Estimating Regression Predictive Distributions with Sample Networks

Ali Harakeh[1, 2,*], **Jordan Sir Kwang Hu**[3, 4], **Naiqing Guan**[4], **Steven Waslander**[3, 4], **Liam Paull**[1, 2]

[1]Mila - Quebec AI Institute
[2]Université de Montréal
[3]University of Toronto Institute for Aerospace Studies
[4]University of Toronto

Abstract

Estimating the uncertainty in deep neural network predictions is crucial for many real-world applications. A common approach to model uncertainty is to choose a parametric distribution and fit the data to it using maximum likelihood estimation. The chosen parametric form can be a poor fit to the data-generating distribution, resulting in unreliable uncertainty estimates. In this work, we propose SampleNet, a flexible and scalable architecture for modeling uncertainty that avoids specifying a parametric form on the output distribution. SampleNets do so by defining an empirical distribution using samples that are learned with the Energy Score and regularized with the Sinkhorn Divergence. SampleNets are shown to be able to well-fit a wide range of distributions and to outperform baselines on large-scale real-world regression tasks.

1 Introduction

Capturing uncertainty in predictions is a crucial ability for machine learning models to be safely used in decision-making systems. Predictive uncertainty is commonly modeled by predicting a probability distribution over the output variable conditioned on the input (Hüllermeier and Waegeman 2021). In this paper, we tackle the problem of predicting probability distributions for regression tasks using deep neural networks.

The most common approach for predicting probability distributions assumes the regression target to follow a particular parametric distribution (Gal and Ghahramani 2016; Lakshminarayanan, Pritzel, and Blundell 2017; Kendall and Gal 2017; Skafte, Jørgensen, and Hauberg 2019; Liu et al. 2019; Seitzer et al. 2022), and uses a neural network to learn its parameters. Such assumptions on the particular form of the predictive distribution limit the modeling capacity of these methods and lead to poor performance when the chosen distribution is a poor fit for the data being modeled.

This problem is exacerbated by the difficulty in choosing the best parametric distributions for complex, multidimensional real-world data. As an example, using a unimodal Gaussian distribution to model the inherently multimodal movement trajectory of the vehicle at an intersection (Cui et al. 2019) results in a poor approximation of the predictive distribution; we can not expect a neural network to accurately fit multimodal data using an ill-fitting parametric model, even on toy examples (figure 1).

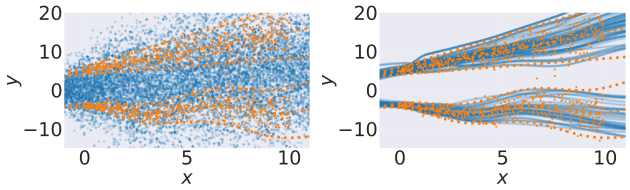


Figure 1: Left: Samples (blue) from a variance network (β-NLL) (Seitzer et al. 2022) fails to fit multimodal data (orange). Right: SampleNet produces well-fitted samples.

Summary of contributions. First, we present SampleNet, a simple and scalable neural network architecture that represents a predictive distribution with samples, avoiding the need to choose a particular parametric distribution when modeling predictive uncertainty. Second, we propose to regularize SampleNet by minimizing an optimal transport (OT) cost with respect to a data prior. Our regularization allows us to avoid overfitting while retaining the ability to encourage the predictive distribution to approximate a scaled instance of the prior if needed. Third, we empirically show that SampleNets outperform baselines on real-world multimodal data, standard regression benchmarks, and monocular depth prediction.

2 Related Work

Uncertainty can originate from two inherently different sources (Hüllermeier and Waegeman 2021), *epistemic uncertainty* which is caused by the lack of knowledge about the best model parameters, and *aleatoric uncertainty* which is caused by the inherent randomness in the underlying process being modeled. Variance networks (Nix and Weigend 1994) model aleatoric uncertainty by learning the parameters of a heteroscedastic Gaussian predictive distribution using the negative log-likelihood (NLL) loss. Variance networks have been improved to capture epistemic uncertainty by using them as members of ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) or by combining them with dropout (Kendall and

*ali.harakeh@mila.quebec

Gal 2017). Additionally, higher quality Gaussian predictive distributions have been generated with a Bayesian treatment over the variance (Stirn and Knowles 2020), learning with modified NLL losses (Skafte, Jørgensen, and Hauberg 2019; Seitzer et al. 2022), or fine-tuning with a maximum mean discrepancy loss (Cui, Hu, and Zhu 2020). Other methods replace the Gaussian parametric assumption with a Laplace distribution (Meyer and Thakurdesai 2020) or a Student's t-distribution (Skafte, Jørgensen, and Hauberg 2019) to better handle outliers during training.

