Estimating Epistemic and Aleatoric Uncertainty with a Single Model

Matthew A. Chan
Department of Computer Science
University of Maryland
College Park, MD 20742
mattchan@umd.edu

Maria J. Molina
Department of Atmospheric and Oceanic Science
University of Maryland
College Park, MD 20742
mjmolina@umd.edu

Christopher A. Metzler
Department of Computer Science
University of Maryland
College Park, MD 20742
metzler@umd.edu

Abstract

Estimating and disentangling epistemic uncertainty, *uncertainty that is reducible with more training data*, and aleatoric uncertainty, *uncertainty that is inherent to the task at hand*, is critically important when applying machine learning to high-stakes applications such as medical imaging and weather forecasting. Conditional diffusion models' breakthrough ability to accurately and efficiently sample from the posterior distribution of a dataset now makes uncertainty estimation conceptually straightforward: One need only train and sample from a large ensemble of diffusion models. Unfortunately, training such an ensemble becomes computationally intractable as the complexity of the model architecture grows. In this work we introduce a new approach to ensembling, *hyper-diffusion models (HyperDM)*, which allows one to accurately estimate both epistemic and aleatoric uncertainty with a single model. Unlike existing single-model uncertainty methods like Monte-Carlo dropout and Bayesian neural networks, HyperDM offers prediction accuracy on par with, and in some cases superior to, multi-model ensembles. Furthermore, our proposed approach scales to modern network architectures such as Attention U-Net and yields more accurate uncertainty estimates compared to existing methods. We validate our method on two distinct real-world tasks: x-ray computed tomography reconstruction and weather temperature forecasting. Source code is publicly available at `https://github.com/matthewachan/hyperdm`.

1 Introduction

Machine learning (ML) based inference and prediction algorithms are being actively adopted in a range of high-stakes scientific and medical applications: ML is already deployed within modern computed tomography (CT) scanners [10], ML is actively used to search for new medicines [29], and over the last year ML has begun to compete with state-of-the-art weather and climate forecasting systems [46, 36, 7]. In mission-critical tasks like weather forecasting and medical imaging/diagnosis, the importance of reliable predictions cannot be overstated. The consequences of erroneous decisions in these domains can range from massive financial costs to, more critically, the loss of human lives. In this context, understanding and quantifying uncertainty is a pivotal step towards improving the robustness and reliability of ML models.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

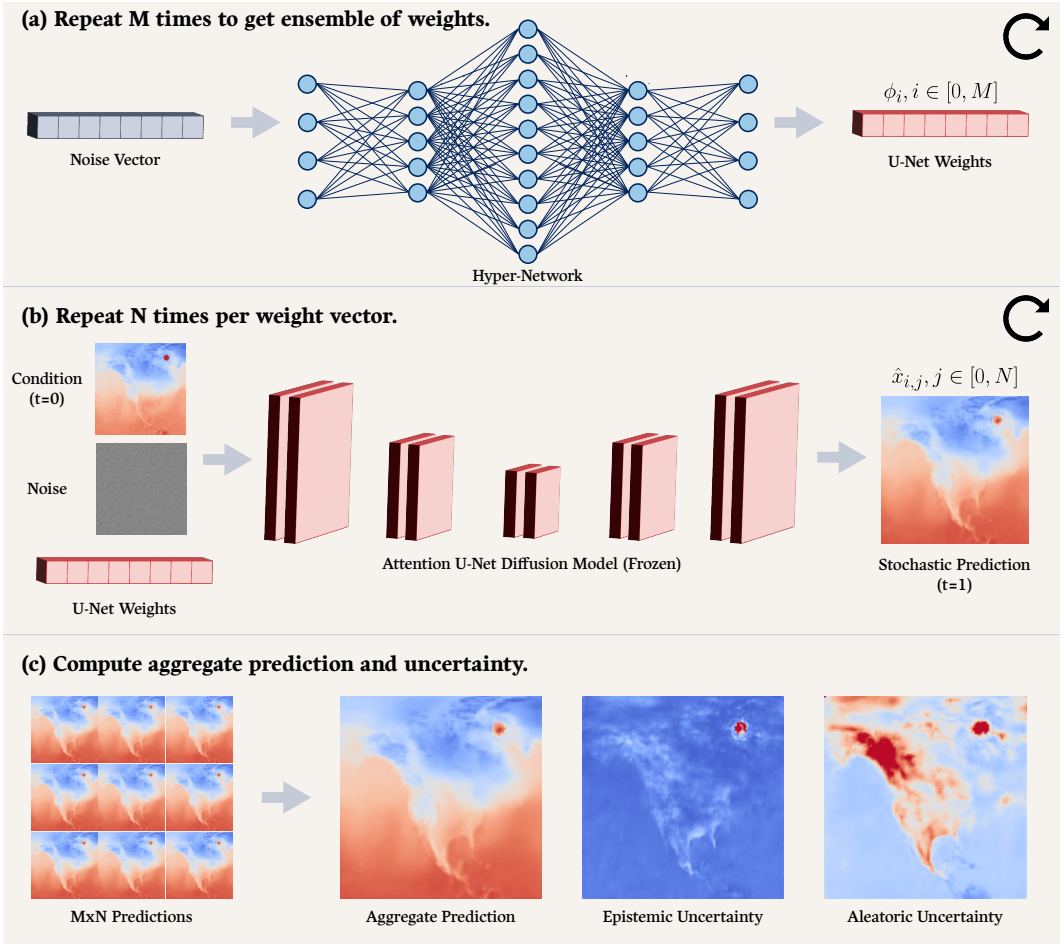

Figure 1: **General framework of HyperDM.** (a) A Bayesian hyper-network is optimized to generate diffusion model weights from randomly sampled noise. This process is repeated M times to obtain an ensemble of M weights. (b) A diffusion model accepts fixed weights from the hyper-network to stochastically generate a prediction. This process is repeated N times for each set of weights, yielding a total of $M \times N$ predictions. (c) The ensemble predictions are aggregated to produce a final prediction and an epistemic / aleatoric uncertainty map.

For an uncertainty estimate to be most useful, it must differentiate between *aleatoric* and *epistemic* uncertainty. Aleatoric uncertainty describes the fundamental variability and ill-posedness of the inference task. By contrast, epistemic uncertainty describes the inference model's lack of knowledge or understanding—which can be reduced with more diverse training data. Distinguishing between these two types of uncertainty provides valuable insights into the strengths and weaknesses of a predictive model, offering pathways towards improving its performance. In applications like weather forecasting, epistemic uncertainty can be used to inform the optimal placement of new weather stations. Additionally, in medical imaging, decomposition of uncertainty into its aleatoric and epistemic components is important for identifying out-of-distribution measurements where model predictions should be verified by trained experts.

This work presents a new approach for estimating aleatoric and epistemic uncertainty *using a single model*. Specifically, our approach uses a novel pipeline integrating a conditional diffusion model [23] and a Bayesian hyper-network [34] to generate an ensemble of predictions. Conditional diffusion models allow one to sample from an implicit representation of the posterior distribution of an inverse problem. Meanwhile, hyper-networks allow one to sample over a collection of networks that are consistent with the training data. Together, these components can efficiently estimate both sources of uncertainty, without sacrificing inference accuracy. Our specific contributions are summarized below:

Table 1: **Comparison of training and inference times.** The time required to train an $M = 10$ member ensemble on the LUNA16 dataset is shown in the second column. The third column shows the time required to generate a predictive distribution of size $M \times N = 1000$ for a single input.

METHOD	TRAINING TIME (MINUTES)	EVALUATION TIME (MINUTES)
MC-DROPOUT [18]	47.03	3.70
DPS-UQ [14]	441.09	3.31
HYPERDM	48.53	3.18

- We apply Bayesian hyper-networks in a novel setting (i.e., diffusion models) to estimate both epistemic and aleatoric uncertainties from a single model.
- We conduct a toy experiment with ground truth uncertainties and show that the proposed method accurately predicts both sources of uncertainty.
- We apply the proposed method on two mission-critical real-world tasks, CT reconstruction and weather forecasting, and demonstrate that our method achieves a significantly lower training overhead and better reconstruction quality compared to existing methods.
- We conduct ablation studies investigating the effects of ensemble size and the number of ensemble predictions on uncertainty quality, which show (i) that larger ensembles improve out-of-distribution detection and (ii) that additional predictions smooth out irregularities in aleatoric uncertainty estimates.

