a pre-assumed distribution

https://doi.org/10.52202/068431-1316

$p(\boldsymbol{y}_T \mid \boldsymbol{x})$. The remaining terms will make the model $p_\theta(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{x})$ approximate the corresponding tractable ground-truth denoising transition step $q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{y}_0, \boldsymbol{x})$ for all timesteps. Different from the vanilla diffusion models, we assume the endpoint of our diffusion process to be

$$p(\boldsymbol{y}_T \mid \boldsymbol{x}) = \mathcal{N}(f_\phi(\boldsymbol{x}), \boldsymbol{I}), \tag{6}$$

where $f_\phi(\boldsymbol{x})$ is the prior knowledge of the relation between $\boldsymbol{x}$ and $\boldsymbol{y}_0$, *e.g.*, a network pre-trained with $\mathcal{D}$ to approximate $\mathbb{E}[\boldsymbol{y} \mid \boldsymbol{x}]$, or $\mathbf{0}$ if we assume the relation is unknown. With a diffusion schedule $\{\beta_t\}_{t=1:T} \in (0,1)^T$, we specify the forward process conditional distributions in a similar fashion as Pandey et al. (2022), but for all timesteps including $t = 1$:

$$q\big(\boldsymbol{y}_t \mid \boldsymbol{y}_{t-1}, f_\phi(\boldsymbol{x})\big) = \mathcal{N}\big(\boldsymbol{y}_t; \sqrt{1-\beta_t}\boldsymbol{y}_{t-1} + (1-\sqrt{1-\beta_t}) f_\phi(\boldsymbol{x}), \beta_t \boldsymbol{I}\big), \tag{7}$$

which admits a closed-form sampling distribution with an arbitrary timestep t:

$$q\big(\boldsymbol{y}_t \mid \boldsymbol{y}_0, f_\phi(\boldsymbol{x})\big) = \mathcal{N}\big(\boldsymbol{y}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{y}_0 + (1-\sqrt{\bar{\alpha}_t}) f_\phi(\boldsymbol{x}), (1-\bar{\alpha}_t) \boldsymbol{I}\big), \tag{8}$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_t \alpha_t$. Note that the mean term in Eq. (7) can be viewed as an interpolation between true data $\boldsymbol{y}_0$ and the predicted conditional expectation $f_\phi(\boldsymbol{x})$, which gradually changes from the former to the latter throughout the forward process.

Such formulation corresponds to a tractable forward process posterior:

$$q(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{y}_0, \boldsymbol{x}) = q\big(\boldsymbol{y}_{t-1} \mid \boldsymbol{y}_t, \boldsymbol{y}_0, f_\phi(\boldsymbol{x})\big) = \mathcal{N}\Big(\boldsymbol{y}_{t-1}; \tilde{\boldsymbol{\mu}}\big(\boldsymbol{y}_t, \boldsymbol{y}_0, f_\phi(\boldsymbol{x})\big), \tilde{\beta}_t \boldsymbol{I}\Big), \tag{9}$$

where

$$\tilde{\boldsymbol{\mu}} := \underbrace{\frac{\beta_t\sqrt{\bar{\alpha}_{t-1}}}{1-\bar{\alpha}_t}}_{\gamma_0}\boldsymbol{y}_0 + \underbrace{\frac{(1-\bar{\alpha}_{t-1})\sqrt{\alpha_t}}{1-\bar{\alpha}_t}}_{\gamma_1}\boldsymbol{y}_t + \underbrace{\left(1 + \frac{(\sqrt{\bar{\alpha}_t}-1)(\sqrt{\alpha_t}+\sqrt{\bar{\alpha}_{t-1}})}{1-\bar{\alpha}_t}\right)}_{\gamma_2} f_\phi(\boldsymbol{x}),$$

$$\tilde{\beta}_t := \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t.$$

We provide the derivation in Appendix A.1. The labels under the terms are used in Algorithm 2.

2.1 CARD for Regression

For regression problems, the goal of the reverse diffusion process is to gradually recover the distribution of the noise term, the aleatoric or local uncertainty inherent in the observations (Kendall and Gal, 2017; Wang and Zhou, 2020), enabling us to generate samples that match the true conditional $p(\boldsymbol{y} \mid \boldsymbol{x})$.

Following the reparameterization introduced by denoising diffusion probabilistic models (DDPM) (Ho et al., 2020), we construct $\boldsymbol{\epsilon}_\theta\big(\boldsymbol{x}, \boldsymbol{y}_t, f_\phi(\boldsymbol{x}), t\big)$, which is a function approximator parameterized by a deep neural network that predicts the forward diffusion noise $\boldsymbol{\epsilon}$ sampled for $\boldsymbol{y}_t$. The training and inference procedure can be carried out in a standard DDPM manner.