The variance networks described above limit the output distribution to have a unimodal parametric form. Mixture Density Networks (MDNs) (Bishop 1994) relax this assumption by predicting a weighted mixture of Gaussian distributions, at the expense of an added difficulty in optimization due to numerical instability at higher dimensions, overfitting, and mode collapse (Makansi et al. 2019). Quantile regression (Tagasovska and Lopez-Paz 2019) allows neural networks to model complex distributions by predicting their quantiles. Unfortunately, many applications depend on a representation of the whole distribution, not just a small number quantile ranges.

Another class of methods avoid specifying a parameteric predictive distribution by generating samples that represent and empirical distribution. Implicit generative models (Mohamed and Lakshminarayanan 2016) and DiscoNets (Bouchacourt, Mudigonda, and Nowozin 2016) achieve this goal by accepting noise vectors as additional information to predict multiple samples for every input. These models are hard to train and can suffer from mode collapse (Tagasovska and Lopez-Paz 2019). Normalizing flows have also been recently employed to estimate aleatoric uncertainty (Si, Kuleshov, and Bishop 2022), but require carefully designing architectures that represent bijective functions with a tractable Jacobian.

SampleNets combine the advantages of the methods listed above to estimate aleatoric uncertainty. They avoid setting a parametric form for the predictive distribution by predicting samples from which sample statistics or quantiles can be extracted. They retain the ability to encourage samples to follow a particular parametric form by employing OT regularization, which also helps to avoid overfitting. Finally, SampleNets are easy to implement with few modifications to base architectures.

3 Preliminaries

We denote a labelled training dataset of N data pairs as $\mathcal{D} = \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$, where $\boldsymbol{x}_n \in \mathbb{R}^c$ are input features and $\boldsymbol{y}_n \in \mathbb{R}^d$ are output regression targets. The pair $\boldsymbol{x}_n, \boldsymbol{y}_n$ is assumed to be an i.i.d. sample from the joint data-generating distribution $p^*(\mathbf{y}, \mathbf{x})$, and following a true conditional distribution $p^*(\mathbf{y}|\mathbf{x})$. Let $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ be a representation of the output predictive (conditional) distribution parameterized by neural network parameters $\boldsymbol{\theta}$. Our goal is to learn the parameters $\boldsymbol{\theta}$ using $\mathcal{D}$ such that $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ closely estimates $p^*(\mathbf{y}|\mathbf{x})$.

Proper Scoring Rules

To learn $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ we minimize a *strictly proper scoring rule* $\mathcal{L}(p_{\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{x}_n), \boldsymbol{y}_n)$ (Lakshminarayanan, Pritzel, and Blundell 2017). Under strict properness (Gneiting and Raftery 2007), the minimum expected score is achieved if and only if $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x}) = p^*(\mathbf{y}|\mathbf{x})$. Using a proper scoring rule allows us to quantitatively evaluate both the sharpness and the calibration of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ (Gneiting, Balabdaoui, and Raftery 2007) on validation and test datasets, where the lower its value, the closer $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ is to $p^*(\mathbf{y}|\mathbf{x})$.

One of the most widely-used strictly proper scoring rules is the negative log-likelihood (NLL), written as

$$\begin{aligned}\mathcal{L}_{\text{NLL}}(\boldsymbol{\theta}) &= \mathbb{E}\left[-\log p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})\right] \\ &\approx \frac{1}{N}\sum_{n=1}^{N} -\log p_{\boldsymbol{\theta}}(\boldsymbol{y}_n|\boldsymbol{x}_n),\end{aligned} \tag{1}$$

where N is the number of samples. Efficiently computing the NLL requires specifying a parametric form of the output distribution $p_{\boldsymbol{\theta}}(\boldsymbol{y}_n|\boldsymbol{x}_n)$. On the other hand, the Energy Score (ES) (Gneiting and Raftery 2007; Gneiting et al. 2008) is a strictly proper scoring rule written as

$$\mathcal{L}_{\text{ES}}(\boldsymbol{\theta}) = \mathbb{E}||\hat{\mathbf{y}} - \mathbf{y}|| - \frac{1}{2}\mathbb{E}||\hat{\mathbf{y}} - \hat{\mathbf{y}}'||, \tag{2}$$