2 Related Work

2.1 Uncertainty Quantification

Probabilistic methods are commonly used to estimate uncertainty by first generating an ensemble of models and subsequently quantifying uncertainty as the variance or entropy over the ensemble's predictions [11]. Deep ensembles [35] explicitly train such an ensemble to predict epistemic uncertainty. However, with modern neural network architectures exceeding a billion parameters, the computational cost required to train deep ensembles is prohibitively expensive.

Other methods attempt to approximate deep ensembles while circumventing its training overhead. Bayesian neural networks (BNNs) [40, 44] use variational inference [19, 65] to model the posterior weight distribution. Monte-Carlo (MC) dropout [18] leverages dropout [57] to stochastically induce variability in the network's predictions. Recent works in weather forecasting [7, 46] perturb network inputs with random noise to similarly generate stochastic predictions. Still, each of these methods has notable trade-offs preventing their widespread adoption. BNNs incur a runtime cost that scales proportionally with the number of model parameters, leading to slow inference and training times. MC dropout and input perturbation introduce noise into the inference process, which adversely affects model prediction quality. Moreover, perturbing inputs with noise is not equivalent to deep ensembles (in the Bayesian sense) as these methods optimize the weights of a single, deterministic model.

A separate branch of research [48, 3] explores distribution-free uncertainty estimation, which uses conformal prediction [4] and quantile regression [32] to estimate bounds on aleatoric uncertainty. Subsequent works [58] have extended this to additionally estimate epistemic uncertainty; however, these methods use deep ensembles to do so—leading to similar issues with computational complexity.

2.2 Hyper-Networks

Hyper-networks [20] employ a unique paradigm where one network—the "hyper-network"—generates weights for another "primary" network. This framework circumvents the need to train multiple task-specific or dataset-specific models. Instead, one need only train a single hyper-network to cover a range of tasks or datasets. Given an input token representing a specific task, a hyper-network learns to generate reasonable weights with which the primary network can accomplish that task [61]. Note that, during training, losses are back-propagated such that only the hyper-network's parameters are updated while the primary network's weights are purely generated by the hyper-network.

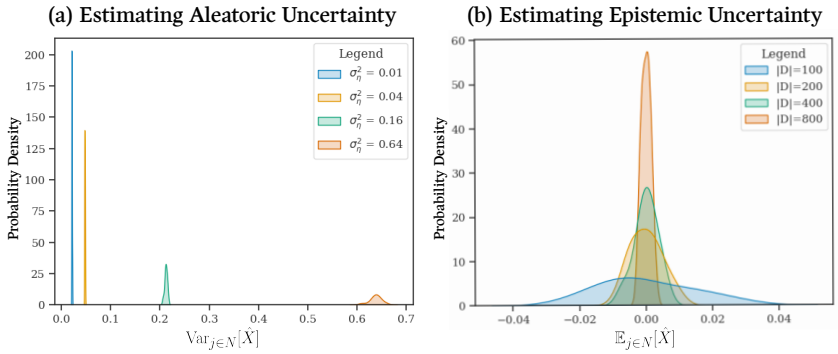

Figure 2: **Accurate uncertainty estimation using HyperDM.** (a) HyperDM is trained on four 1D datasets with aleatoric uncertainty determined by noise variance σ_η^2. Variances across diffusion model predictions are visualized as one distribution per training dataset. Aleatoric estimates (i.e., the mean of each distribution) accurately predict σ_η^2. (b) HyperDM is trained on four datasets with epistemic uncertainty determined by dataset size $|\mathcal{D}|$. Prediction means are visualized as one distribution per training dataset. Epistemic estimates (i.e., the variance of each distribution) grow inversely with $|\mathcal{D}|$.

Bayesian hyper-networks (BHNs) [34, 33] extend hyper-networks to quantify uncertainty. Rather than accepting task-specific tokens as inputs, BHNs accept random noise and stochastically generate weights for the primary network. BHNs thus serve as an implicit representation for the true posterior weight distribution [47, 28]. Epistemic uncertainty is measured as the variance across predictions yielded by the primary network for different weights sampled from the BHN.

2.3 Diffusion Models

Diffusion models (DMs) [52, 55, 56] represent a class of generative machine learning models that learns to sample from a target distribution. These models fit to the Stein score function [39] of the target distribution by iteratively transitioning between an easy-to-sample (typically Gaussian) distribution and the target distribution. During training, samples from the target distribution are corrupted by running the forward "noising" diffusion process, and the network learns to estimate the added noise. To generate samples, the network iteratively denoises images of pure noise until they looks like they were sampled from the target distribution [23]. DMs have shown success in generating high-quality, realistic images and capturing diverse data distributions [12]. To date, however, there has been limited research [6] investigating the use of DMs for uncertainty estimation.

3 Problem Definition

Given measurements $y \sim \mathcal{Y}$ corresponding to signals of interest $x \sim \mathcal{X}$, our objective is to train a model which can simultaneously recover x and quantify the aleatoric and epistemic uncertainty of its predictions. The predictive distribution of a such a model is given by

$$p(x|y, \mathcal{D}) = \int p(x|y, \phi)p(\phi|\mathcal{D})d\phi \tag{1}$$

where $p(x|y, \phi)$ is the likelihood function, and $p(\phi|\mathcal{D})$ is the posterior over model parameters ϕ for a training dataset $\mathcal{D}$ [59]. Uncertainty on this distribution stems from two distinct sources: aleatoric uncertainty and epistemic uncertainty [13].

3.1 Aleatoric Uncertainty

Aleatoric uncertainty (AU) arises from inherent randomness in the underlying measurement process and is represented by the likelihood function in Equation (1). Most notably, this source of uncertainty is irreducible for a given measurement process [17, 30]. In the context of predictive modeling, AU represents how ill-posed the task is and is often associated with noise, measurement errors, or inherent unpredictability in the observed phenomena.

Consider an inverse problem where the goal is to recover x from measurements

$$y = \mathcal{F}(x) + \eta, \tag{2}$$

defined by forward operator $\mathcal{F}$ and non-zero measurement noise $\eta \sim \mathcal{N}(0, \sigma^2)$, by learning the inverse mapping $\mathcal{F}^{-1} : \mathcal{Y} \to \mathcal{X}$. Even with a perfect model capable of sampling from the true likelihood $p(x|y)$, irreducible errors are still present due to the ambiguity η around which $x \sim p(x)$ explains the observed measurement y. This ambiguity captures the inherent randomness (i.e., the *aleatoric uncertainty*) of the inverse problem and is measured by the variance σ^2 of η [27].

3.2 Epistemic Uncertainty

Epistemic uncertainty (EU) relates to a lack of knowledge or incomplete understanding of a problem and is reducible with additional training data [27]. This type of uncertainty reflects limitations in a model's knowledge and its ability to accurately capture underlying patterns in the data. Assume we initialize M models with random weights $\{\phi_i\}_{i=0}^{M}$ and sufficient capacity to perfectly capture the inverse model described in Section 3.1. After training, discrepancies (i.e., *epistemic uncertainty*) inevitably arise in the final weights learned by each model, due to the random weight initialization process. As additional training data is provided, model weights converge more strongly—corresponding to a reduction in EU [11, 27].

4 Method

We measure uncertainty using variance and apply the law of total variance [11, 60, 50] to decompose total uncertainty (TU) across model predictions $\hat{X} \sim p(x|y, \mathcal{D})$ into its AU and EU components

$$\text{Var}(\hat{X}) = \underbrace{\text{Var}_{\phi \sim p(\phi|\mathcal{D})}\left[\mathbb{E}_{\hat{x} \sim p(x|y,\phi)}\left[\hat{X}\right]\right]}_{\text{EU}} + \underbrace{\mathbb{E}_{\phi \sim p(\phi|\mathcal{D})}\left[\text{Var}_{\hat{x} \sim p(x|y,\phi)}\left[\hat{X}\right]\right]}_{\text{AU}}. \tag{3}$$

The first term captures the explainable uncertainty, given by the variance of sampled weights $\phi \sim p(\phi|\mathcal{D})$ over the expected values of samples $\hat{x} \sim p(x|y, \phi)$ from the likelihood function. This term ignores variance caused by the ill-posedness of the likelihood function and therefore represents EU. The second term captures the unexplainable uncertainty and is given by the expectation of sampled weights $\phi \sim p(\phi|\mathcal{D})$ over the variance of samples $\hat{x} \sim p(x|y, \phi)$ from the likelihood function. This term ignores variance caused by the sampling of weights from the posterior and therefore represents AU.