Algorithm 1 Training (Regression)

```
Pre-train $f_\phi(\boldsymbol{x})$ that predicts $\mathbb{E}(\boldsymbol{y} \mid \boldsymbol{x})$ with MSE
repeat
    Draw $y_0 \sim q(\boldsymbol{y}_0 \mid \boldsymbol{x})$
    Draw $t \sim$ Uniform($\{1 \ldots T\}$)
    Draw $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
    Compute noise estimation loss
        $\mathcal{L}_{\boldsymbol{\epsilon}} = \big|\big|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\big(\boldsymbol{x}, \sqrt{\bar{\alpha}_t}\boldsymbol{y}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} + (1-\sqrt{\bar{\alpha}_t}) f_\phi(\boldsymbol{x}), f_\phi(\boldsymbol{x}), t\big)\big|\big|^2$
    Take numerical optimization step on:
        $\nabla_\theta \mathcal{L}_{\boldsymbol{\epsilon}}$
until Convergence
```

Algorithm 2 Inference (Regression)

```
$\boldsymbol{y}_T \sim \mathcal{N}(f_\phi(\boldsymbol{x}), \boldsymbol{I})$
for $t = T$ to $1$ do
    Draw $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$if $t > 1$
    Calculate reparameterized $\hat{\boldsymbol{y}}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}\Big(\boldsymbol{y}_t - (1 - \sqrt{\bar{\alpha}_t})f_\phi(\boldsymbol{x}) - \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}_\theta\big(\boldsymbol{x}, \boldsymbol{y}_t, f_\phi(\boldsymbol{x}), t\big)\Big)$
    Let $\boldsymbol{y}_{t-1} = \gamma_0 \hat{\boldsymbol{y}}_0 + \gamma_1 \boldsymbol{y}_t + \gamma_2 f_\phi(\boldsymbol{x}) + \sqrt{\tilde{\beta}_t}\boldsymbol{z}$ if $t > 1$, else set $\boldsymbol{y}_{t-1} = \hat{\boldsymbol{y}}_0$
end for
return $\boldsymbol{y}_0$
```

2.2 CARD for Classification

We formulate the classification tasks in a similar fashion as in Section 2.1, where we:

1. Replace the continuous response variable with a one-hot encoded label vector for $\boldsymbol{y}_0$;
2. Replace the mean estimator with a pre-trained classifier that outputs softmax probabilities of the class labels for $f_\phi(\boldsymbol{x})$.

This construction no longer assumes $\boldsymbol{y}_0$ to be drawn from a categorical distribution, but instead treats each one-hot label as a class prototype, *i.e.*, we assume a continuous data and state space, which enables us to keep the Gaussian diffusion model framework. The sampling procedure would output reconstructed $\boldsymbol{y}_0$ in the range of real numbers for each dimension, instead of a vector in the probability simplex. Denoting C as the number of classes and $\boldsymbol{1}_C$ as a C-dimensional vector of 1s, we convert such output to a probability vector in a softmax form of a temperature-weighted Brier score (Brier, 1950), which computes the squared error between the prediction and $\boldsymbol{1}_C$. Mathematically, the probability of predicting the k^{th} class and the final point prediction $\hat{y}$ can be expressed as

$$\text{Pr}(y = k) = \frac{\exp(-(\boldsymbol{y}_0 - \boldsymbol{1}_C)_k^2/\tau)}{\sum_{i=1}^{C}\exp(-(\boldsymbol{y}_0 - \boldsymbol{1}_C)_i^2/\tau)};\ \hat{y} = \arg\max_k \big(-(\boldsymbol{y}_0 - \boldsymbol{1}_C)_k^2\big). \tag{10}$$

where $\tau > 0$ is the temperature parameter, and $(\boldsymbol{y}_0 - \boldsymbol{1}_C)_k^2$ indicates the k^{th} dimension of the vector of element-wise square error between $\boldsymbol{y}_0$ and $\boldsymbol{1}_C$, *i.e.*, $(\boldsymbol{y}_0 - \boldsymbol{1}_C)_k^2 = \|\boldsymbol{y}_{0_k} - 1\|^2$. Intuitively, this construction would assign the class whose raw output in the sampled $\boldsymbol{y}_0$ is closest to the true class, encoded by the value of 1 in the one-hot label, with the highest probability.

Conditional on the same covariates $\boldsymbol{x}$, the stochasticity of the generative model would give us a different class prototype reconstruction after each reverse process sampling, which enables us to construct predicted probability intervals for all class labels. Such stochastic reconstruction is in a similar fashion as DALL-E 2 (Ramesh et al., 2022) that applies a diffusion prior to reconstruct the image embedding by conditioning on the text embedding during the reverse diffusion process, which is a key step in the diversity of generated images.

3 Related Work

Under the supervised learning settings, to model the conditional distribution $p(\boldsymbol{y} \,|\, \boldsymbol{x})$ besides just the conditional mean $\mathbb{E}[\boldsymbol{y} \,|\, \boldsymbol{x}]$ through deep neural networks, existing works have been focusing on quantifying predictive uncertainty, and several lines of work have been proposed. Bayesian neural networks (BNNs) model such uncertainty by assuming distributions over network parameters, capturing the plausibility of the model given the data (Blundell et al., 2015; Hernández-Lobato and Adams, 2015; Gal and Ghahramani, 2016; Kingma et al., 2015; Tomczak et al., 2021). Kendall and Gal (2017) also model the uncertainties in the model outputs besides model parameters, by including the additive noise term as part of the neural network output. Meanwhile, ensemble-based methods (Lakshminarayanan et al., 2017; Liu et al., 2022) have been proposed to model predictive uncertainty by combining multiple neural networks with stochastic outputs. Furthermore, the Neural Processes Family (Garnelo et al., 2018b,a; Kim et al., 2019; Gordon et al., 2020) has introduced a series of models that capture predictive uncertainty in an out-of-distribution fashion, particularly designed for few-shot learning settings.