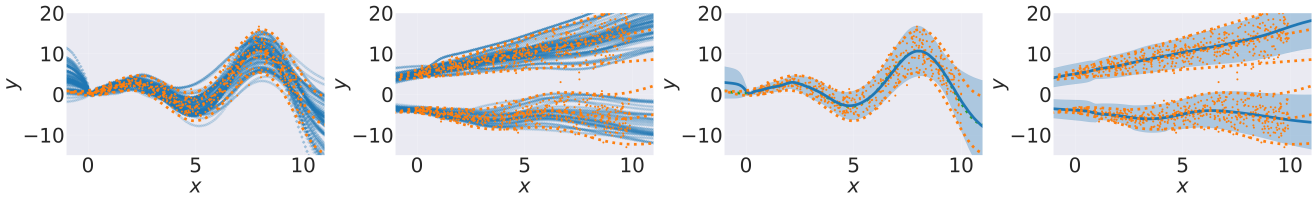
where $||.||$ is the euclidean distance, $\{\hat{\mathbf{y}}, \hat{\mathbf{y}}'\} \sim p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, and $\{\mathbf{y}\} \sim p^*(\mathbf{y}|\mathbf{x})$. Eq. 2 shows that the ES is defined using expectations, and as such *does not put restrictions on the parametric form of the predictive distribution to be learned or evaluated*. Given $\hat{\boldsymbol{y}}_{n,m} \forall m \in \{1, \dots, M\}$ drawn from $p_{\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{x}_n)$ and $\boldsymbol{y}_n$ drawn from $p^*(\mathbf{y}|\boldsymbol{x}_n)$, the ES can be approximated as

$$\begin{aligned}\mathcal{L}_{\text{ES}}(\boldsymbol{\theta}) \approx \frac{1}{N}\sum_{n=1}^{N}\Bigg(&\frac{1}{M}\sum_{i=1}^{M}||\hat{\boldsymbol{y}}_{n,i} - \boldsymbol{y}_n|| \\ &- \frac{1}{2M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}||\hat{\boldsymbol{y}}_{n,i} - \hat{\boldsymbol{y}}_{n,j}||\Bigg),\end{aligned} \tag{3}$$

In sect. 5 we will be using eq. 3 to train SampleNet. We will also use eq. 3 alongside the (Gaussian) NLL in eq. 1 to rank the predictive distributions generated from all methods. For more details on proper scoring rules and the energy score, we refer the reader to (Gneiting and Raftery 2007; Harakeh and Waslander 2021).

Geometric Divergences

Norms and divergences commonly used to compare two probability distributions operate in a point-wise manner, failing to capture the geometric nature of the problem when distributions have non-overlapping supports (Feydy et al. 2019). Two families of distances that account for the geometry of the underlying space are Maximum Mean Discrepancies (MMD) and Optimal Transport (OT) distances. OT distances have appealing geometric properties (Villani 2009), but suffer from poor sample efficiency while being computationally very expensive. MMDs on the other hand are cheaper to compute and have a small sample complexity, but require significant tuning of their kernel's bandwidth parameter (Genevay, Peyré, and Cuturi 2018) and can have vanishing gradients (Feydy et al. 2019).



(a) Unimodal Samples (b) Multimodal Samples (c) Sample Statistics (d) HPD Intervals

Figure 2: (a) and (b): Samples (blue) predicted by SampleNet to estimate a Gaussian distribution and a multimodal distribution, respectively. Training data and the groundtruth 95% confidence intervals of the data-generating distributions are plotted in orange. (c): The mean (blue line) and the predicted 95% confidence interval (shaded region) extracted from samples in (a). (d): The modes (blue line) and the 75% highest posterior density (HPD) intervals (shaded region) extracted from samples in (b).