Both the likelihood function $p(\phi|\mathcal{D})$ and the posterior $p(x|y, \phi)$ do not have an explicit closed-form, making computation of (3) intractable. To circumvent this, we instead learn their respective implicit distributions [28, 47] $q(\phi)$ and $q(x|y)$.

4.1 Implicit Likelihood Function

As demonstrated in [55], DMs enable sampling from an implicit conditional distribution $q(x|y)$ by learning to invert a diffusion process that gradually transforms a target data distribution into a simple (typically Gaussian) data distribution [56]. The forward diffusion process can be described by a T length Markov chain

$$q(x^{(t)}|x^{(t-1)}) := \mathcal{N}\left(x^{(t)}; \sqrt{1 - \sigma_t^2}x^{(t-1)}, \sigma_t^2\right) \tag{4}$$

that transforms samples $x^{(0)}$ from the data distribution into samples $x^{(T)}$ from a Gaussian distribution [23]. Conversely, the reverse diffusion process

$$p(x^{(t-1)}|x^{(t)}, y) := \mathcal{N}(x^{(t-1)}; x^{(t)} + \sigma_t^2 \nabla_{x^{(t)}} \log p(x^{(t)}|y), \sigma_t^2) \tag{5}$$

transforms pure noise into samples from $p(x|y)$ [63].

Since explicit computation of the score function $\nabla_{x_t} \log p(x^{(t)}|y)$ is intractable [16], a neural network $s(x, t|y, \phi)$ is typically trained to approximate it via an L2 minimization objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\left\|\nabla_{x^{(t)}} \log p(x^{(t)}|y) - s(x^{(t)}, t|y, \phi)\right\|_2^2\right] \tag{6}$$

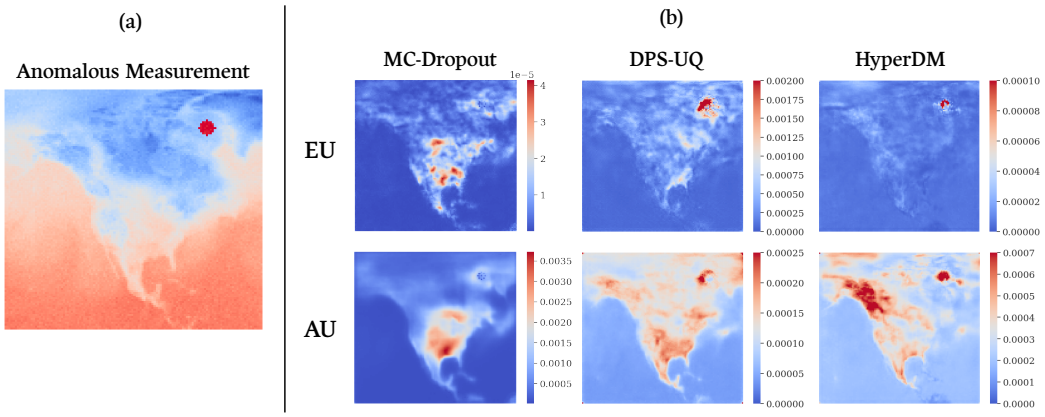

Figure 3: **Weather forecasting on out-of-distribution data.** (a) An out-of-distribution measurement is formed by synthetically inserting a hot spot in the northeastern part of Canada. (b) Epistemic and aleatoric uncertainty maps are produced by each method on the provided measurement. Compared to other methods, HyperDM is best able to isolate the abnormal feature in its epistemic estimate.

where $\mathcal{D} = \{(x_0, y_0), \ldots, (x_N, y_N)\}$ represents the training dataset. Once the DM has finished training, we can sample from the implicit likelihood function $q(x|y)$ by first sampling random noise $x^{(T)} \sim \mathcal{N}(0, \sigma^2)$ and iteratively denoising the image T times following Equation (5) to obtain $x^{(0)} \sim q(x|y)$ [42].

4.2 Implicit Posterior Distribution

Similar to the likelihood function, the posterior weight distribution has no explicit closed-form representation, so we instead make use of an implicit distribution $q(\phi)$ to approximate $p(\phi|\mathcal{D})$. As mentioned in Section 2.2, BHNs [34] enable sampling from $q(\phi)$ by transforming samples $z \sim \mathcal{N}(0, \sigma^2)$ into weights $\phi \sim q(\phi)$ for the primary network. In the case of the inverse imaging problem from Section 3.1, the primary network would be a network $f(\cdot|\phi)$ with parameters ϕ that learns the inverse mapping from measurements to signals $\mathcal{Y} \to \mathcal{X}$.

Training a BHN differs from conventional deep learning methods in that the weights of the primary network ϕ are generated by a hyper-network and are thus not learnable parameters. Instead, the weights θ of a BHN h_θ are optimized via the minimization objective

$$\mathbb{E}_{(x,y)\sim\mathcal{D}, z\sim\mathcal{N}(0,\sigma^2)} \left[\| f(y \mid h_\theta(z)) - x \|_2^2 \right] \tag{7}$$

where h_θ maps random input vectors $z \sim \mathcal{N}(0, \sigma^2)$ to weights ϕ (see Figure 1a). Importantly, weights produced by the BHN do not collapse to a mode because there are many network weights which yield plausible predictions with respect to L2 distance. As less data is available during training, a broader range of network weights reasonably explain that data.

4.3 Estimation of Aleatoric and Epistemic Uncertainty with a Single Model

We leverage DMs and BHNs to implicitly model $p(x|y, \phi)$ and $p(\phi|\mathcal{D})$, respectively, thus enabling sampling from both distributions. Specifically, our framework consists of a BHN h_θ that generates weights $\phi_i \sim q(\phi)$ for a DM $s(\cdot|\phi)$, which we collectively refer to as a hyper-diffusion model (HyperDM). At inference time, we sample $i \in M$ weights from h_θ and for each weight ϕ_i generate $j \in N$ samples from $s(\cdot|\phi_i)$—yielding a distribution of $M \times N$ predictions $\hat{x}_{i,j}$ (see Figure 1). This framework is a transformation of Equation (1), where both posterior and likelihood have been replaced with implicit distributions to yield a tractable approximation of the predictive distribution

$$p(x|y, \mathcal{D}) \approx \int q(x|y) q(\phi) d\phi. \tag{8}$$

Table 2: **Ensemble prediction quality on real-world data.** Baseline image quality assessment scores are calculated on test data from a CT dataset (i.e., LUNA16) and a weather forecasting dataset (i.e., ERA5). Best scores are highlighted in red and second best scores are highlighted in blue.

	LUNA16			ERA5		
METHOD	SSIM ↑	PSNR (DB) ↑	CRPS ↓	SSIM ↑	PSNR (DB) ↑	CRPS ↓
MC-DROPOUT [18]	0.77	30.25	0.023	0.93	31.34	0.034
DPS-UQ [14]	0.89	34.95	0.01	0.94	32.83	0.013
HYPERDM	0.87	35.16	0.01	0.95	33.15	0.012

Applying Equation (3), uncertainty over the predictive distribution $\hat{X} = \{\hat{x}_{i,j}, \ldots, \hat{x}_{M,N}\}$ is decomposed into its respective aleatoric and epistemic components,

$$\widehat{\text{AU}} = \mathbb{E}_{i \in M}\left[\text{Var}_{j \in N}\left[\hat{X}\right]\right] \tag{9}$$

$$\widehat{\text{EU}} = \text{Var}_{i \in M}\left[\mathbb{E}_{j \in N}\left[\hat{X}\right]\right], \tag{10}$$

such that $\widehat{\text{TU}} = \widehat{\text{AU}} + \widehat{\text{EU}}$. Following existing ensemble methods [35, 46, 7], we compute the aggregate ensemble prediction as the expectation over $\hat{X}$, formally expressed as

$$\mathbb{E}_{i \in M, j \in N}\left[\hat{X}\right]. \tag{11}$$

Compared to other aggregation methods (e.g., median, mode), we observe the best performance when taking the ensemble mean. Please refer to Figure 5 and Table 3 in the supplement for more details.