These above mentioned models have all assumed a parametric form in $p(\boldsymbol{y} \,|\, \boldsymbol{x})$, namely Gaussian distribution, or a mixture of Gaussians, and optimize the network parameters based on a Gaussian negative log-likelihood objective function. Deep generative models, on the other hand, have been known for modeling implicit distributions without parametric distributional assumptions, but very few works have been proposed to utilize such feature to tackle regression tasks. GAN-based models are introduced by Zhou et al. (2021) and Liu et al. (2021) for conditional density estimation and predictive uncertainty quantification. For classification tasks, on the other hand, generative classifiers (Revow et al., 1996; Fetaya et al., 2020; Ardizzone et al., 2020; Mackowiak et al., 2021) is a class of models that also perform classification with generative models; among them, Zimmermann et al. (2021) propose score-based generative classifiers to tackle classification tasks with score-based generative models (Song et al., 2021b,c). They model $p(\boldsymbol{x} \,|\, \boldsymbol{y})$ and predict the label with the largest conditional likelihood of $\boldsymbol{x}$, while CARD models $p(\boldsymbol{y} \,|\, \boldsymbol{x})$ instead.

In recent years, the class of diffusion-based (or score-based) deep generative models has demonstrated its outstanding performance in modeling high-dimensional multi-modal distributions (Ho et al., 2020; Song et al., 2021a; Kawar et al., 2022; Xiao et al., 2022; Dhariwal and Nichol, 2021; Song and Ermon, 2019, 2020), with most work focusing on Gaussian diffusion processes operating in continuous state spaces. Hoogeboom et al. (2021) introduce extensions of diffusion models for categorical data, and Austin et al. (2021) have proposed diffusion models for discrete data as a generalization of the multinomial diffusion models, which could provide an alternative way of performing classification with diffusion-based models.

4 Experiments

For the hyperparameters of CARD in both regression and classification tasks, we set the number of timesteps as $T = 1000$, a linear noise schedule with $\beta_1 = 10^{-4}$ and $\beta_T = 0.02$, same as Ho et al. (2020). We provide a more detailed walk-through of the experimental setup, including training and network architecture, in Appendix A.8.

4.1 Regression

Putting aside its statistical interpretation, the word *regress* indicates a direction opposite to *progress*, suggesting a less developed state. Such semantics in fact translates well into the statistical domain, in the sense that traditional regression analysis methods often only focus on estimating $\mathbb{E}(\boldsymbol{y} \,|\, \boldsymbol{x})$, while leaving out all remaining details about $p(\boldsymbol{y} \,|\, \boldsymbol{x})$. In recent years, Bayesian neural networks (BNNs) have emerged as a class of models that aims at estimating the uncertainty (Hernández-Lobato and Adams, 2015; Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Tomczak et al., 2021), providing a more complete picture of $p(\boldsymbol{y} \,|\, \boldsymbol{x})$. The metric that they use to quantify uncertainty estimation, negative log-likelihood (NLL), is computed with a Gaussian density, implying their assumption such that the conditional distributions $p(\boldsymbol{y} \,|\, \boldsymbol{x} = x)$ for all x are Gaussian. However, this assumption is very difficult to verify for real-world datasets: the covariates can be arbitrarily high-dimensional, making the feature space increasingly sparse with respect to the number of collected observations.

To accommodate the need for uncertainty estimation without imposing such restriction for the parametric form of $p(\boldsymbol{y} \,|\, \boldsymbol{x})$, we apply the following two metrics, both of which are designed to empirically evaluate the level of similarity between the learned and the true conditional distributions:

1. Prediction Interval Coverage Probability (PICP);
2. Quantile Interval Coverage Error (QICE).

PICP has been described in Yao et al. (2019), whereas QICE is a new metric proposed by us. We describe both of them in what follows.

4.1.1 PICP and QICE

The PICP is computed as

$$\text{PICP} := \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}_{y_n \geq \hat{y}_n^{\text{low}}} \cdot \mathbb{1}_{y_n \leq \hat{y}_n^{\text{high}}}, \tag{11}$$

https://doi.org/10.52202/068431-1316

where $\hat{y}_n^{\text{low}}$ and $\hat{y}_n^{\text{high}}$ represent the low and high percentiles, respectively, of our choice for the predicted $\boldsymbol{y}$ outputs given the same $\boldsymbol{x}$ input. This metric measures the proportion of true observations that fall in the percentile range of the generated $\boldsymbol{y}$ samples given each $\boldsymbol{x}$ input. Intuitively, when the learned distribution represents the true distribution well, this measurement should be close to the difference between the selected low and high percentiles. In this paper, we choose the 2.5^{th} and 97.5^{th} percentile, thus an ideal PICP value for the learned model should be 95%.

Meanwhile, there is a caveat for this metric: for example, imagine a situation where the 2.5^{th} to 97.5^{th} percentile of the learned distribution happens to cover the data between the 1^{st} and 96^{th} percentiles from the true distribution. Given enough samples, we shall still obtain a PICP value close to 95%, but clearly there is a mismatch between the learned distribution and the true one.

Based on such reasoning, we propose a new empirical metric QICE, which by design can be viewed as PICP with finer granularity, and without uncovered quantile ranges. To compute QICE, we first generate enough $\boldsymbol{y}$ samples given each $\boldsymbol{x}$, and divide them into M bins with roughly equal sizes. We would obtain the corresponding quantile values at each boundary. In this paper, we set $M = 10$, and obtain the following 10 quantile intervals (QIs) of the generated $\boldsymbol{y}$ samples: below the 10^{th} percentile, between the 10^{th} and 20^{th} percentiles, $\dots$, between the 80^{th} and 90^{th} percentiles, and above the 90^{th} percentile. Optimally, when the learned conditional distribution is identical to the true one, given enough samples from both learned and true distribution we shall observe about 10% of true data falling into each of these 10 QIs.

