Evaluating Aleatoric Uncertainty via Conditional Generative Models

Ziyi Huang, Henry Lam, Haofeng Zhang
Columbia University
New York, NY 10027
`zh2354, khl2114, hz2553@columbia.edu`

Abstract

Aleatoric uncertainty quantification seeks for distributional knowledge of random responses, which is important for reliability analysis and robustness improvement in machine learning applications. Previous research on aleatoric uncertainty estimation mainly targets closed-formed conditional densities or variances, which requires strong restrictions on the data distribution or dimensionality. To overcome these restrictions, we study conditional generative models for aleatoric uncertainty estimation. We introduce two metrics to measure the discrepancy between two conditional distributions that suit these models. Both metrics can be easily and unbiasedly computed via Monte Carlo simulation of the conditional generative models, thus facilitating their evaluation and training. We demonstrate numerically how our metrics provide correct measurements of conditional distributional discrepancies and can be used to train conditional models competitive against existing benchmarks.

1 Introduction

Uncertainty quantification plays a pivotal role in machine learning systems, especially for downstream decision-making tasks involving reliability analysis and optimization. There are two major types of uncertainty, *aleatoric* uncertainty and *epistemic* uncertainty. Aleatoric uncertainty refers to the intrinsic stochasticity of the output given a specific input [21], while epistemic uncertainty accounts for the model uncertainty caused by data and modeling limitations [24]. Most classical machine learning algorithms that focus on mean response prediction primarily address epistemic uncertainty, but aleatoric uncertainty, which describes the distribution of responses beyond summary statistics like the mean, has been gaining importance because of risk and safety-critical considerations.

Existing approaches for aleatoric uncertainty estimation can be largely divided into the following directions: negative log-likelihood (NLL) loss-based estimation, forecaster calibration, and conditional density estimation (CDE). While powerful, these approaches are limited by several drawbacks arising from real-world applications:

1. *Negative Log-Likelihood Loss:* In regression tasks, aleatoric uncertainty can be estimated through the conditional mean and variance from models (heteroscedastic neural networks) optimized by the NLL loss [41, 2, 4, 31, 24, 5]. However, this approach requires scalar-type output, which cannot be easily extended to broader computer vision applications, such as image generation. In addition, the computation of NLL loss relies on assumptions of conditional Gaussian or Gaussian-like distribution, which may not be followed by real-world datasets.

2. *Forecaster Calibration:* In the calibration literature, aleatoric uncertainty estimators are also known as forecasters [12, 26, 49] with multiple definitions of calibration modes [12, 49, 8, 26, 5]. Under these definitions, the ground-truth conditional distribution function is well calibrated,

but not vice versa. Thus, some intuitive sharpness criteria are typically applied to avoid trivial forecasters such as the unconditional distribution. However, little is known about how to recover the ground-truth conditional distribution function via calibration, even asymptotically.

3. *Conditional Density Estimation:* In CDE-based approaches [20, 22, 52, 45, 46, 6, 7], aleatoric uncertainty is directly calculated by estimating conditional densities in a certain form (such as kernel density). Most of CDE methods can only apply on low-dimensional responses following absolutely continuous conditional distributions. Moreover, the output of CDE methods is an explicit formula of the conditional density function. Thus, numerical characteristics such as conditional quantiles may be hard to obtain, as it involves numerical integration that is generally difficult to implement, especially in higher-dimensional settings.

To address the above challenges, we study a framework using conditional generative models to estimate aleatoric uncertainty. Compared to previous approaches, conditional generative models [38, 47] are more scalable and flexibly applicable regardless of the dimension and distribution of the input/output vector. Moreover, they can easily generate numerical characteristics of the underlying distributions or other performance estimations through Monte Carlo methods.

At the core of our framework is the construction of distance metrics between the generative model and the ground-truth distribution, which is required for both model evaluation and training [13, 42, 1]. In particular, we generalize the maximum mean discrepancy (MMD) [14, 34] to the setting of conditional distributions, by constructing two new metrics that we call joint maximum mean discrepancy (JMMD) and average maximum mean discrepancy (AMMD). We derive statistical properties in estimating these metrics and illustrate that both metrics admit easy-to-implement and computationally scalable unbiased estimators. Based on these, we further develop two approaches to optimize conditional generative models suited for different tasks and conduct comprehensive experiments to show the correctness and effectiveness of our framework.

Our approach has the following strengths relative to previous methods: 1) A similar study with conditional MMD can be found in [47] which, as far as we know, is the most relevant work on MMD-based conditional generative models. However, their framework involves unrealistic technical assumptions that may not hold in practice, as well as matrix inversion operations that suffer from instability and scalability issues (see Section 4.1). 2) Both JMMD and AMMD are evaluation metrics that are desirably "distribution-free" (i.e., the data are not assumed any particular type of distributions) and "model-free" (i.e., the evaluation does not involve additional estimated models such as the discriminator). In previous research, Fréchet Inception Distance (FID) [19, 36] is a standard metric to assess the quality of unconditional generative models. However, the closed-form computation of FID assumes that both generative models and data follow multivariate Gaussian distributions. Another commonly used evaluation approach is Indirect Sampling Likelihood (ISL) [3, 13], which computes the NLL under a fitted kernel density based on generative models. However, kernel density estimation deteriorates in quality when dimensionality increases and could fit poorly into the generative models. Finally, the value of loss on testing data is an alternative for performance examination. However, typical losses such as using f-divergence or Wasserstein distance cannot indicate the performance of the generator alone (see Section 2).

2 Related Work

Learning and Evaluation Criterion. Evaluation criteria on generative models against data are typically borrowed from discrepancy measures between two probability distributions in the statistics literature. The latter includes two major types: f-divergence and integral probability metrics. The seminal paper [13] used Jensen-Shannon divergence in its original form and then [42] extended it to general f-divergence motivated from the benefits of other divergence function choices. The computation of integral probability metrics have two important sub-directions, MMD [34, 33] and Wasserstein distance [1, 15]. Among these criteria, a discriminator is typically needed for approaches with f-divergence (variational representation) and Wasserstein distance (dual representation), while it is not required for MMD methods. The loss function from f-divergence and Wasserstein distance cannot be directly use to evaluate generative models alone due to their dependency on the quality of discriminators. In addition, other conditional distance measures may encounter challenges when using generative models. For instance, NLL value [31] and CDE value [6] require an explicit form of the model's density function.