Unlike deep ensembles which require training M distinct models to compute EU and AU, HyperDM only requires training a single model (i.e., a BHN)—theoretically consuming up to M-fold fewer computational resources. Furthermore, unlike many pseudo-ensembling methods [18, 7, 46], HyperDM doesn't need to exploit randomness caused by perturbations to model $p(\phi|\mathcal{D})$—avoiding adverse effects on model performance. Moreover, unlike BNNs, HyperDM is relatively cheap to sample from in terms of computational runtime and resources—making it significantly faster and more scalable compared to BNN-based uncertainty estimation methods.

Differing from prior work [60], we make no Gaussian assumptions on the predictive distribution $p(x|y, \mathcal{D})$ nor on the likelihood function $p(x|y, \phi)$. This is because our method approximates $p(x|y, \phi)$ by repeatedly sampling $\hat{x} \sim q(x|y)$ from a DM, rather than explicitly modeling the distribution as a Gaussian with mean μ and variance σ^2. Therefore, our aggregate predictive distribution $p(x|y, \mathcal{D})$ is not restricted to a Gaussian mixture model $\mathcal{N}(\mu_*(x), \sigma_*^2(x))$ over the collective mean μ_* and variance σ_*^2 of all ensemble members.

5 Experiments

Please refer to Appendix A for training details (e.g., network architectures and loss functions).

Baselines. We focus on comparing HyperDM against methods which are similarly capable of estimating both EU and AU. Our benchmark consists of a state-of-the-art method, deep-posterior sampling for uncertainty quantification [14] (henceforth referred to as DPS-UQ), and a dropout-based method (referred to as MC-Dropout). DPS-UQ is implemented as an M-member ensemble of deep-posterior sampling (DPS) DMs. MC-Dropout is implemented as a single DM with weights sampled from $q(\phi)$ using dropout instead of a BHN. Despite its inability to jointly predict EU and AU, we also include a BNN baseline in our initial experiments to illustrate the advantages of our method in terms of prediction speed and accuracy.

Metrics. We evaluate the quality of baseline predictions using both full-reference image quality metrics and distribution-based metrics. Specifically, we compute peak signal-to-noise ratio (PSNR) [24] and structural similarity index (SSIM) [62] between mean predictions (see Equation (11)) and their corresponding ground truth references. We also compute the continuous ranked probability score (CRPS) [41] as a holistic indicator of the quality of $\hat{X}$, given by

$$\text{CRPS}(F, a) = \int_{-\infty}^{\infty} \left[F(a) - \mathbb{1}_{a \geq x}\right]^2 da, \tag{12}$$

where F is the cumulative distribution function of $\hat{X}$ and $\mathbb{1}$ is the Heaviside step function.

In our initial experiments, we compare baseline estimates of AU and EU against ground truth uncertainty. However, extending such validation to more complex tasks and datasets is difficult because uncertainty is affected by a wide variety of environmental factors (e.g., measurement noise, sampling rates) which are often unreported. As a result, in subsequent experiments, we follow [14] and evaluate uncertainty by generating out-of-distribution (OOD) measurements and verifying whether baseline estimates of $\widehat{\text{EU}}$ correctly predict OOD pixels.

5.1 Toy Problem

We first evaluate our method on a toy inverse problem to establish the correctness of our uncertainty estimates under a simple forward model where ground truth uncertainty is explicitly quantifiable. Training datasets are generated using the function

$$x = \sin(y) + \eta \tag{13}$$

where $\eta \sim \mathcal{N}(0, \sigma_\eta^2)$ and measurements $y \sim \mathcal{U}(-5, 5)$. We conduct two separate experiments to validate our method's ability to estimate uncertainty against ground-truth EU and AU.

Estimating AU. To test our method's ability to estimate AU, we generate four training datasets using Equation (13) with ground truth AU characterized by noise variances $\sigma_\eta^2 \in \{0.01, 0.04, 0.16, 0.64\}$. Each training dataset has $|\mathcal{D}| = 500$ examples, and a HyperDM is trained on each dataset for 500 epochs. After training, we sample $M = 10$ weights from h_θ and $N = 10000$ realizations from $s(\cdot|\phi)$ to obtain a distribution of $M \times N$ predictions. We compute $\widehat{\text{AU}}$ for each ensemble member using Equation (9) and visualize $\text{Var}_{j \in N}[\hat{X}]$ across all M weights in Figure 2a. The AU estimates across the four datasets are $\widehat{\text{AU}} = \{0.02, 0.05, 0.21, 0.64\}$, which closely match the ground-truth.

Estimating EU. We test our method's ability to estimate EU by generating four training datasets of varying sizes $|\mathcal{D}| \in \{100, 200, 400, 800\}$ and fixed noise variance $\sigma_\eta^2 = 0.01$. Unlike AU, ground truth EU cannot be explicitly quantified because it is independent from the training data [5]. As a result, we follow prior works [35, 13, 38] and validate EU qualitatively. We train a HyperDM on each dataset for 500 epochs and draw $M \times N$ samples from it where $M = 10, N = 10000$. We then calculate $\widehat{\text{EU}}$ using Equation (10) and plot $\mathbb{E}_{j \in N}[\hat{X}]$ for all M weights in Figure 2b. The EU estimates across the four datasets are $\widehat{\text{EU}} = \left\{1.92 \times 10^{-4}, 2.20 \times 10^{-5}, 1.17 \times 10^{-5}, 1.83 \times 10^{-6}\right\}$, ordered by increasing $|\mathcal{D}|$. As expected, $\widehat{\text{EU}}$ decreases as $|\mathcal{D}|$ grows increasingly larger.

To highlight the key advantages of HyperDM over traditional uncertainty estimation techniques, we also train a BNN on the same datasets and sample $M = 10000$ predictions. The resulting estimates $\widehat{\text{EU}} = \{0.091, 0.071, 0.050, 0.012\}$ indicate a similar inversely proportional relationship with $|\mathcal{D}|$. However, despite having the same backbone architecture and training hyper-parameters, we observe aggregate predictions of lower individual and mean quality from the BNN when compared to HyperDM (see Table 4 of the supplement). Moreover, we observe that BNNs take over $2\times$ longer to train compared to HyperDM (i.e., 70 seconds vs. 30 seconds), and inference is an order of magnitude slower (i.e., 8.7 seconds vs. 0.7 seconds to generate 10,000 predictions), due to BNN's need to sample from the weight distribution at runtime.

5.2 Computed Tomography

In this experiment, we demonstrate our method's applicability for medical imaging tasks, specifically CT reconstruction. Using the Lung Nodule Analysis 2016 (LUNA16) [51] dataset, we form a target image distribution $\mathcal{X}$ by extracting 1,200 CT images, applying $4\times$ pixel binning to produce 128×128 resolution images, and normalizing each image by mapping pixel values between $[-1000, 3000]$ Hounsfield units to the interval $[-1, 1]$. We subsequently compute the sparse Radon transform with 45 projected views and add Gaussian noise with variance $\sigma^2 = 0.16$ to the resulting sinograms. Using filtered back-projection (FBP) [25], we obtain low-quality reconstructions $\mathcal{Y}$ of original images $\mathcal{X}$. The dataset is finally split into a training dataset comprised of 1,000 image-measurement pairs and a validation dataset of 200 data pairs. Following the training procedure described in Appendix A, we train MC-Dropout, DPS-UQ, and HyperDM on LUNA16. For fair comparison, all baselines are sampled from using $M = 10$ and $N = 100$ for a total of $M \times N = 1000$ predictions. We refrain