We define *QICE* to be the mean absolute error between the proportion of true data contained by each QI and the optimal proportion, which is $1/M$ for all intervals:

$$\text{QICE} := \frac{1}{M}\sum_{m=1}^{M}\left|r_m - \frac{1}{M}\right|, \text{ where } r_m = \frac{1}{N}\sum_{n=1}^{N}\mathbb{1}_{y_n \geq \hat{y}_n^{\text{low}_m}} \cdot \mathbb{1}_{y_n \leq \hat{y}_n^{\text{high}_m}}. \tag{12}$$

Intuitively, under optimal scenario with enough samples, we shall obtain a QICE value of 0. Note that each r_m is indeed the PICP for the corresponding QI with boundaries at $\hat{y}_n^{\text{low}_m}$ and $\hat{y}_n^{\text{high}_m}$. Since the true $\boldsymbol{y}$ for each $\boldsymbol{x}$ is guaranteed to fall into one of these QIs, we are thus able to overcome the mismatch issue described in the above example for PICP: fewer true instances falling into one QI would result in more instances captured by another QI, thus increasing the absolute error for both QIs.

QICE is similar to NLL in the sense that it also utilizes the summary statistics of the samples from the learned distribution conditional on each new $\boldsymbol{x}$ to empirically evaluate how well the model fits the true data. Meanwhile, it does not assume any parametric form on the conditional distribution, making it a much more generalizable metric to measure the level of distributional match between the learned and the underlying true conditional distributions, especially when the true conditional distribution is known to be multi-modal. We will demonstrate this point through the regression toy examples.

4.1.2 Toy Examples

To demonstrate the effectiveness of CARD in regression tasks for not only learning the conditional mean $\mathbb{E}(\boldsymbol{y}\,|\,\boldsymbol{x})$, but also recreating the ground truth data generating mechanism, we first apply CARD on 8 toy examples, whose data generating functions are designed to possess different statistical characteristics: some have a uni-modal symmetric distribution for their error term (linear regression, quadratic regression, sinusoidal regression), others have heteroscedasticity (log-log linear regression, log-log cubic regression) or multi-modality (inverse sinusoidal regression, 8 Gaussians, full circle). We show that the trained CARD models can generate samples that are visually indistinguishable from the true response variables of the new covariates, as well as quantitatively match the true distribution in terms of some summary statistics. We present the scatter plots of both true and generated data for all 8 tasks in Figure 1. For tasks with uni-modal conditional distribution, we fill the region between the 2.5^{th} and 97.5^{th} percentile of the generated $\boldsymbol{y}$'s. We observe that within each task, the generated samples blend remarkably well with the true test instances, suggesting the capability of reconstructing the underlying data generation mechanism by CARD. A more detailed description of the toy examples, including more quantitative analyses, is presented in Appendix A.13.

4.1.3 UCI Regression Tasks

We continue to investigate our model through experiments on real-world datasets. We adopt the same set of 10 UCI regression benchmark datasets (Dua and Graff, 2017) as well as the experimental

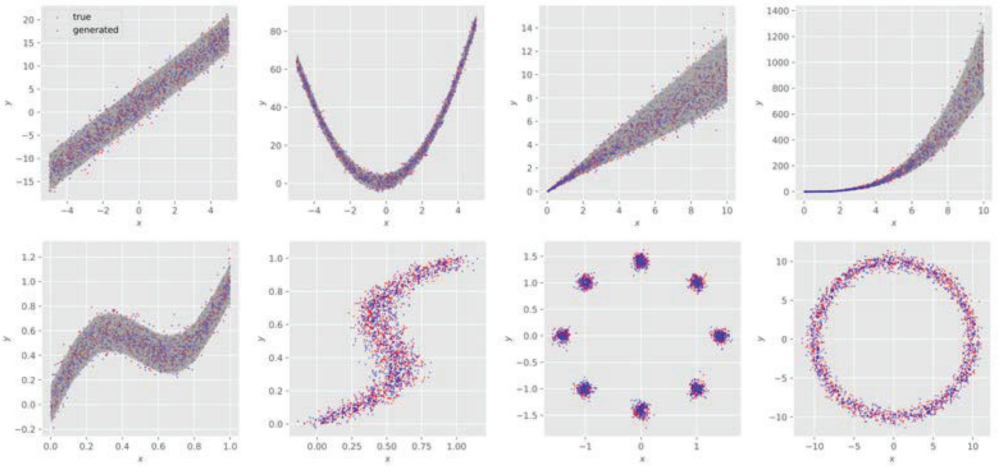

Figure 1: Regression toy example scatter plots. (**Top**) left to right: linear regression, quadratic regression, log-log linear regression, log-log cubic regression; (**Bottom**) left to right: sinusoidal regression, inverse sinusoidal regression, 8 Gaussians, full circle.

protocol proposed by Hernández-Lobato and Adams (2015) and followed by Gal and Ghahramani (2016) and Lakshminarayanan et al. (2017). The dataset information is provided in Table 14.