Sinkhorn Divergences (Genevay, Peyré, and Cuturi 2018) are a recently proposed class of geometric divergences that interpolate between OT distances and MMDs (Feydy et al. 2019), achieving advantages of both by being differentiable, cheap to compute, and scalable to large batch sizes with low sample complexity. We follow (Genevay, Peyré, and Cuturi 2018) and use the entropy regularized Wasserstein-2 distance, $W_c^\epsilon(.,.)$ (Cuturi 2013), with respect to a ground metric c to define the Sinkhorn Divergence as,

$$S_c^\epsilon(\boldsymbol{\alpha}, \boldsymbol{\beta}) = W_c^\epsilon(\boldsymbol{\alpha}, \boldsymbol{\beta}) - \frac{1}{2}\left(W_c^\epsilon(\boldsymbol{\alpha}, \boldsymbol{\alpha}) + W_c^\epsilon(\boldsymbol{\beta}, \boldsymbol{\beta})\right), \quad (4)$$

where $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are the two probability distributions we want to compare, and $\epsilon > 0$ is the entropic regularization strength. In sect. 4 we will use eq. 4 for geometric regularization when training SampleNets, where we use $\epsilon = 0.0025$ and $c = 2$.

4 SampleNets

The key idea in SampleNets is to predict a set of M samples that represent an empirical predictive distribution $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$. This can be achieved by modifying the output layer of any regression architecture to predict $M \times d$ values instead of their usual $1 \times d$ output. The samples $\{\hat{\boldsymbol{y}}_{n,1} \dots \hat{\boldsymbol{y}}_{n,M}\}, \hat{\boldsymbol{y}}_{n,i} \in \mathbb{R}^d$ are trained to match the groundtruth sample $\boldsymbol{y}_n$ by minimizing the ES in eq. 3.

Predicting samples from $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$ eliminates the need for overly restrictive assumptions on the parametric form of $p_{\boldsymbol{\theta}}(\mathbf{y}|\mathbf{x})$, resulting in a general architecture that can learn to fit complex (multimodal, skewed) data-generating distributions. We provide a qualitative example in figure 2, where the same architecture and hyperparameters are used to predict samples that fit a unimodal Gaussian distribution in figure 2a, and a multimodal distribution in figure 2b. A description of the toy data can be found in sect. A. From predicted samples, we can also estimate summaries of probability distributions such as sample statistics (figure 2c) or Highest Posterior Density (HPD) (Hyndman 1996) intervals (figure 2d).

From figure 2, we observe that SampleNets learn M functions of predicted samples over the input space, leading to great flexibility when learning complex distributions. Unfortunately, such flexibility is potentially problematic in that it could lead to a portion of the learned functions overfitting to outliers in training data, particularly when the data-generating distribution is simple. Such overfitting behavior is not specific to SampleNets and can be observed in other over-parameterized architectures like MDNs (Makansi et al. 2019). A second issue that can arise while training SampleNets is that the sample-based losses used for training suffer from a quadratic memory footprint, causing poor scalability as the number of predicted samples increases. We address these two concerns in sections 4 and 4 respectively.

Regularization with the Sinkhorn Divergence

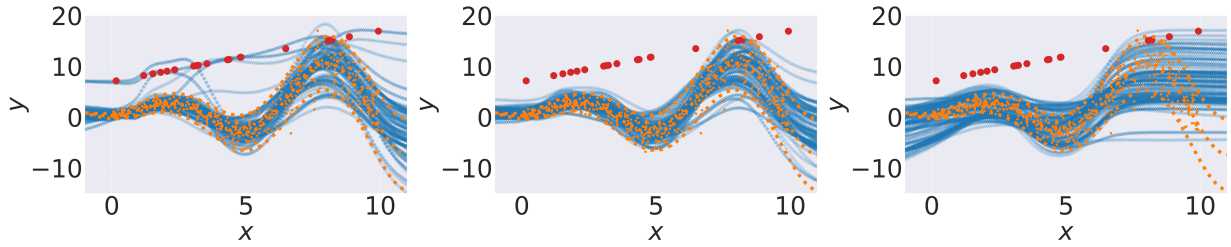
The goal of our proposed regularization is twofold. First, we want to avoid the overfitting behavior that originates from overparameterization when training with the ES. Second, we want to preserve the performance that is obtained by state-of-the-art methods when the underlying predictive distribution actually does follow a simple parametric form, e.g. when the data generating distribution is truly Gaussian.