Aleatoric Uncertainty in Deep Learning. Besides the directions discussed in Section 1, aleatoric uncertainty on classification tasks can be estimated from the output of softmax layers in deep neural networks [40, 31, 17]. Previous research [16] pointed out that directly using softmax outputs for estimation could be inaccurate, as the softmax probability on predicted class did not reflect the ground-truth correctness likelihood. The ground-truth conditional mass function has zero calibration error but not vice versa. Hence forecasters with zero calibration error, which have been studied extensively [30, 37, 29], are not sufficient to recover the ground-truth conditional mass function. The forecasters could be heuristically improved by a second-level metric named sharpness (or refinement error) [30, 27, 28]. Since aleatoric uncertainty in classification can be captured by vector-valued maps such as softmax responses, it is not necessary to use a conditional generative model for this task, and thus we do not focus on classification in this paper.

3 Conditional Generative Models and Maximum Mean Discrepancy

3.1 Conditional Generative Models

In this section, we provide rigorous definitions on conditional generative models. Consider a standard statistical framework where a pair of random vectors $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ follows a joint distribution $P_{X,Y}$ with marginal distributions $X \sim P_X$ and $Y \sim P_Y$. We assume the space $\mathcal{X} \subset \mathbb{R}^d$ with $d \geq 1$ which is allowed to contain either continuous or discrete components. Denote the conditional distribution of Y given X by $P_{Y|X}$. For a given value x of X, denote the conditional distribution as $P_{Y|X=x}$. Typically, we regard X as a vector of input (example) and Y as a vector of output (label). For instance, $Y \subset \mathbb{R}^q$ with $q \geq 1$ in regression and $Y \subset [K] := \{1, \ldots, K\}$ in classification. Alternatively, in image generation tasks, X refers to auxiliary information (such as the image attributes or labels) and Y refers to the image in order to keep the notation consistent.

Our goal is to quantify the conditional distribution $P_{Y|X}$ via conditional generative models. More precisely, let $\xi \in \mathbb{R}^m$ be a random vector independent of X with a known distribution P_ξ (specified by the learner) and the goal is to construct a function $G : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ such that the conditional distribution of $G(\xi, X)|X = x$ is the same as $P_{Y|X=x}$. The following lemma demonstrates the existence of such function G, termed as the conditional generative model $G : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$.

Lemma 1 (Adapted from Theorem 5.10 in [23]). *Let (X, Y) be a random pair taking values in $\mathcal{X} \times \mathcal{Y}$ with joint distribution $P_{X,Y}$. Suppose Y is a standard Borel space. Then there exist a random vector $\xi \sim P_\xi = Uniform([0, 1]^m)$ and a Borel-measurable function $G : \mathbb{R}^m \times \mathcal{X} \to \mathcal{Y}$ such that ξ is independent of X and $(X, Y) = (X, G(\xi, X))$ almost everywhere. In particular, such G satisfies that $Y|X = x \sim G(\xi, X)|X = x$ for a.e. x with respect to P_X.*

The conditional generative model can provide more information than standard regression models with single-point prediction. In regression problems, the conditional mean can be estimated by taking the sample mean of multiple draws from $G(\xi_i, X)|X = x$. Meanwhile, other numerical characteristics of the underlying target distribution, such as conditional variance and conditional quantile can also be estimated by Monte Carlo sampling from the conditional generative model, beyond what single-point prediction could offer.

In the rest of this paper, we use $P_{Y|X}$ for the ground-truth conditional distribution and $Q_{Y|X}$ for the distribution of the conditional generative model $G(\xi, X)|X$. We denote $Q_{X,G(\xi,X)}$ as the joint distribution of $(X, G(\xi, X))$. For each given x, the generative model is able to generate conditionally independent and identically distributed (i.i.d.) samples $G(\xi_i, x)$ from the conditional distribution $Q_{Y|X=x}$. We parametrize the conditional generative model in a hypothesis class $\{G_\theta(\xi, X) : \theta \in \Theta\}$ with parameter θ. To learn $G(\xi, X)|X$ as an estimate of $P_{Y|X}$, we need a metric to quantify the difference between $G(\xi, X)|X$ and $P_{Y|X}$ using finite training data, which relates to Two-Sample Test. To this end, we will use the (kernel) maximum mean discrepancy (MMD) [14], which is described in the next subsection.

3.2 Two-Sample Test via Maximum Mean Discrepancy

We review the standard MMD in the setting of unconditional distribution on $\mathcal{Y}$. Section A provides preliminaries on the reproducing kernel Hilbert space (RKHS). Suppose that $\mathcal{F}_X$ ($\mathcal{F}_Y$) is the RKHS defined on the space $\mathcal{X}$ ($\mathcal{Y}$) with kernel k_1 (k_2) and feature map ϕ_1 (ϕ_2). We adopt the following two

basic assumptions throughout this paper (i.e., all theorems make these assumptions without explicit mentioning). Detailed explanations on Assumptions 1 and 2 can be found in Section A. The Gaussian kernels for instance satisfy both assumptions.

Assumption 1. *We assume the following: 1) $k_1(\cdot, \cdot)$ is measurable and $\mathbb{E}_{x \sim P_X}[k_1(x, x)] < \infty$. 2) $k_2(\cdot, \cdot)$ is measurable and $\mathbb{E}_{y \sim P_Y}[k_2(y, y)] < \infty$. Moreover, $\mathbb{E}_{y \sim P_{Y|X=x}}[k_2(y, y)|X = x] < \infty$ for any $x \in \mathcal{X}$. In addition, these assumptions also hold when replacing the data distribution P by the generative distribution Q.*

Assumption 2. *We assume the following: 1) k_1 is characteristic. 2) k_2 is characteristic. 3) $k_1 \otimes k_2$ is characteristic.*