We apply multiple train-test splits with $90\%/10\%$ ratio in the same way as Hernández-Lobato and Adams (2015) (20 folds for all datasets except 5 for Protein and 1 for Year), and report the metrics by their mean and standard deviation across all splits. We compare our method to all aforementioned BNN frameworks: PBP, MC Dropout, and Deep Ensembles, as well as another deep generative model that estimates a conditional distribution sampler, GCDS (Zhou et al., 2021). We note that GCDS is related to a concurrent work of Yang et al. (2022), who share a comparable idea but use it in a different application. Following the same paradigm of BNN model assessment, we evaluate the accuracy and predictive uncertainty estimation of CARD by reporting RMSE and NLL. Furthermore, we also report QICE for all methods to evaluate distributional matching. Since this new metric was not applied in previous methods, we re-ran the experiments for all BNNs and obtained comparable or slightly better results in terms of other commonly used metrics reported in their literature. Further details about the experimental setup for these models can be found in Appendix A.9. The experiment results with corresponding metrics are shown in Tables 1, 2, and 3, with the number of times that each model achieves the best corresponding metric reported in the last row.

We observe that CARD outperforms existing methods, often by a considerable margin (especially on larger datasets), in all metrics for most of the datasets, and is competitive with the best method for the remaining ones: we obtain state-of-the-art results in 9 out of 10 datasets in terms of RMSE, 8 out of 10 for NLL, and 5 out of 10 for QICE. It is worth noting that although we do not explicitly optimize our model by MSE or by NLL, we still obtain better results than models trained with these objectives.

4.2 Classification

Similar to Lakshminarayanan et al. (2017), our motivation for classification is not to achieve state-of-the-art performance in terms of mean accuracy on the benchmark datasets, which is strongly related to network architecture design. Our goal is two-fold:

1. We aim to solve classification problems via a generative model, emphasizing its capability to improve the performance of a base classifier with deterministic outputs in terms of accuracy;
2. We intend to provide an alternative sense of uncertainty, by introducing the idea of model confidence at the instance level, *i.e.*, how sure the model is about *each* of its predictions, through the stochasticity of outputs from a generative model.

As another type of supervised learning problems, classification is different from regression mainly for the response variable being discrete class labels instead of continuous values. The conventional operation is to cast the classifier output as a point estimate, with a value between 0 and 1. Such

Table 1: RMSE of UCI regression tasks. For both ^{1}Kin8nm and ^{2}Naval dataset, we multiply the response variable by 100 to match the scale of others.

Dataset			RMSE ↓		
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	2.89 ± 0.74	3.06 ± 0.96	3.17 ± 1.05	2.75 ± 0.58	**2.61 ± 0.63**
Concrete	5.55 ± 0.46	5.09 ± 0.60	4.91 ± 0.47	5.39 ± 0.55	**4.77 ± 0.46**
Energy	1.58 ± 0.21	1.70 ± 0.22	2.02 ± 0.32	0.64 ± 0.09	**0.52 ± 0.07**
Kin8nm^1	9.42 ± 0.29	7.10 ± 0.26	8.65 ± 0.47	8.88 ± 0.42	**6.32 ± 0.18**
Naval^2	0.41 ± 0.08	0.08 ± 0.03	0.09 ± 0.01	0.14 ± 0.05	**0.02 ± 0.00**
Power	4.10 ± 0.15	4.04 ± 0.14	4.02 ± 0.15	4.11 ± 0.16	**3.93 ± 0.17**
Protein	4.65 ± 0.02	4.16 ± 0.12	4.45 ± 0.02	4.50 ± 0.02	**3.73 ± 0.01**
Wine	0.64 ± 0.04	**0.62 ± 0.04**	0.63 ± 0.04	0.66 ± 0.04	0.63 ± 0.04
Yacht	0.88 ± 0.22	0.84 ± 0.27	1.19 ± 0.49	0.79 ± 0.26	**0.65 ± 0.25**
Year	8.86± NA	8.77± NA	8.79± NA	9.20± NA	**8.70**± NA
# best	0	1	0	0	**9**

Table 2: NLL of UCI regression tasks.

Dataset			NLL ↓		
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	2.53 ± 0.27	2.46 ± 0.12	2.35 ± 0.16	18.66 ± 8.92	**2.35 ± 0.12**
Concrete	3.19 ± 0.05	3.21 ± 0.18	**2.93 ± 0.12**	13.64 ± 6.88	2.96 ± 0.09
Energy	2.05 ± 0.05	1.50 ± 0.11	1.40 ± 0.27	1.46 ± 0.72	**1.04 ± 0.06**
Kin8nm	−0.83 ± 0.02	−1.14 ± 0.05	−1.06 ± 0.02	−0.38 ± 0.36	**−1.32 ± 0.02**
Naval	−3.97 ± 0.10	−4.45 ± 0.38	−5.94 ± 0.10	−5.06 ± 0.48	**−7.54 ± 0.05**
Power	2.92 ± 0.02	2.90 ± 0.03	2.89 ± 0.02	2.83 ± 0.06	**2.82 ± 0.02**
Protein	3.05 ± 0.00	2.80 ± 0.08	2.89 ± 0.02	2.81 ± 0.09	**2.49 ± 0.03**
Wine	1.03 ± 0.03	0.93 ± 0.06	0.96 ± 0.06	6.52 ± 21.86	**0.92 ± 0.05**
Yacht	1.58 ± 0.08	1.73 ± 0.22	1.11 ± 0.18	**0.61 ± 0.34**	0.90 ± 0.08
Year	3.69± NA	3.42± NA	3.44± NA	3.43± NA	**3.34**± NA
# best	0	0	1	1	**8**

Table 3: QICE (in %) of UCI regression tasks.