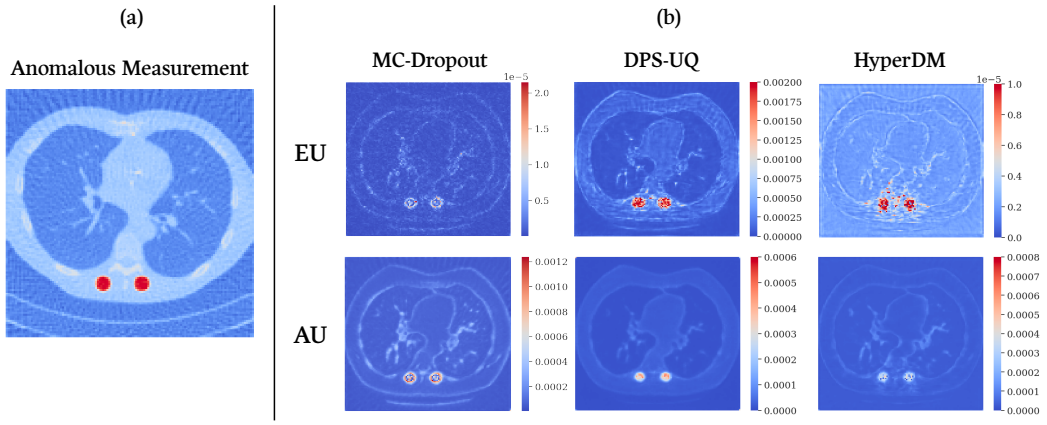

Figure 4: **CT reconstruction on out-of-distribution data.** (a) An out-of-distribution CT measurement formed by synthetically inserting metal implants along the spine. (b) Epistemic and aleatoric uncertainty maps are produced by each method on the out-of-distribution measurement. Both DPS-UQ and HyperDM are able to distinguish the abnormal feature in their epistemic prediction.

from training a BNN baseline on this dataset due to the high computational resources and runtime required to scale to the image domain.

In Table 2, we show average CRPS, PSNR, and SSIM scores computed over the test dataset. The relatively low image quality scores obtained by MC-Dropout are indicative of the adverse effects caused by randomly dropping network weights at inference time. Meanwhile, DPS-UQ reconstructions achieve a 15.5% higher average PSNR than MC-Dropout, but at the cost of an eight-fold increase in training time (see Table 1). On the other hand, HyperDM yields predictions of similar—and sometimes better—quality than DPS-UQ while only adding a 3% overhead in training time compared to MC-Dropout. Note that the discrepancy in training times between DPS-UQ and HyperDM will continually widen as we scale the ensemble size beyond $M = 10$. However, due to the high computational costs required to train $M > 10$ member deep ensembles, we limited baselines to ten-member ensembles for this experiment.

To evaluate the quality of baseline uncertainty predictions, we first select a random in-distribution image $x \in \mathcal{X}$ and generate its corresponding OOD measurement by first artificially inserting an abnormal feature (i.e., metal implants along the spinal column) and subsequently computing the corresponding FBP measurement y. Results in Figure 4 show that DPS-UQ and HyperDM yield comparable results in that their $\widehat{\text{EU}}$ predictions successfully highlight the OOD implant. In contrast, MC-Dropout fails to highlight OOD pixels in its $\widehat{\text{EU}}$ prediction. While prior work [43] suggests that AU estimates are unreliable—and should be subsequently disregarded—whenever EU is high, we nonetheless include $\widehat{\text{AU}}$ results in Figure 4 to demonstrate that HyperDM produces $\widehat{\text{AU}}$ predictions similar to that of a deep ensemble.

5.3 Weather Forecasting

In this experiment, we demonstrate the applicability of HyperDM for climate science—specifically two-meter surface temperature forecasting. Using the European Centre for Medium-Range Weather Forecasts Reanalysis v5 (ERA5) dataset [22], we generate a dataset comprised of 1,240 surface air temperature maps sampled at six-hour time intervals (i.e., $00, 06, 12, 18$ UTC) in January between 2009-2018. Images are binned down to 128×128 resolution and normalized such that pixel values between $[210, 313]$ Kelvin map to the interval $[-1, 1]$. Following experiments done in [46], we form data pairs (x, y) using historical temperature data at time t as the initial measurement image y and data at time $t + 6$ hours as the target image x. A total of 200 images are held-out and used for validation and testing purposes.

Using the same training procedure as Section 5.2, we train MC-Dropout, DPS-UQ, and HyperDM on ERA5 and generate predictions with sampling rates $M = 10$ and $N = 100$. Baseline PSNR, SSIM,

and CRPS scores are reported in Table 2, where we observe trends similar to the prior experiment: DPS-UQ achieves a 5% higher average PSNR score compared to MC-Dropout, and HyperDM achieves 1% higher average PSNR score compared to DPS-UQ. Training overhead for DPS-UQ remains around $8\times$ that of MC-Dropout and HyperDM due to the need to repeat training M times.

To generate OOD measurements, we first obtain an in-distribution measurement and subsequently insert an anomalous hot spot over northeastern Canada. Inspecting results in Figure 3, we observe that HyperDM's $\widehat{\text{EU}}$ prediction more accurately identifies OOD pixels than DPS-UQ. In contrast, MC-Dropout fails to identify the hot spot in its $\widehat{\text{EU}}$ prediction and instead incorrectly identifies regions in the central United States as OOD. Interestingly, all methods predict lower $\widehat{\text{AU}}$ over the ocean versus the North American continent, which aligns with our expectations, as water has less temperature variability compared to land due to its higher specific heat. Additional qualitative results showing the decomposition of $\widehat{\text{TU}}$ into its $\widehat{\text{EU}}$ and $\widehat{\text{AU}}$ components are provided in Figure 8 of the supplement.

6 Limitations and Future Work

We acknowledge two main limitations of our approach and identify potential avenues for improvement. Firstly, as a consequence of their iterative denoising process, inference on DMs is slow compared to inference on classical neural network architectures. However, recent advances in accelerated sampling strategies have largely mitigated this issue and allow for few [53] (and in some cases single [54]) step sampling from DMs. Secondly, hyper-networks suffer from a scalability problem in that their number of parameters scales with the number of primary network parameters. This stems from the fact that the dimensionality of the hyper-network's output layer is (in most cases) proportional to the number of parameters in the primary network [9]. Several works address this issue by proposing more efficient weight generation strategies [61, 2, 26]. Nonetheless, these problems remain a promising avenue for future research.

7 Conclusion

The growing application of ML to impactful scientific and medical problems has made accurate estimation of uncertainty more important than ever. Unfortunately, the gold standard for uncertainty estimation—deep ensembles—is prohibitively expensive to train, especially on modern network architectures containing billions of parameters. In this work, we propose HyperDM, a framework capable of approximating deep ensembles at a fraction of the computational training cost. Specifically, we combine Bayesian hyper-networks and diffusion models to generate a distribution of predictions with which we can estimate total uncertainty and its epistemic and aleatoric sub-components. Our experiments on weather forecasting and CT reconstruction demonstrate that HyperDM significantly outperforms pseudo-ensembling techniques like Bayesian neural networks and Monte Carlo dropout in terms of prediction quality. Moreover, when compared against deep ensembles, HyperDM achieves up to an $M\times$ reduction in training time while yielding predictions of similar (if not superior) quality, where M is the ensemble size. This work thus makes a major stride towards developing accurate and scalable estimates of uncertainty.

Acknowledgments and Disclosure of Funding

This work was supported in part by a University of Maryland Grand Challenges Seed Grant and NSF Award No. 2425735. M.A.C. and C.A.M. were supported in part by AFOSR Young Investigator Program Award No. FA9550-22-1-0208 and ONR Award No. N00014-23-1-2752.

References

[1] A. F. Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[2] Y. Alaluf, O. Tov, R. Mokady, R. Gal, and A. Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of the IEEE/CVF conference on computer Vision and pattern recognition*, pages 18511–18521, 2022.

[3] A. N. Angelopoulos, A. P. Kohli, S. Bates, M. Jordan, J. Malik, T. Alshaabi, S. Upadhyayula, and Y. Romano. Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *International Conference on Machine Learning*, pages 717–730. PMLR, 2022.

[4] V. Balasubramanian, S.-S. Ho, and V. Vovk. *Conformal prediction for reliable machine learning: theory, adaptations and applications*. Newnes, 2014.

[5] V. Bengs, E. Hüllermeier, and W. Waegeman. Pitfalls of epistemic uncertainty quantification through loss minimisation. *Advances in Neural Information Processing Systems*, 35:29205–29216, 2022.

[6] L. Berry, A. Brando, and D. Meger. Shedding light on large generative networks: Estimating epistemic uncertainty in diffusion models. In *The 40th Conference on Uncertainty in Artificial Intelligence*.