The integral probability metric aims to measure the discrepancy between two distributions. Let $\mathcal{G}$ denote a set of functions $\mathcal{Y} \to \mathbb{R}$. Given two distributions P_Y and Q_Y on $\mathcal{Y}$, the integral probability metric is defined as

$$IPM(P_Y, Q_Y) = \sup_{f \in \mathcal{G}} |\mathbb{E}[f(Y)] - \mathbb{E}[f(\hat{Y})]|$$

where $Y \sim P_Y$ and $\hat{Y} \sim Q_Y$. MMD is a special case of integral probability metrics, as it chooses $\mathcal{G}$ to be the unit ball in the RKHS $\mathcal{F}_Y$. Let μ_{P_Y} denote the kernel mean embedding of P_Y in $\mathcal{F}_Y$: $\mu_{P_Y} := \mathbb{E}_{y \sim P_Y}[\phi_2(y)]$. μ_{P_Y} is guaranteed to be an element in the RKHS $\mathcal{F}_Y$ under Assumption 1 [14]. With these discussions, the square of MMD distance between P_Y and Q_Y is formally defined as

$$MMD^2(P_Y, Q_Y) = \sup_{f \in \mathcal{F}_Y, \|f\|_{\mathcal{F}_Y} \leq 1} |\mathbb{E}[f(Y_1)] - \mathbb{E}[f(\hat{Y}_1)]|^2$$

$$= \|\mu_{P_Y} - \mu_{Q_Y}\|_{\mathcal{F}_Y}^2 = \mathbb{E}[k_2(Y_1, Y_2)] - 2\mathbb{E}[k_2(Y_1, \hat{Y}_1)] + \mathbb{E}[k_2(\hat{Y}_1, \hat{Y}_2)] \tag{1}$$

where $Y_1, Y_2 \overset{i.i.d.}{\sim} P_Y$ and $\hat{Y}_1, \hat{Y}_2 \overset{i.i.d.}{\sim} Q_Y$.

Theorem 2 ([14]). *$MMD^2(P_Y, Q_Y) \geq 0$ and $MMD^2(P_Y, Q_Y) = 0$ if and only if $P_Y = Q_Y$.*

Suppose we have data $y_i \overset{i.i.d.}{\sim} P_Y$ ($i \in [n]$) and $\hat{y}_j \overset{i.i.d.}{\sim} Q_Y$ ($j \in [m]$). Then a standard unbiased estimator of $MMD^2(P_Y, Q_Y)$ [14] is

$$\mathcal{L}_{MMD^2}^u = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{i' \neq i, i'=1}^{n} k_2(y_i, y_{i'}) - \frac{2}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} k_2(y_i, \hat{y}_j) + \frac{1}{m(m-1)} \sum_{j=1}^{m} \sum_{j' \neq j, j'=1}^{m} k_2(\hat{y}_j, \hat{y}_{j'})$$

4 Generalization to Conditional Two-Sample Test

In this section, we generalize MMD to conditional Two-Sample Test. We first explain the limitation of conditional maximum mean discrepancy (CMMD), the state-of-the-art approach to use MMD for conditional models [47]. Then, we introduce two metrics, JMMD and AMMD, which bypass the limitations of CMMD on strong restrictions and biased estimation. We also present the connections and comparisons among these metrics, and describe how to use them to construct conditional generative models.

4.1 Previous Work on Conditional Maximum Mean Discrepancy

In [47], conditional generative moment-matching networks (CGMMNs) were developed for conditional distribution generation. In particular, they leveraged previous work on conditional mean embeddings of the conditional distribution $C_{P_{Y|X}}$ [51, 9, 50, 39] (Section A provides a review on this topic.) They used the discrepancy between $C_{P_{Y|X=x}}$ and $C_{Q_{Y|X=x}}$ to measure the difference of two conditional distributions, termed as CMMD, which is defined formally as:

$$CMMD^2 = \|C_{P_{Y|X}} - C_{Q_{Y|X}}\|_{\mathcal{F}_X \otimes \mathcal{F}_Y}^2 \tag{2}$$

where P represents the ground-truth data distribution and Q represents the generative distribution. The estimator of $CMMD^2$ developed in [47] is as follows:

$$\mathcal{L}_{C^2}(P, Q) = \|\tilde{C}_{P_{Y|X}} - \tilde{C}_{Q_{Y|X}}\|_{\mathcal{F}_X \otimes \mathcal{F}_Y}^2, \tag{3}$$

4

$$\tilde{C}_{P_{Y|X}} = \tilde{C}_{PYX}(\tilde{C}_{PXX} + \lambda I)^{-1} = \Phi_2(K_X + \lambda nI)^{-1}\Phi_1{}^1 \tag{4}$$

where $\Phi_2 = (\phi_2(y_1), ..., \phi_2(y_n))$, $\Phi_1 = (\phi_1(x_1), ..., \phi_1(x_n))$, $K_X = \Phi_1^\top \Phi_1$, and I is the identity matrix.

While [47] is the most relevant study to our problem setting, directly applying CMMD on aleatoric uncertainty estimation has following limitations:

Computationally Expensive: The matrix inverse in the estimator is computationally expensive for practical implementation. The running time for a single inversion in one iteration is at the order of $O(B^3)$, where B is the batch size. Meanwhile, the batch size should be sufficient large to achieve good performance for generative models [34].

Strong Technical Assumptions and Existence of Inversion: 1) The existence of the conditional mean embedding operator $C_{P_{Y|X}}$ typically requires strong assumptions: $\forall g \in \mathcal{F}_Y$, $\mathbb{E}_{P_{Y|X}}[g(Y)|X] \in \mathcal{F}_X$. This assumption is not necessarily true for continuous domains [51], and simple counterexamples using the Gaussian kernel can be found [9]. 2) In general, $\tilde{C}_{XX}^{-1}$ does not exist when $\mathcal{F}_X$ is infinite dimensional, since $\tilde{C}_{XX}$ is a compact operator and thus has an arbitrary small positive eigenvalue [39]. When the matrix is singular, the matrix inversion could be unstable and the performance of the estimator $\tilde{C}_{P_{Y|X}}$ in [47] might be degraded after adding λI to avoid the singularity. 3) Even though the first two points could be relieved in some sense [43], the CMMD metric (2) is well-defined only if $C_{P_{Y|X}}, C_{Q_{Y|X}} \in \mathcal{F}_X \otimes \mathcal{F}_Y$. However, this requires a much stronger assumption than the existence of $\tilde{C}_{XX}^{-1}$ (See Assumption 6 in Section A).