Dataset			QICE ↓		
	PBP	MC Dropout	Deep Ensembles	GCDS	CARD (ours)
Boston	3.50 ± 0.88	3.82 ± 0.82	**3.37 ± 0.00**	11.73 ± 1.05	3.45 ± 0.83
Concrete	2.52 ± 0.60	4.17 ± 1.06	2.68 ± 0.64	10.49 ± 1.01	**2.30 ± 0.66**
Energy	6.54 ± 0.90	5.22 ± 1.02	**3.62 ± 0.58**	7.41 ± 2.19	4.91 ± 0.94
Kin8nm	1.31 ± 0.25	1.50 ± 0.32	1.17 ± 0.22	7.73 ± 0.80	**0.92 ± 0.25**
Naval	4.06 ± 1.25	12.50 ± 1.95	6.64 ± 0.60	5.76 ± 2.25	**0.80 ± 0.21**
Power	**0.82 ± 0.19**	1.32 ± 0.37	1.09 ± 0.26	1.77 ± 0.33	0.92 ± 0.21
Protein	1.69 ± 0.09	2.82 ± 0.41	2.17 ± 0.16	2.33 ± 0.18	**0.71 ± 0.11**
Wine	**2.22 ± 0.64**	2.79 ± 0.56	2.37 ± 0.63	3.13 ± 0.79	3.39 ± 0.69
Yacht	6.93 ± 1.74	10.33 ± 1.34	7.22 ± 1.41	**5.01 ± 1.02**	8.03 ± 1.17
Year	2.96± NA	2.43± NA	2.56± NA	1.61± NA	**0.53**± NA
# best	2	0	2	1	**5**

design is intended for prediction interpretability: since humans already have a cognitive intuition for probabilities (Cosmides and Tooby, 1996), the output from a classification model is intended to convey a sense of likelihood for a particular class label. In other words, the predicted probability should reflect its confidence, *i.e.*, a level of certainty, in predicting such a label. Guo et al. (2017) provide the following example of a good classifier, whose output aligns with human intuition for probabilities: if the model outputs a probability prediction of 0.8, we hope it indicates that the model is 80% sure that its prediction is correct; given 100 predictions of 0.8, one shall expect roughly 80 of them are correct.

In that sense, a good classification algorithm not only can predict the correct label, but also can reflect the true correctness likelihood through its probability predictions, *i.e.*, providing calibrated confidence (Guo et al., 2017). To evaluate the level of miscalibration by a model, metrics like Expected Calibration Error (ECE) and Maximum Calibration Error (MCE) (Naeini et al., 2015) have been adopted in recent literature (Kristiadi et al., 2022; Rudner et al., 2021) for image classification tasks, and calibration methods like Platt scaling and isotonic regression have been developed to improve such alignment (Guo et al., 2017).

Note that these methods are all based on point estimate predictions by the classifier. Furthermore, these alignment metrics can only be computed at a subgroup level in practice, instead of at the instance level. In other words, one may not be able to make the claim with the existing classification framework such that *given a particular test instance*, how confident the classifier is in its prediction to be correct. We discuss our analysis in ECE with more details in Appendix A.16, which may help justify our motivation in introducing an alternative way to measure model confidence at the level of *individual test instances* in Section 4.2.1.

4.2.1 Predict with Instance Level Model Confidence via Generative Models

We propose the following framework to assess model confidence for its predictions *at the instance level*: for each test instance, we first sample N class prototype reconstructions by CARD through the classification version of Algorithm 2, and then perform the following computations:

1. We directly calculate the prediction interval width (PIW) between the 2.5^{th} and 97.5^{th} percentiles of the N reconstructed values for all classes, *i.e.*, with C different classes in total, we would obtain C PIWs for each instance;
2. We then convert the samples into probability space with Eq. (10), and apply paired two-sample t-test as an uncertainty estimation method proposed in Fan et al. (2021): we obtain the most and second most predicted classes for each instance, and test whether the difference in their mean predicted probability is statistically significant.

This framework would require the classifier to not produce the exact same output each time, since the goal is to construct prediction intervals for each of the class labels. Therefore, the class of generative models is a preferable modeling choice due to its ability to produce stochastic outputs, instead of just a point estimate by traditional classifiers.

In practice, we view each one-hot label as a class prototype in real continuous space (introduced in Section 2.2), and we use a generative model to reconstruct this prototype in a stochastic fashion. The intuition is that if the classifier is sure about the class that a particular instance belongs to, it would precisely reconstruct the original prototype vector without much uncertainty; otherwise, different class prototype reconstructions of the same test instance tend to have more variations: under the context of denoising diffusion models, given different samples from the prior distribution at timestep T, the label reconstructions would appear rather different from each other.

4.2.2 Classification with Model Confidence on CIFAR-10 Dataset

We demonstrate our experimental results on the CIFAR-10 dataset. We first contextualize the performance of CARD in conventional metrics including accuracy and NLL with other BNNs in ResNet-18 architecture in Table 4. The metrics of other methods were reported in Tomczak et al. (2021), a recent work in BNNs that proposes tighter ELBOs to improve variational inference performance and prior hyperparameter optimization. Following the recipe in Section 2.2, we first pre-train a deterministic classifier with the same ResNet-18 architecture, and achieve a test accuracy of 90.39%, with which we proceed to train CARD. We then obtain our instance prediction through majority vote, *i.e.*, the most predicted class label among its N samples for each image input, and achieve an improved test accuracy with a mean of 90.93% across 10 runs, showing its ability to improve test accuracy from the base classifier. Our NLL result is competitive among the best ones, even though the model is not optimized with a cross-entropy objective function, as we assume the class labels to be in the real continuous space.

Table 4: Comparison of accuracy (in $\%$) and NLL for CIFAR-10 classification with other BNNs.