[7] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. Accurate medium-range global weather forecasting with 3d neural networks. *Nature*, 619(7970):533–538, 2023.

[8] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[9] V. K. Chauhan, J. Zhou, P. Lu, S. Molaei, and D. A. Clifton. A brief review of hypernetworks in deep learning. *arXiv preprint arXiv:2306.06955*, 2023.

[10] J. Chen, Y. Li, L. Guo, X. Zhou, Y. Zhu, Q. He, H. Han, and Q. Feng. Machine learning techniques for ct imaging diagnosis of novel coronavirus pneumonia: A review. *Neural Computing and Applications*, pages 1–19, 2022.

[11] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning*, pages 1184–1193. PMLR, 2018.

[12] P. Dhariwal and A. Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[13] C. Ekmekci and M. Cetin. Uncertainty quantification for deep unrolling-based computational imaging. *IEEE Transactions on Computational Imaging*, 8:1195–1209, 2022.

[14] C. Ekmekci and M. Cetin. Quantifying generative model uncertainty in posterior sampling methods for computational imaging. In *NeurIPS 2023 Workshop on Deep Learning and Inverse Problems*, 2023.

[15] S. Elfwing, E. Uchibe, and K. Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107:3–11, 2018.

[16] B. T. Feng, J. Smith, M. Rubinstein, H. Chang, K. L. Bouman, and W. T. Freeman. Score-based diffusion models as principled priors for inverse imaging. In *International Conference on Computer Vision (ICCV)*. IEEE, 2023.

[17] Y. Gal. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016.

[18] 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. PMLR, 2016.

[19] A. Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

[20] D. Ha, A. Dai, and Q. V. Le. Hypernetworks, 2016.

[21] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, et al. The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, 146(730):1999–2049, 2020.

[23] J. Ho, A. Jain, and P. Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[24] A. Horé and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, 2010.

[25] G. N. Hounsfield. Computerized transverse axial scanning (tomography): Part 1. description of system. *The British journal of radiology*, 46(552):1016–1022, 1973.

[26] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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

[28] F. Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv:1702.08235*, 2017.

[29] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[30] 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.

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

[32] R. Koenker and G. Bassett Jr. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[33] A. Kristiadi, S. Däubener, and A. Fischer. Predictive uncertainty quantification with compound density networks. *arXiv preprint arXiv:1902.01080*, 2019.

[34] D. Krueger, C.-W. Huang, R. Islam, R. Turner, A. Lacoste, and A. Courville. Bayesian hypernetworks. *arXiv preprint arXiv:1710.04759*, 2017.

[35] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[36] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, et al. Learning skillful medium-range global weather forecasting. *Science*, page eadi2336, 2023.

[37] S. Lee, H. Kim, and J. Lee. Graddiv: Adversarial robustness of randomized neural networks via gradient diversity regularization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[38] J. Liu, J. Paisley, M.-A. Kioumourtzoglou, and B. Coull. Accurate uncertainty estimation and decomposition in ensemble learning. *Advances in neural information processing systems*, 32, 2019.

[39] Q. Liu, J. D. Lee, and M. I. Jordan. A kernelized stein discrepancy for goodness-of-fit tests and model evaluation, 2016.

[40] D. J. MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

[41] J. E. Matheson and R. L. Winkler. Scoring rules for continuous probability distributions. *Management science*, 22(10):1087–1096, 1976.

[42] M. T. McCann, H. Chung, J. C. Ye, and M. L. Klasky. Score-based diffusion models for bayesian image reconstruction. *arXiv preprint arXiv:2305.16482*, 2023.

[43] J. Mukhoti, A. Kirsch, J. van Amersfoort, P. H. Torr, and Y. Gal. Deep deterministic uncertainty: A new simple baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24384–24394, 2023.

[44] R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[45] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert. Attention u-net: Learning where to look for the pancreas, 2018.

[46] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, P. Hassanzadeh, K. Kashinath, and A. Anandkumar. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[47] N. Pawlowski, A. Brock, M. C. Lee, M. Rajchl, and B. Glocker. Implicit weight uncertainty in neural networks. *arXiv preprint arXiv:1711.01297*, 2017.

[48] T. Pearce, A. Brintrup, M. Zaki, and A. Neely. High-quality prediction intervals for deep learning: A distribution-free, ensembled approach. In *International conference on machine learning*, pages 4075–4084. PMLR, 2018.

[49] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[50] J. S. Schreck, D. J. Gagne II, C. Becker, W. E. Chapman, K. Elmore, G. Gantos, E. Kim, D. Kimpara, T. Martin, M. J. Molina, et al. Evidential deep learning: Enhancing predictive uncertainty estimation for earth system science applications. *arXiv preprint arXiv:2309.13207*, 2023.

[51] A. A. A. Setio, A. Traverso, T. De Bel, M. S. Berens, C. Van Den Bogaard, P. Cerello, H. Chen, Q. Dou, M. E. Fantacci, B. Geurts, et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: the luna16 challenge. *Medical image analysis*, 42:1–13, 2017.

[52] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[53] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv:2010.02502*, October 2020.

[54] Y. Song, P. Dhariwal, M. Chen, and I. Sutskever. Consistency models. *arXiv preprint arXiv:2303.01469*, 2023.

[55] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[56] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[57] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[58] N. Tagasovska and D. Lopez-Paz. Single-model uncertainties for deep learning. *Advances in neural information processing systems*, 32, 2019.

[59] M. E. Tipping. Bayesian inference: An introduction to principles and practice in machine learning. In *Summer School on Machine Learning*, pages 41–62. Springer, 2003.

[60] M. Valdenegro-Toro and D. S. Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022.

[61] J. von Oswald, C. Henning, B. F. Grewe, and J. Sacramento. Continual learning with hypernetworks. In *International Conference on Learning Representations*, 2020.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[63] L. Wu, B. L. Trippe, C. A. Naesseth, D. M. Blei, and J. P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. *arXiv preprint arXiv:2306.17775*, 2023.

[64] Y. Wu and K. He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[65] C. Zhang, J. Bütepage, H. Kjellström, and S. Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

A Training Details

All baselines are trained on a single NVIDIA RTX A6000 using a batch size of 32, an Adam [31] optimizer, and a learning rate of 1×10^{-4}. Training is run over 500 epochs in our initial experiment and 400 epochs in our CT and weather experiments. DMs are trained using a Markov chain of $T = 100$ timesteps.

A.1 Network Architecture

The backbone architecture for all baselines (i.e., BNN, DPS-UQ, HyperDM) in the toy experiment from Section 5.1 is a multi-layer perceptron (MLP) [49] with five linear layers and rectified linear unit (ReLU) [1] activation functions. For experiments described in Sections 5.2 and 5.3, we scale the DM's backbone architecture up to an Attention U-Net [45] for all baselines. The U-Net consists of an initial 2D convolutional layer, followed by four $2\times$ downsampling ResNet [21] blocks, two middle ResNet blocks, four $2\times$ upsampling ResNet blocks, and a final 2D convolutional layer. Each ResNet block consists of two 2D convolutional layers—with group normalization [64] and Sigmoid Linear Units (SiLU) [15] activation function—as well an additional attention layer.

A.2 Loss Functions

The training procedure for HyperDM is identical to that of a standard DM, except that the DM's weights are sampled from a BHN h_θ. For each training pair (x, y), we sample DM weights by first sampling random noise $z \sim \mathcal{N}(0, \sigma_z^2), z \in \mathbb{R}^8$ and then computing $\phi \sim h_\theta(z)$. We manually set the DM weights equal to ϕ and compute the loss function

$$\mathcal{L}_{\text{HyperDM}} = \|\epsilon - s(x^{(t)}, t|y, h_\theta(z))\|_2^2, \tag{14}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is the noise added to x at time step t and $s(\cdot|\phi)$ represents the DM. In general, we found HyperDM training to be stable across a variety of training hyper-parameters and did not encounter any over-fitting issues.