Bias: CMMD does not admit any obvious unbiased estimator. The estimator $\tilde{C}_{P_{Y|X}}$ in (4) is biased, even in the asymptotic sense if λ is fixed [51].

To bypass the above limitations, we propose two alternative metrics which only require basic assumptions on the existence of the cross-covariance operator and the characteristic property of the kernels (Assumptions 1 and 2). In particular, we do not require the existence of the inversion of any operator or matrix, which makes our metrics easily implemented for real-world applications.

4.2 Average Maximum Mean Discrepancy (AMMD)

We first introduce a rather straightforward approach, which we term the AMMD metric. AMMD shows better potential for multi-output problems (such as image generation) where data consists of i.i.d. inputs x_i with conditionally independent outputs $y_{i,j}$ at each x_i; see Section C for a more detailed discussion. In AMMD, at each x, we use (1) to build unbiased estimators of the MMD on the conditional distribution of $Y|X = x$. Then, these estimators are averaged with respect to the marginal P_X. More specifically, we define

$$AMMD^2(P, Q) = \mathbb{E}_{x \sim P_X}[MMD_{X=x}^2(P_{Y|X=x}, Q_{Y|X=x})]$$

where

$$MMD_{X=x}^2(P_{Y|X=x}, Q_{Y|X=x}) := \|\mu_{P_{Y|X=x}} - \mu_{Q_{Y|X=x}}\|_{\mathcal{F}_Y}^2$$
$$= \mathbb{E}[k_2(Y_1^x, Y_2^x)|X = x] - 2\mathbb{E}[k_2(Y_1^x, \hat{Y}_1^x)|X = x] + \mathbb{E}[k_2(\hat{Y}_1^x, \hat{Y}_2^x)|X = x] \tag{5}$$

is a function of x and for fixed x, $Y_1^x, Y_2^x \overset{i.i.d.}{\sim} P_{Y|X=x}$, and $\hat{Y}_1^x, \hat{Y}_2^x \overset{i.i.d.}{\sim} Q_{Y|X=x}$. Note that $\mu_{P_{Y|X=x}}, \mu_{Q_{Y|X=x}} \in \mathcal{F}_Y$ guaranteed by Assumption 1 so $MMD_{X=x}^2$ is well-defined. Hence $AMMD^2(P, Q)$ is also well-defined being the expectation with respect to non-negative measurable functions.

Remark 3. *In* (5)*, Y_1^x, Y_2^x are drawn in a **conditionally independent** manner for each x. This is not equivalent to globally draw two unconditionally independent samples Y_1, Y_2 and consider $Y_1|X = x, Y_2|X = x$ for each x because the latter is not conditionally independent in general. Therefore, we have that in general, $\mathbb{E}_{x \sim P_X}[\mathbb{E}[k_2(Y_1^x, Y_2^x)|X = x]] \neq \mathbb{E}[k_2(Y_1, Y_2)]$.*

Theorem 4. *$AMMD^2(P, Q) \geq 0$ and $AMMD^2(P, Q) = 0$ if and only if for a.e. x with respect to P_X, $P_{Y|X=x} = Q_{Y|X=x}$.*

---

[1]The sample size n should appear in the formula [39] but seems missing in the paper [47].

Theorem 4 shows that $AMMD^2(P, Q)$ offers a metric to measure $P_{Y|X}$ versus $Q_{Y|X}$. Next, we propose a Monte Carlo estimator of $AMMD^2$ for conditional generative models:

1. Take a batch $\{(x_i, y_{i,l}) : i \in [n], l \in [r]\}$ from P of batch size rn where $y_{i,l}$ ($l \in [r]$) are the outputs at the same x_i. Here, r is restricted by the specification of the task: $r = 1$ in single-output problems but can be greater than 1 in multi-output problems such as image generation; $n \geq 1$.

2. Generate a batch $\{(x_i, G(\xi_{i,j}, x_i)) : i \in [n], j \in [m]\}$ from Q of batch size mn where $\xi_{1,1}, \ldots, \xi_{n,m}$ are i.i.d. and independent of $x_1, \ldots, x_n$; $m \geq 2$.

3. Compute

$$\hat{A}^2(P, Q) = \frac{1}{n} \sum_{i=1}^{n} \Big(- \frac{2}{mr} \sum_{j=1}^{m} \sum_{l=1}^{r} k_2(y_{i,l}, G(\xi_{i,j}, x_i))$$
$$+ \frac{1}{m(m-1)} \sum_{j=1}^{m} \sum_{j' \neq j, j'=1}^{m} k_2(G(\xi_{i,j}, x_i), G(\xi_{i,j'}, x_i)) \Big) \qquad (6)$$

The next theorem establishes the error analysis of the estimator $\hat{A}^2(P, Q)$.

Theorem 5. $\hat{A}^2(P, Q)$ *is an unbiased estimator of* $AMMD^2(P, Q) - C_0$ *where* C_0 *is a constant independent of* Q *given by* $C_0 = \mathbb{E}_{x \sim P_X}[\mathbb{E}[k_2(Y_1^x, Y_2^x)|X = x]]$. *Moreover, the variance of* $\hat{A}^2(P, Q)$ *is* $O(\frac{1}{n \min\{m, r\}}) + \frac{1}{n} K_0$ *where* $K_0 = Var_{x \sim P_X}[-2\mathbb{E}[k_2(Y_1^x, \hat{Y}_1^x)|X = x] + \mathbb{E}[k_2(\hat{Y}_1^x, \hat{Y}_2^x)|X = x]]$ *is independent of* n, m, r. *(The explicit formula of the variance is given in Section B.)*

C_0 in Theorem 5 is free of the conditional generative model and thus does not need to be embodied at the training/evaluation phase. This is in the same spirit of NLL which is formed by a free-of-model constant and the Kullback–Leibler divergence between the model and data. Theorem 5 shows that if n is not allowed to be large (e.g., n is bounded above by the number of class in the label-based image generation problems), the variance of the estimator $\hat{A}^2(P, Q)$ should be reduced by increasing m and r. On the other hand, if n is allowed to be large (e.g., in regression problems with continuous features), then given a fixed computational budget, we should increase n while maintaining the small possible values $m = 2$ and $r = 1$ to reduce the variance of $\hat{A}^2(P, Q)$ efficiently.