Model	CMV-MF-VI	CM-MF-VI	CV-MF-VI	MF-VI	MC Dropout	MAP	CARD
Accuracy	86.25 ± 0.06	86.66 ± 0.24	79.78 ± 0.30	77.08 ± 1.14	83.64 ± 0.28	84.69 ± 0.35	$\mathbf{90.93 \pm 0.02}$
NLL	0.41 ± 0.00	$\mathbf{0.39 \pm 0.00}$	0.59 ± 0.00	0.68 ± 0.02	0.49 ± 0.00	0.93 ± 0.02	0.46 ± 0.00

We now present the results from one model run with the proposed framework in Section 4.2.1 for evaluating the instance-level prediction confidence. After obtaining the PIW and paired two-sample t-test ($\alpha = 0.05$) result from each test instance, we first split the test instances into two groups by the correctness of majority-vote predictions, then we obtain only the PIW corresponding to the true class for each instance, and compute the mean PIW of the true class within each group. In addition, we split the test instances by t-test rejection status, and compute the mean accuracy in each group. We report the results from these two grouping procedures in Table 5, where the metrics are computed across all test instances and at the level of each true class label.

Table 5: PIW (multiplied by 100) and t-test results for CIFAR-10 classification task.

Class	Accuracy	PIW		Accuracy by t-test Status	
		Correct	Incorrect	Rejected	Not-Rejected (Count)
All	90.95%	2.37	21.52	91.25%	42.86% (63)
1	91.00%	3.28	18.83	91.51%	45.45% (11)
2	96.00%	0.55	29.27	96.19%	33.33% (3)
3	87.30%	2.65	24.40	87.55%	25.00% (4)
4	81.90%	5.48	21.45	82.10%	63.64% (11)
5	93.30%	2.41	30.02	93.67%	20.00% (5)
6	84.70%	4.16	19.57	85.21%	46.15% (13)
7	94.20%	1.84	26.01	94.38%	33.33% (3)
8	92.80%	1.96	19.35	93.07%	25.00% (4)
9	95.30%	0.56	15.75	95.49%	33.33% (3)
10	93.00%	1.50	14.04	93.26%	50.00% (6)

We observe from Table 5 that under the scope of the entire test set, the mean PIW of the true class label among the correct predictions is narrower than that of the incorrect predictions by an order of magnitude, indicating that when CARD is making correct predictions, its class label reconstructions have much smaller variations. We may interpret such results as that CARD can reveal what it does not know through the relativity in reconstruction variations. Furthermore, when comparing the mean PIWs across different classes, we observe that the class with a higher prediction accuracy tends to have a sharper contrast in true label PIW between correct and incorrect predictions; additionally, the PIW values of both correct and incorrect predictions tend to be larger in a less accurate class. Meanwhile, it is worth noting that if we predict the class label by the one with the narrowest PIW for each instance, we can already obtain a test accuracy of 87.84%, suggesting a strong correlation between the prediction correctness and instance-level model confidence (in terms of label reconstruction variability). Moreover, we observe that the accuracy of test instances rejected by the t-test is much higher than that of the not-rejected ones, both across the entire test set and within each class.

We point out that these metrics can reflect how sure CARD is about the correctness of its predictions, and can thus be used as an important indicator of whether the model prediction *of each instance* can be trusted or not. Therefore, it has the potential to be further applied in the human-machine collaboration domain (Madras et al., 2018; Raghu et al., 2019; Wilder et al., 2020; Gao et al., 2021), such that one can apply such uncertainty measurement to decide if we can directly accept the model prediction, or we need to allocate the instance to humans for further evaluation.

5 Conclusion

In this paper, we propose Classification And Regression Diffusion (CARD) models, a class of conditional generative models that approaches supervised learning problems from a conditional generation perspective. Without training with objectives directly related to the evaluation metrics, we achieve state-of-the-art results on benchmark regression tasks. Furthermore, CARD exhibits a strong ability to represent the conditional distribution with multiple density modes. We also propose a new metric Quantile Interval Coverage Error (QICE), which can be viewed as a generalized version of negative log-likelihood in evaluating how well the model fits the data. Lastly, we introduce a framework to evaluate prediction uncertainty at the instance level for classification tasks.

Acknowledgments

The authors acknowledge the support of NSF IIS 1812699 and 2212418, and the Texas Advanced Computing Center (TACC) for providing HPC resources that have contributed to the research results reported within this paper.

References

Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020.

Jacob Austin, Daniel Johnson, Jonathan Ho, Danny Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Christopher Bishop. Mixture density networks. In *Aston University Neural Computing Research Group Report*, 1994.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

Glenn W. Brier. Verification of forecasts expressed in terms of probability. In *Monthly Weather Review*, volume 8, page 1, 1950.

Leda Cosmides and John Tooby. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgment under uncertainty. In *cognition*, volume 58(1), pages 1–73, 1996.

David R. Cox. Prediction by exponentially weighted moving averages and related methods. *Journal of the Royal Statistical Society: Series B (Methodological)*, 23(2):414–422, 1961.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Dheeru Dua and Casey Graff. UCI Machine Learning Repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Xinjie Fan, Shujian Zhang, Korawat Tanwisuth, Xiaoning Qian, and Mingyuan Zhou. Contextual dropout: An efficient sample-dependent dropout module. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

William Feller. On the theory of stochastic processes, with particular reference to applications. In *Proceedings of the 1st Berkeley Symposium on Mathematical Statistics and Probability*, pages 403–432. University of California Press, 1949.