We follow [37] and train our BNN baseline $b(\cdot|\phi)$ by minimizing the loss function

$$\mathcal{L}_{\text{BNN}} = \|x - b(y|\phi)\|_2^2 + \lambda \text{KL}(q(\phi) \parallel p(\phi|\mathcal{D})), \tag{15}$$

which consists of a data fidelity term and an additional Kullback-Leibler (KL) divergence term between the true posterior $p(\phi|\mathcal{D})$ and the implicit distribution $q(\phi)$—approximated using Bayes by Backprop [8]. The weights of each BNN layer are sampled from a zero-mean normal distribution with standard deviation $\sigma = 0.1$, and the KL component of the loss term is down-weighted by $\lambda = 0.01$.

The training procedure for DPS-UQ and MC-Dropout are identical in that they share the same loss objective

$$\mathcal{L}_{\text{DPS-UQ}} = \mathcal{L}_{\text{MC-Dropout}} = \|\epsilon - s(x^{(t)}, t|y, \phi)\|_2^2, \tag{16}$$

where the weights ϕ are randomly initialized at the start of training and updated via backpropagation. However, we train M separate DM instances for DPS-UQ, whereas only a single DM is trained for MC-Dropout.

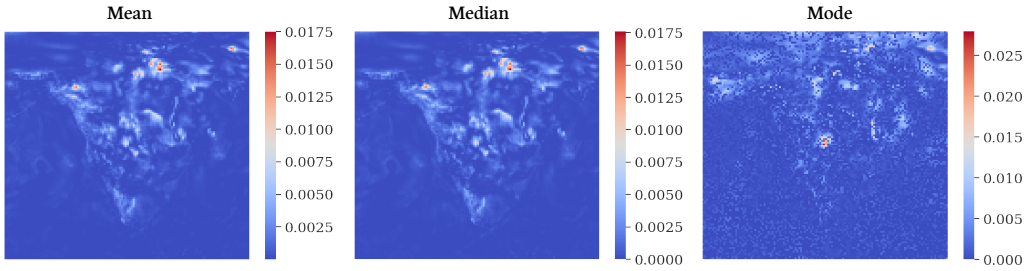

Figure 5: **Aggregation of ensemble predictions.** Ensemble predictions are aggregated using conventional methods (e.g., mean, median, mode). Mean and median aggregation results are similar, while mode aggregation results are noticeably more noisy.

Table 3: **Reconstruction quality of different ensemble aggregation methods.** HyperDM reconstruction results on the ERA5 test set are shown for three different ensemble aggregation strategies: mean, median, and mode. Best scores are highlighted in red and second best scores are highlighted in blue.

AGGREGATION	SSIM ↑	PSNR (DB) ↑	L1 ↓	CRPS ↓
MEAN	0.9455	32.93	0.018	0.01292
MEDIAN	0.9452	33.06	0.017	0.01294
MODE	0.6690	25.54	0.044	0.01293

To build a predictive distribution of size $M \times N$ with MC-Dropout, we first seed a pseudo-random number generator (RNG), which we use to deterministically sample dropout masks from a Bernoulli distribution. These masks are used to zero-out input tensor elements at each network layer. We then reset the RNG using the same initial seed—fixing the drop-out configuration—and continually sample from the DM until we obtain N predictions for that seed. This process is repeated across M different seeds for a total of $M \times N$ predictions. In all experiments, we train and test MC-Dropout with dropout probability $p = 0.3$.

B Ablation Studies

B.1 Sampling Rates

HyperDM provides flexibility at inference time to arbitrarily choose the number of network weights M to sample—analogous to the number of ensemble members in a deep ensemble—and the number of predictions N to generate per sampled weight. In this study, we examine the effect of sampling rates M, N on $\widehat{\text{EU}}$ and $\widehat{\text{AU}}$ on our OOD experiment from Section 5.3.

In our first test, we estimate EU on an OOD measurement for fixed $N = 100$ and variable $M = \{2, 4, 8, 16\}$. Results in Figure 6 indicate that under-sampling weights (i.e., $M \leq 4$) leads to uncertainty maps which underestimate uncertainty around OOD features and overestimate uncertainty around in-distribution features. However, as we continue to sample additional network weights, we observe increased uncertainty in areas around the abnormal feature and suppressed uncertainty around in-distribution features. This result indicates the importance of large ensembles in correctly isolating OOD features from in-distribution features for EU estimation.

In our second test, we repeat the same process but instead fix $M = 10$ and sample $N = \{2, 4, 8, 16\}$ predictions from the DM. Examining the results shown in Figure 7, we observe irregular peaks in the predicted AU at low sampling rates $N \leq 4$. However, as we sample more from the DM and the sample mean converges, $\widehat{\text{AU}}$ becomes more uniformly spread across the entire continental landmass. This result suggests the importance of sampling a large number of predictions for adequately capturing the characteristics of the true likelihood distribution.

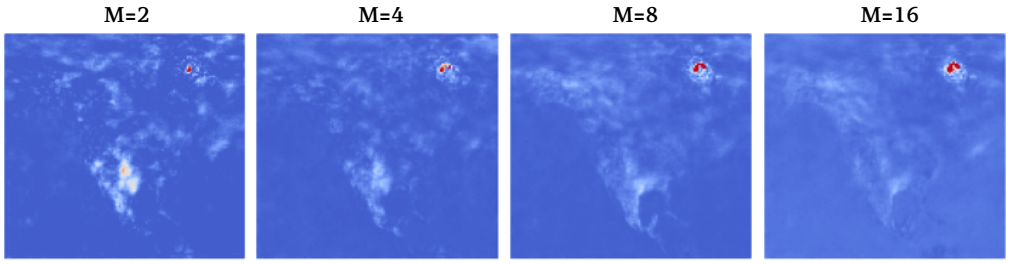

Figure 6: **Effect of sampling more weights on epistemic uncertainty.** As we increase the number M of sampled weights from the hyper-network, uncertainty around out-of-distribution features (i.e., the hot spot in the upper-right) grows and uncertainty around in-distribution features (i.e., everything else in the image) shrinks.

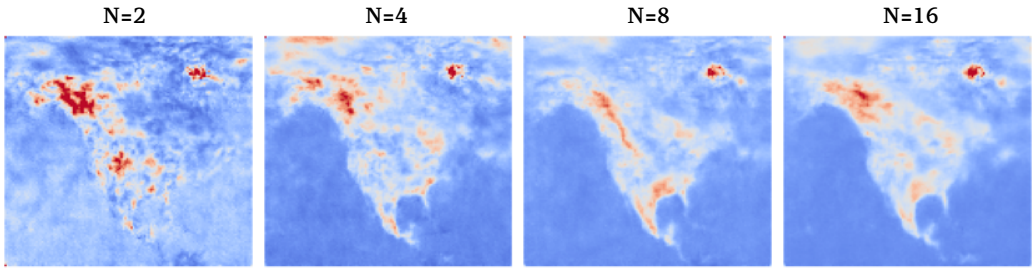

Figure 7: **Effect of sampling more predictions on aleatoric uncertainty.** As we increase the number N of sampled predictions from the diffusion model, aleatoric uncertainty predictions smooth out more evenly.

Table 4: **Ensemble prediction quality versus BNNs.** When trained on four datasets of various sizes, we observe that HyperDM produces more accurate mean predictions (as indicated by PSNR scores) and higher quality predictive distributions (as indicated by CRPS) than BNNs, except in the extreme low data regime. Highest scores are displayed in **boldface**.