4.3 Joint Maximum Mean Discrepancy (JMMD)

We then introduce the JMMD metric, which is based on the joint distribution. Compared with AMMD, JMMD is more suitable for single-output tasks (such as regression) where data consists of joint i.i.d. samples (x_i, y_i); see Section C for a more detailed discussion. According to the observation in Lemma 1, the matching of $Q_{Y|X=x}$ (the conditional distribution of $G(\xi, X)|X = x$) with $P_{Y|X=x}$ for a.e. x can be transferred to the matching of $Q_{X,Y}$ (the joint distribution of $(X, G(\xi, X))$) with $P_{X,Y}$. This motivates us to define the following metric which we term as JMMD:

$$JMMD^2(P, Q) = MMD^2(P_{X,Y}, Q_{X,Y})$$
$$= \mathbb{E}[k_3((X_1, Y_1), (X_2, Y_2))] - 2\mathbb{E}[k_3((X_1, Y_1), (\hat{X}_1, \hat{Y}_1))] + \mathbb{E}[k_3((\hat{X}_1, \hat{Y}_1), (\hat{X}_2, \hat{Y}_2))]$$

where $k_3 = k_1 \otimes k_2$ is the kernel of the tensor product space $\mathcal{F}_X \otimes \mathcal{F}_Y$ and $(X_1, Y_1), (X_2, Y_2) \overset{i.i.d.}{\sim} P_{X,Y}$ and $(\hat{X}_1, \hat{Y}_1), (\hat{X}_2, \hat{Y}_2) \overset{i.i.d.}{\sim} Q_{X,Y}$. Note that $JMMD^2(P, Q)$ can be viewed alternatively as the discrepancy of the cross-covariance operators $C_{P_{YX}}, C_{Q_{YX}}$ defined on the tensor product space $\mathcal{F}_X \otimes \mathcal{F}_Y$. Since $C_{P_{YX}}, C_{Q_{YX}} \in \mathcal{F}_X \otimes \mathcal{F}_Y$ guaranteed by Assumption 1, $JMMD^2(P, Q)$ is well-defined (see Section A for more details).

Theorem 6. $JMMD^2(P, Q) \geq 0$ *and* $JMMD^2(P, Q) = 0$ *if and only if for a.e.* x *with respect to* P_X, $P_{Y|X=x} = Q_{Y|X=x}$.

Theorem 6 shows that $JMMD^2(P, Q)$ offers a metric to measure $P_{Y|X}$ versus $Q_{Y|X}$. In parallel to AMMD, we propose a Monte Carlo estimator of $JMMD^2$ for conditional generative models: Take a batch of samples from P of batch size r $\{(x_l, y_l) : l \in [r]\}$; $r \geq 2$. Generate a batch

---

Algorithm 1 Algorithm Framework of A-CGM

    Input: Training Dataset $\mathcal{D} = \{(x_i, y_i) : i \in \mathcal{I}\}$.
    Output: Finalized parameters θ in the generative model $G_\theta(\xi, x)$.
 1: Randomly divide the training dataset $\mathcal{D}$ into mini batches.
 2: **for** $t = 0, \ldots, T-1$ **do**
 3:    Set $\mathcal{B}^G = \emptyset$
 4:    **for** each mini batch $\mathcal{B}$ in $\mathcal{D}$ **do**
 5:        **for** each $x \in \mathcal{B}$ **do**
 6:            Draw multiple i.i.d. copies $\xi_1, \ldots, \xi_m$ from P_ξ
 7:            Generate conditional samples by forward-propagating through $G_\theta(\xi_j, x)$
 8:            Add $(x, G_\theta(\xi_1, x)), \ldots, (x, G_\theta(\xi_m, x))$ into $\mathcal{B}^G$
 9:        **end for**
10:        Optimize θ by $\hat{A}^2(P, Q_\theta)$ in (6) based on $\mathcal{B}$ and $\mathcal{B}^G$
11:    **end for**
12: **end for**

---

from Q of batch size m $\{(\hat{x}_j, G(\xi_j, \hat{x}_j)) : j \in [m]\}$ where $\xi_1, \ldots, \xi_m$ are i.i.d. and independent of $x_1, \ldots, x_r, \hat{x}_1, \ldots, \hat{x}_m$; $m \geq 2$. Compute

$$\hat{J}^2(P, Q) = -\frac{2}{mr} \sum_{j=1}^{m} \sum_{l=1}^{r} k_3((x_l, y_l), (\hat{x}_j, G(\xi_j, \hat{x}_j)))$$

$$+ \frac{1}{m(m-1)} \sum_{j=1}^{m} \sum_{j' \neq j, j'=1}^{m} k_3((\hat{x}_j, G(\xi_j, \hat{x}_j)), (\hat{x}_{j'}, G(\xi_{j'}, \hat{x}_{j'}))) \tag{7}$$

The next theorem establishes the error analysis of the estimator $\hat{J}^2(P, Q)$.

Theorem 7. $\hat{J}^2(P, Q)$ *is an unbiased estimator of* $JMMD^2(P, Q) - C_1$ *where* C_1 *is a constant independent of* Q *given by* $C_1 = \mathbb{E}[k_3((X_1, Y_1), (X_2, Y_2))]$. *Moreover, the variance of* $\hat{J}^2(P, Q)$ *is* $O(\frac{1}{\min\{m,r\}})$. *(The explicit formula of the variance is given in Section B.)*

C_1 in Theorem 7 is free of the conditional generative model. Theorem 7 shows that the variance of $\hat{J}^2(P, Q)$ is decreasing at the order of $\frac{1}{\min\{m,r\}}$. Therefore, given a fixed computational budget B, we should set $m = \Theta(r) = \Theta(\sqrt{B})$ to achieve the minimum variance of the estimator $\hat{J}^2(P, Q)$.

4.4 Connections and Comparisons among Metrics

We establish the theoretical connections among CMMD, JMMD, and AMMD as follows.