Ethan Fetaya, Joern-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. Understanding the limitations of conditional generative models. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*. PMLR, 2016.

Yarin Gal, Jiri Hron, and Alex Kendall. Concrete dropout. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

Ruijiang Gao, Maytal Saar-Tsechansky, Maria De-Arteaga, Ligong Han, Min Kyung Lee, and Matthew Lease. Human-AI collaboration with bandit feedback. In *Proceedings of the 30th International Joint Conferences on Artificial Intelligence*, 2021.

Marta Garnelo, Dan Rosenbaum, Chris J. Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, 2018a.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S.M. Ali Eslami, and Yee Whye Teh. Neural processes. In *ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models*, 2018b.

https://doi.org/10.52202/068431-1316

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 2017.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Per Joachims. nnuncert: Uncertainty quantification with BNNs, 2021. URL `https://github.com/nnuncert/nnuncert`.

Zahra Kadkhodaie and Eero P. Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Bahjat Kawar, Michael Elad, Stefano Ermon, and Jiaming Song. Denoising diffusion restoration models. In *ICLR 2022 Workshop on Deep Generative Models for Highly Structured Data*, 2022.

Alex Kendall and Yarin Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

Durk P. Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, 2015.

Agustinus Kristiadi, Matthias Hein, and Philipp Hennig. Being a bit frequentist improves Bayesian neural networks. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics*, 2022.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st Conference on Neural Information Processing Systems*, 2017.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998. doi: 10.1109/5.726791.

Yuanzhi Liang, Linchao Zhu, Xiaohan Wang, and Yi Yang. A simple episodic linear probe improves visual recognition in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

https://doi.org/10.52202/068431-1316

Shiao Liu, Xingyu Zhou, Yuling Jiao, and Jian Huang. Wasserstein generative learning of conditional distribution. *arXiv preprint arXiv:2112.10039*, 2021.

Shiwei Liu, Tianlong Chen, Zahra Atashgahi, Xiaohan Chen, Ghada Sokar, Elena Mocanu, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Deep ensembling with no overhead for either training or testing: The all-round blessings of dynamic sparsity. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

Radek Mackowiak, Lynton Ardizzone, Ullrich Köthe, and Carsten Rother. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

David Madras, Toniann Pitassi, and Richard Zemel. Predict responsibly: Improving fairness and accuracy by learning to defer. In *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 2018.

Jishnu Mukhoti and Yarin Gal. Evaluating Bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv: 1811.12709*, 2018.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, 2015.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, page 113, 2011.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the 39th International Conference on Machine Learning*. PMLR, 2022.

Kushagra Pandey, Avideep Mukherjee, Piyush Rai, and Abhishek Kumar. DiffuseVAE: Efficient, controllable and high-fidelity generation from low-dimensional latents. *Transactions on Machine Learning Research*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.

Hieu Pham and Quoc V. Le. Autodropout: Learning dropout patterns to regularize deep networks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint arXiv:2204.06125*, 2022.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *Proceedings of the 6th International Conference on Learning Representations*, 2018.

Hongyu Ren, Shengjia Zhao, and Stefano Ermon. Adaptive antithetic sampling for variance reduction. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.

Michael Revow, Christopher KI Williams, and Geoffrey E Hinton. Using generative models for handwritten digit recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):592–606, 1996.

https://doi.org/10.52202/068431-1316

Tim G. J. Rudner, Zonghao Chen, Yee Whye Teh, and Yarin Gal. Tractable function-space variational inference in Bayesian neural networks. In *ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning*, 2021.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the 9th International Conference on Learning Representations*, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations*, 2021c.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Marcin B. Tomczak, Siddharth Swaroop, Andrew Y. K. Foong, and Richard E. Turner. Collapsed variational bounds for Bayesian neural networks. In *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Zhendong Wang and Mingyuan Zhou. Thompson sampling via local uncertainty. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 2020.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*. PMLR, 2011.

Bryan Wilder, Eric Horvitz, and Ece Kamar. Learning to complement humans. In *Proceedings of the 29th International Joint Conferences on Artificial Intelligence*, 2020.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Shentao Yang, Zhendong Wang, Huangjie Zheng, Yihao Feng, and Mingyuan Zhou. A regularized implicit policy for offline reinforcement learning. *arXiv preprint arXiv:2202.09673*, 2022.

Jiayu Yao, Weiwei Pan, Soumya Ghosh, and Finale Doshi-Velez. Quality of uncertainty quantification for Bayesian neural network inference. In *ICML 2019 Workshop on Uncertainty and Robustness in Deep Learning*, 2019.

Huangjie Zheng, Xu Chen, Jiangchao Yao, Hongxia Yang, Chunyuan Li, Ya Zhang, Hao Zhang, Ivor Tsang, Jingren Zhou, and Mingyuan Zhou. Contrastive attraction and contrastive repulsion for representation learning. *arXiv preprint arXiv:2105.03746*, 2021.

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models and diffusion-based adversarial auto-encoders. *arXiv preprint arXiv:2202.09671*, 2022.

Xingyu Zhou, Yuling Jiao, Jin Liu, and Jian Huang. A deep generative approach to conditional sampling. *Journal of the American Statistical Association*, pages 1–28, 2021.

https://doi.org/10.52202/068431-1316

Roland S. Zimmermann, Lukas Schott, Yang Song, Benjamin A. Dunn, and David A. Klindt. Score-based generative classifiers. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Appendix B
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix B
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See Appendix and URL in Abstract
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix C
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-1316