DATASET SIZE	100	200	400	800	100	200	400	800
METHOD	PSNR (DB) ↑				CRPS ↓			
BNN	**10.34**	11.09	13.53	13.78	**0.20**	0.18	0.14	0.13
HYPERDM	8.47	**18.43**	**20.28**	**20.44**	0.23	**0.09**	**0.07**	**0.07**

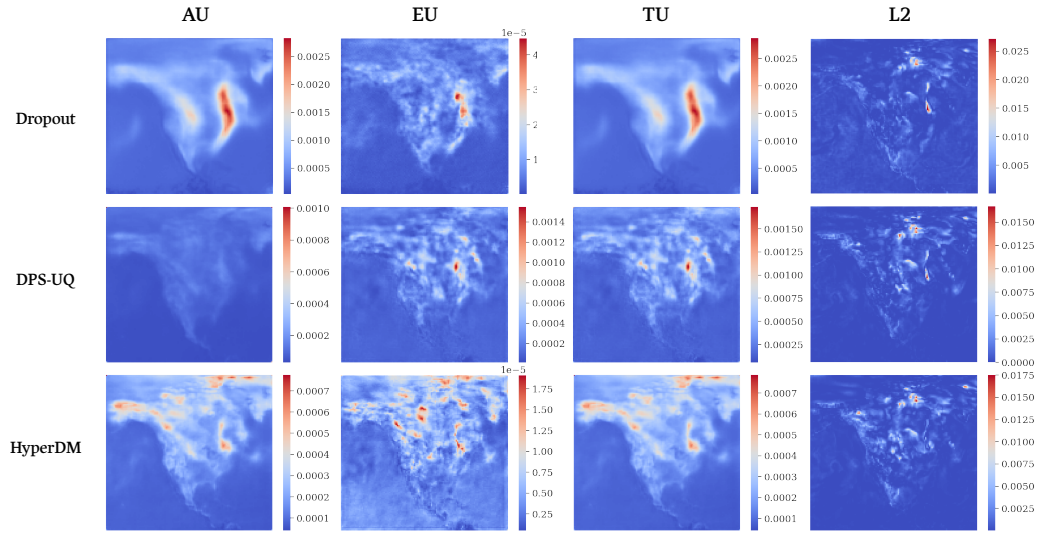

Figure 8: **Uncertainty decomposition on temperature data.** Total uncertainty and its decomposition into epistemic and aleatoric components is shown in the left three columns. The L2 error between the aggregated mean ensemble prediction and the ground truth is shown in the rightmost column. We observe higher total uncertainty around the North American continent, which corresponds with the increased L2 errors around the same areas.

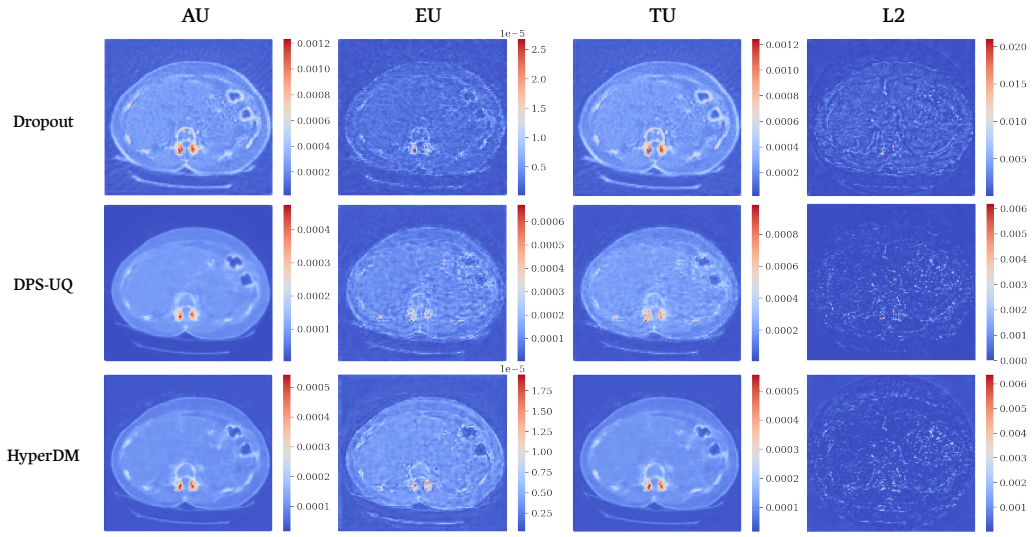

Figure 9: **Uncertainty decomposition on CT data.** Total uncertainty is high near strong features such as the spine and lining of the thoracic cavity, which corresponds to the noisy spots in L2 error map around those areas.

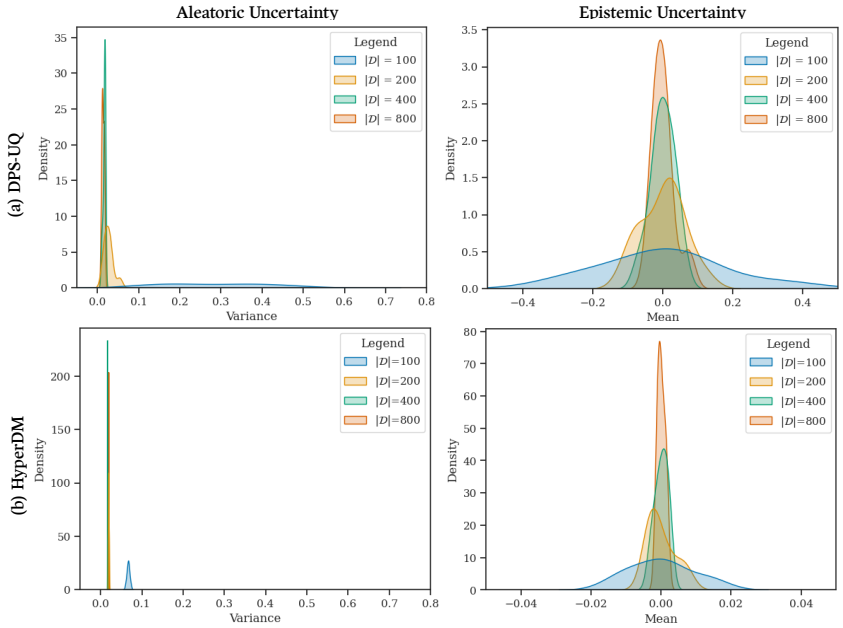

Figure 10: **Varying epistemic uncertainty.** Aleatoric and epistemic uncertainty estimates predicted by (a) DPS-UQ and (b) HyperDM when trained on dataset sizes $|\mathcal{D}| = [100, 200, 400, 800]$.

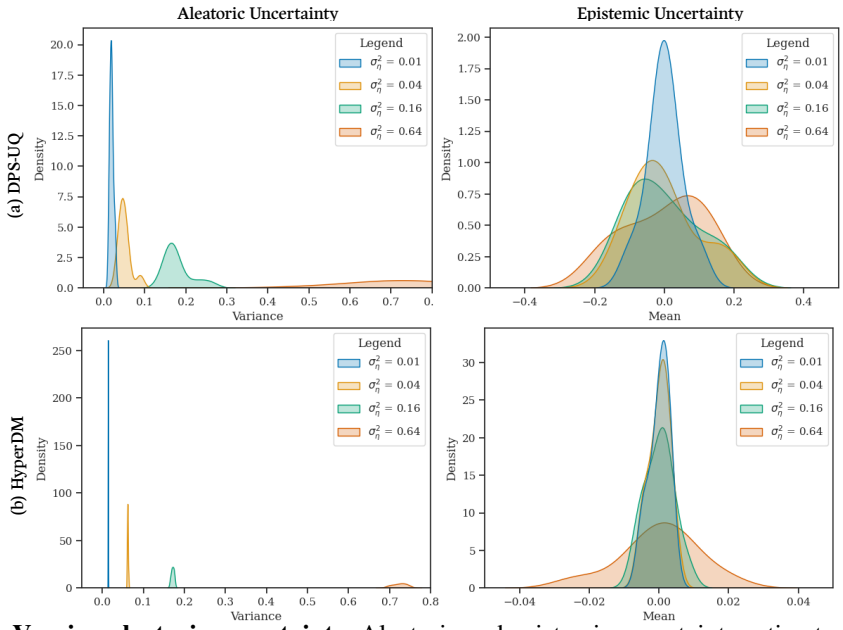

Figure 11: **Varying aleatoric uncertainty.** Aleatoric and epistemic uncertainty estimates predicted by (a) DPS-UQ and (b) HyperDM when trained on noisy datasets $\sigma_\eta^2 = [0.01, 0.04, 0.16, 0.64]$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our claims accurately reflect our contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Section 4 lists assumptions and provides relevant references where proofs can be found.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our method is fully described. Anonymized code that can replicate our method is provided in the supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Anonymized code that can replicate our method is provided in the supplementary materials.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Details can be found in Section A.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Not all our experiments include errors bars.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Our hardware resources are described in Section A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research is compliant.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive societal benefits of our work in the introduction and elsewhere. We do not foresee any negative societal impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our uncertainty estimation method does not present obvious opportunities for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We created all figures ourselves. We used PyTorch, which is disclosed.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The anonymized code in the supplement includes a readme and requirement list.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No humans subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: No IRB required. No research with humans.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.