Theorem 8. *Suppose that* $AMMD$ *and* $JMMD$ *are well-defined. Then we have that*

$$JMMD^2 \leq \mathbb{E}_{x \sim P_X}[k_1(x, x)]AMMD^2.$$

Theorem 9. *Suppose that* $AMMD$ *is well-defined. Moreover, suppose that for all* $g \in \mathcal{F}_Y$, $\mathbb{E}_{P_{Y|X}}[g(Y)|X] \in \mathcal{F}_X$ *and* $\mathbb{E}_{Q_{Y|X}}[g(Y)|X] \in \mathcal{F}_X$ *so that the conditional mean embeddings* $C_{P_{Y|X}}$, $C_{Q_{Y|X}}$ *are well-defined. Furthermore, we assume* $C_{P_{Y|X}} \in \mathcal{F}_X \otimes \mathcal{F}_Y$, $C_{Q_{Y|X}} \in \mathcal{F}_X \otimes \mathcal{F}_Y$ *so that CMMD* (2) *is well-defined. Then we have*

$$AMMD^2 \leq \mathbb{E}_{x \sim P_X}[k_1(x, x)]CMMD^2.$$

We further highlight the strengths of AMMD and JMMD on conditional generative model evaluation which is a challenging task due to delicacies of evaluation at the conditional distribution level. First, both metrics are "distribution-free", i.e., the data or conditional generative model are not restricted to a specific type of distributions. In contrast, FID for instance requires the Gaussian assumption. Second, they are "model-free", i.e., their evaluation does not involve additional estimated models beyond the conditional generative model itself, such as kernel density estimators in Indirect Sampling Likelihood [3, 13] or estimated discriminators [42, 1].

Dataset	CGMMN			J-CGM			A-CGM		
	JMMD	AMMD	FID	JMMD	AMMD	FID	JMMD	AMMD	FID
Boston	$1.92 \cdot 10^{-2}$	$\mathbf{8.41 \cdot 10^{-5}}$	$1.35 \cdot 10^{-2}$	$\mathbf{2.92 \cdot 10^{-4}}$	$8.49 \cdot 10^{-4}$	$7.84 \cdot 10^{-3}$	$3.17 \cdot 10^{-4}$	$8.74 \cdot 10^{-5}$	$\mathbf{5.20 \cdot 10^{-3}}$
Concrete	$9.57 \cdot 10^{-3}$	$1.67 \cdot 10^{-4}$	$\mathbf{8.64 \cdot 10^{-3}}$	$\mathbf{1.53 \cdot 10^{-4}}$	$1.78 \cdot 10^{-4}$	$9.86 \cdot 10^{-3}$	$2.47 \cdot 10^{-4}$	$\mathbf{1.99 \cdot 10^{-5}}$	$9.74 \cdot 10^{-3}$
Energy	$1.87 \cdot 10^{-2}$	$1.40 \cdot 10^{-4}$	$\mathbf{7.96 \cdot 10^{-3}}$	$3.16 \cdot 10^{-4}$	$9.45 \cdot 10^{-4}$	$9.16 \cdot 10^{-3}$	$\mathbf{3.09 \cdot 10^{-4}}$	$\mathbf{1.23 \cdot 10^{-4}}$	$9.38 \cdot 10^{-3}$
Wine	$1.09 \cdot 10^{-2}$	$3.02 \cdot 10^{-4}$	$1.01 \cdot 10^{-2}$	$\mathbf{1.14 \cdot 10^{-4}}$	$3.61 \cdot 10^{-4}$	$9.94 \cdot 10^{-3}$	$1.16 \cdot 10^{-4}$	$\mathbf{2.72 \cdot 10^{-4}}$	$\mathbf{9.86 \cdot 10^{-3}}$
Yacht	$1.28 \cdot 10^{-2}$	$5.76 \cdot 10^{-5}$	$1.11 \cdot 10^{-2}$	$6.60 \cdot 10^{-4}$	$4.75 \cdot 10^{-4}$	$1.13 \cdot 10^{-2}$	$\mathbf{1.67 \cdot 10^{-4}}$	$\mathbf{4.63 \cdot 10^{-5}}$	$\mathbf{7.68 \cdot 10^{-3}}$
Kin8nm	$1.00 \cdot 10^{-2}$	$1.46 \cdot 10^{-3}$	$1.41 \cdot 10^{-2}$	$1.10 \cdot 10^{-4}$	$1.39 \cdot 10^{-3}$	$\mathbf{9.69 \cdot 10^{-3}}$	$\mathbf{1.01 \cdot 10^{-4}}$	$1.38 \cdot 10^{-3}$	$9.76 \cdot 10^{-3}$
Protein	$9.20 \cdot 10^{-3}$	$7.87 \cdot 10^{-3}$	$9.83 \cdot 10^{-3}$	$\mathbf{7.08 \cdot 10^{-5}}$	$8.08 \cdot 10^{-3}$	$\mathbf{9.81 \cdot 10^{-3}}$	$8.60 \cdot 10^{-5}$	$2.18 \cdot 10^{-3}$	$1.00 \cdot 10^{-2}$
CCPP	$1.64 \cdot 10^{-2}$	$2.42 \cdot 10^{-3}$	$9.99 \cdot 10^{-3}$	$2.91 \cdot 10^{-4}$	$2.24 \cdot 10^{-3}$	$\mathbf{9.52 \cdot 10^{-3}}$	$2.57 \cdot 10^{-4}$	$\mathbf{1.59 \cdot 10^{-3}}$	$1.02 \cdot 10^{-2}$

Table 1: Conditional generative models in regression. Best results are in **bold**.

4.5 Conditional Generative Model Construction

With the evaluation metrics, we present two corresponding deep-learning-based methods to construct conditional generative models. Our approaches are named as J-CGM and A-CGM, with special targets on the JMMD/AMMD for different tasks. For performance measurement, the values of JMMD and AMMD are estimated by drawing samples from the generative model optimized in J-CGM/A-CGM. Denote $G_\theta(\xi, X)$ as the generative model optimized in J-CGM/A-CGM with parameters θ. Note that $G_\theta(\xi, X)$ takes both the given conditional variables X and the extra random vector ξ as inputs. Let Q_θ be the joint distribution of $(X, G_\theta(\xi, X))$. A detailed step-by-step pseudo-code of A-CGM is listed in Algorithm 1. A similar procedure for J-CGM is presented in Algorithm 2 in Section C.

5 Experiments

Experimental Setup. We empirically verify the effectiveness of our proposed approaches on both regression and image generation tasks. In both tasks, we compare the performance of our approaches with the state-of-the-art MMD-based conditional generative model, CGMMN [47]. For regression, our experiments are conducted on the following widely used real-world benchmark datasets: Boston, Concrete, Energy, Wine, Yacht, Kin8nm, Protein, and CCPP [18, 10, 31, 44]. Besides the JMMD and AMMD values, we also report the scores of FID [19, 36], which is a standard metric for assessing the quality of generative models. In our experiments, FID is computed based on the joint distribution of (X, Y), since it is originally defined in the unconditional sense. In the label-based image generation task, we adopt the benchmark dataset MNIST [32] to evaluate the correctness of our framework. In this task, X is the label of the image and the generative model $G_\theta(\xi, X)$ is expected to output random image samples with the attribute of class X. We provide visuals to directly show the generation performance of different approaches. All experiments are conducted on a GeForce RTX 2080 Ti GPU. More experimental results are presented in Section C.

5.1 Aleatoric Uncertainty in Regression

Kernel Selections. As the regression data are low-dimensional, we choose k_1 and k_2 to be the standard Gaussian kernels $k_1(x_1, x_2) := \exp\left(-\frac{1}{2}\|x_1 - x_2\|_2^2\right)$, $k_2(y_1, y_2) := \exp\left(-\frac{1}{2}\|y_1 - y_2\|_2^2\right)$. Note that they readily satisfy both Assumptions 1 and 2.

Implementation Details. For regression tasks, we apply a simple network architecture with 2 hidden layers to avoid overfitting. We use the ReLU function as the activation function and the number of neurons in each hidden layer is 32. The input of the generative model is concatenated by two vectors, the covariate vector X and the extra random vector ξ following a 10-dimensional uniform distribution Uniform$([-1, 1]^{10})$. Our network is optimized by the Adam optimizer [25] with learning rate 0.0005. For the preprocessing step, we follow the same experimental procedure in [44] on data normalization and dataset splitting. The AMMD evaluation omits the free-of-model constant C_0 as justified in Theorem 5.

We evaluate the performance of our proposed J-CGM and A-CGM with the state-of-the-art baseline CGMMN [47] on multiple real-world benchmark regression datasets. Table 1 reports the evaluation metrics from different models on the testing data. As shown, J-CGM and A-CGM achieve competitive performance under the AMMD and JMMD evaluation criteria, while A-CGM tends to produce better results on AMMD. In contrast, although CGMMN can produce satisfactory results on AMMD, it underperforms on JMMD in general. Under the FID criterion, J-CGM and A-CGM achieve slightly
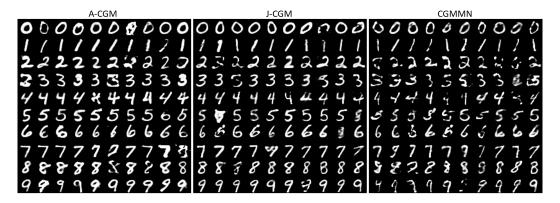
Figure 1: Random conditional samples generated by different approaches.

better results than CGMMN. However, note that FID is a heuristic criterion without the statistical properties we develop for AMMD and JMMD and is thus less reliable.

5.2 Aleatoric Uncertainty in Image Generation

Kernel Selections. As pointed out by previous studies on deep kernels [34, 33, 53, 35, 11], for complicated and high-dimensional real-world data, a kernel test using a simple kernel such as Gaussian kernel should be conducted on the code/feature space instead of the original data space to provide stronger signals for discrepancy measurement of high-dimensional distributions. Following this guidance, we apply an auto-encoder [48] to learn the representative features of the input images in the preprocessing step. Precisely, suppose the pre-trained auto-encoder network is given by $B_{\omega'} \circ A_{\omega}$ where $A_{\omega} : \mathcal{Y} \to \hat{\mathcal{Y}}$ is the encoder network with $\hat{\mathcal{Y}}$ being the lower-dimensional code space; $B_{\omega'} : \hat{\mathcal{Y}} \to \mathcal{Y}$ is the decoder network. We use the following feature-aware deep kernel for the image space $\mathcal{Y}$: $k_2(y_1, y_2) = \Big((1 - \epsilon_0)\kappa_1(A_{\omega}(y_1), A_{\omega}(y_2)) + \epsilon_0\Big)\kappa_2(y_1, y_2)$, where κ_1 is a Gaussian kernel defined on the code space $\hat{\mathcal{Y}}$; κ_2 is a Gaussian kernel defined on the original image space $\mathcal{Y}$; $\epsilon_0 \in (0, 1)$ is introduced to ensure that $k_2(y_1, y_2)$ is a characteristic kernel [35, 11]. We set k_1 to be the standard Gaussian kernels since $\mathcal{X}$ is low-dimensional.

Corresponding to our kernel, we now assume that all conditional generative models output samples in the code space for the convenience of MMD tests: $G_{\theta}(\xi, X) : \mathbb{R}^m \times \mathcal{X} \to \hat{\mathcal{Y}}$. The generative image is then given by $B_{\omega'} \circ G_{\theta}(\xi, X)$.

Implementation Details. In the auto-encoder network, the encoder/decoder networks A_{ω} and $B_{\omega'}$ have a single hidden layer with 1024 neurons. The dimension of the code space $\hat{\mathcal{Y}}$ is 32. The generative network is formed by 3 hidden layers with ReLU function as the activation function. The number of neurons in each hidden layer is 64, 256, and 256. The networks also take two vectors as input, the one-hot encoding vector of label X and the extra random vector ξ following a 10-dimensional uniform distribution Uniform($[-1, 1]^{10}$). The generative network is optimized by the Adam optimizer [25] with learning rate 0.001.

In Figure 1, we show a few random conditional samples of the reconstructed images from A-CGM, J-CGM, and CGMMN. Overall, all models can generate clear and recognizable samples of handwritten digits. In particular, the reconstructed images from J-CGM are more diverse with multiple writing types, while those from A-CGM are more clearly distinct. These results evidently demonstrate the effectiveness of our approaches on multiple real-world applications.

6 Conclusions

In this paper, we study the feasibility of leveraging conditional generative model on aleatoric uncertainty estimation. With theoretical justification, we propose two metrics for discrepancy measurement between two conditional distributions and demonstrate that both metrics can be easily and unbiasedly computed via Monte Carlo simulation. Experimental evaluations on multiple tasks corroborate our theory and further demonstrate the effectiveness of our approaches on real-world

applications. Our study explores a new direction on aleatoric uncertainty estimation, which overcomes a few limitations in the previous research. In the future, we will extend our approaches for aleatoric uncertainty estimation on more real-world applications such as super-resolution image generation.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223. PMLR, 2017.

[2] C. M. Bishop. Mixture density networks. 1994.

[3] O. Breuleux, Y. Bengio, and P. Vincent. Quickly generating representative samples from an rbm-derived process. *Neural Computation*, 23(8):2058–2073, 2011.

[4] G. C. Cawley, N. L. Talbot, and O. Chapelle. Estimating predictive variances with kernel ridge regression. In *Machine Learning Challenges Workshop*, pages 56–77. Springer, 2005.

[5] P. Cui, W. Hu, and J. Zhu. Calibrated reliable regression using maximum mean discrepancy. *Advances in Neural Information Processing Systems*, 33:17164–17175, 2020.

[6] N. Dalmasso, T. Pospisil, A. B. Lee, R. Izbicki, P. E. Freeman, and A. I. Malz. Conditional density estimation tools in python and r with applications to photometric redshifts and likelihood-free cosmological inference. *Astronomy and Computing*, 30:100362, 2020.

[7] V. Dutordoir, H. Salimbeni, J. Hensman, and M. Deisenroth. Gaussian process conditional density estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

[8] M. Fasiolo, S. N. Wood, M. Zaffran, R. Nedellec, and Y. Goude. Fast calibrated additive quantile regression. *Journal of the American Statistical Association*, 116(535):1402–1412, 2021.

[9] K. Fukumizu, L. Song, and A. Gretton. Kernel bayes' rule: Bayesian inference with positive definite kernels. *The Journal of Machine Learning Research*, 14(1):3753–3783, 2013.

[10] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016.

[11] R. Gao, F. Liu, J. Zhang, B. Han, T. Liu, G. Niu, and M. Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *International Conference on Machine Learning*, pages 3564–3575. PMLR, 2021.

[12] T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2):243–268, 2007.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[14] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[15] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. *Advances in Neural Information Processing Systems*, 30, 2017.

[16] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2017.

[17] D. Hendrycks and K. Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

[18] J. M. Hernández-Lobato and R. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869, 2015.

[19] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] M. P. Holmes, A. G. Gray, and C. L. Isbell. Fast nonparametric conditional density estimation. *arXiv preprint arXiv:1206.5278*, 2012.

[21] E. Hüllermeier and W. Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

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

[23] O. Kallenberg and O. Kallenberg. *Foundations of modern probability*, volume 2. Springer, 1997.

[24] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] V. Kuleshov, N. Fenner, and S. Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International conference on machine learning*, pages 2796–2804. PMLR, 2018.

[27] V. Kuleshov and P. S. Liang. Calibrated structured prediction. *Advances in Neural Information Processing Systems*, 28, 2015.

[28] M. Kull and P. Flach. Novel decompositions of proper scoring rules for classification: Score adjustment as precursor to calibration. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 68–85. Springer, 2015.

[29] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in Neural Information Processing Systems*, 32, 2019.

[30] A. Kumar, P. S. Liang, and T. Ma. Verified uncertainty calibration. *Advances in Neural Information Processing Systems*, 32, 2019.

[31] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[32] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[33] C.-L. Li, W.-C. Chang, Y. Cheng, Y. Yang, and B. Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in Neural Information Processing Systems*, 30, 2017.

[34] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727. PMLR, 2015.

[35] F. Liu, W. Xu, J. Lu, G. Zhang, A. Gretton, and D. J. Sutherland. Learning deep kernels for non-parametric two-sample tests. In *International Conference on Machine Learning*, pages 6316–6326. PMLR, 2020.

[36] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. *Advances in neural information processing systems*, 31, 2018.

[37] M. Minderer, J. Djolonga, R. Romijnders, F. Hubis, X. Zhai, N. Houlsby, D. Tran, and M. Lucic. Revisiting the calibration of modern neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[38] M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[39] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[40] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 625–632, 2005.

[41] D. A. Nix and A. S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 ieee international conference on neural networks (ICNN'94)*, volume 1, pages 55–60. IEEE, 1994.

[42] S. Nowozin, B. Cseke, and R. Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in Neural Information Processing Systems*, 29, 2016.

[43] J. Park and K. Muandet. A measure-theoretic approach to kernel conditional mean embeddings. *Advances in Neural Information Processing Systems*, 33:21247–21259, 2020.

[44] T. Pearce, M. Zaki, A. Brintrup, and A. Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. *arXiv preprint arXiv:1802.07167*, 2018.

[45] T. Pospisil and A. B. Lee. Rfcde: Random forests for conditional density estimation. *arXiv preprint arXiv:1804.05753*, 2018.

[46] T. Pospisil and A. B. Lee. (f) rfcde: Random forests for conditional density estimation and functional data. *arXiv preprint arXiv:1906.07177*, 2019.

[47] Y. Ren, J. Zhu, J. Li, and Y. Luo. Conditional generative moment-matching networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[48] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[49] H. Song, T. Diethe, M. Kull, and P. Flach. Distribution calibration for regression. In *International Conference on Machine Learning*, pages 5897–5906. PMLR, 2019.

[50] L. Song, K. Fukumizu, and A. Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

[51] L. Song, J. Huang, A. Smola, and K. Fukumizu. Hilbert space embeddings of conditional distributions with applications to dynamical systems. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 961–968, 2009.

[52] M. Sugiyama, I. Takeuchi, T. Suzuki, T. Kanamori, H. Hachiya, and D. Okanohara. Conditional density estimation via least-squares density ratio estimation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 781–788. JMLR Workshop and Conference Proceedings, 2010.

[53] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing. Deep kernel learning. In *Artificial Intelligence and Statistics*, pages 370–378. PMLR, 2016.