Inspired by noise contrastive priors (Hafner et al. 2020), we propose to constrain the parameters $\boldsymbol{\theta}$ of SampleNets to predict samples $\{\hat{\boldsymbol{y}}_{n,1}, \dots, \hat{\boldsymbol{y}}_{n,M}\}$ by minimizing the optimal transport cost of moving the samples from a data prior $\tilde{\boldsymbol{y}} \sim p_{ot}(\mathbf{y}|\mathbf{x})$. We achieve this by training SampleNets with a combined loss,

$$\begin{aligned} \mathcal{L}_{\text{total}}(\boldsymbol{\theta}) &= \mathcal{L}_{\text{ES}}(\boldsymbol{\theta}) + \eta \mathcal{L}_{S_c^\epsilon}(\boldsymbol{\theta}) \\ \mathcal{L}_{S_c^\epsilon}(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{n=1}^{N} S_c^\epsilon(p_{ot}(\mathbf{y}|\boldsymbol{x}_n), p_{\boldsymbol{\theta}}(\mathbf{y}|\boldsymbol{x}_n)), \end{aligned} \quad (5)$$

where $\mathcal{L}_{\text{ES}}$ is the ES loss from eq. 3, S_c^ϵ is the Sinkhorn Divergence from eq. 4, and η is a hyperparameter that sets the strength of the regularization term.

We can choose $p_{ot}(\mathbf{y}|\boldsymbol{x})$ to be a standard distribution (Uniform, Gaussian) and then normalize the predicted samples $\{\hat{\boldsymbol{y}}_{n,1}, \dots \hat{\boldsymbol{y}}_{n,M}\}$ before computing the Sinkhorn Divergence. Normalization allows us to decouple the choice of the prior from the magnitude of the specific problem at hand. Specifically, normalization allows the Sinkhorn Divergence in 5 to be minimized when used with a standard prior distribution as long as the predicted samples follow a scaled version of this prior. The Sinkhorn Divergence only penalizes the shape of the output distribution as it diverges from the prior, preventing the regularization loss from overly restricting the support of the output distribution. More details on normalization can be found in sect. A.



(a) $\eta = 0$ (b) $\eta = 2$ (c) $\eta = 200$

Figure 3: $M = 100$ samples predicted by a SampleNet for various values of η in eq. 5. The groundtruth distribution and training data are shown in orange. Synthetic outliers are shown in red.

We visualize the impact of our proposed regularization in figure 3 when using a standard Gaussian prior distribution. Using the same choice of hyperparameters, we train a SampleNet to predict $M = 100$ samples using 500 training data points from a unimodal Gaussian distribution (orange). Additionally, we append 20 synthesized outliers to the training data, which we show in red. Figure 3a shows that when training SampleNet with $\eta = 0$, some of the 100 learned functions are assigned to fit the outliers in training data. When regularization with ($\eta = 2$) (figure 3b) is added, the overfitting behavior is reduced as the optimal transport cost penalizes the predicted samples from fitting the training data outliers, while simultaneously pushing those to follow a Gaussian distribution. On the other hand, figure 3c shows that at a large value of $\eta = 200$, the regularization loss overwhelms the total loss in eq. 5, which leads to the output samples underfitting the training data. Specifically, at a high regularization strength, the output samples are pushed to ignore the geometry of the training data as the total loss can sufficiently be minimized by predicting an arbitrary Gaussian distribution for any input $\boldsymbol{x}_n$. We treat η as a tunable hyperparameter and study its impact in sect. 5.

Reducing Compute Costs with Minibatch Losses

The sample-based losses in equations 3 and 4 suffer from a quadratic compute time and memory growth as a function of M. To resolve this issue, we propose to use minibatch subsampling proposed by Genevay *et al.* (Genevay, Peyré, and Cuturi 2018) to generate unbiased estimators for the ES and the Sinkhorn Divergence that are computed with a lower sample size $K < M$. Given a number of repetitions $L > 0$, we write the minibatch ES loss as,

$$\mathcal{L}_{\text{ES}}(\theta) = \frac{1}{NL}\sum_{n=1}^{N}\sum_{l=1}^{L}\Bigg(\frac{1}{K}\sum_{i=1}^{K}||\hat{\boldsymbol{y}}_{n,i}^{(l)} - \boldsymbol{y}_n|| - \frac{1}{2K^2}\sum_{i=1}^{K}\sum_{j=1}^{K}||\hat{\boldsymbol{y}}_{n,i}^{(l)} - \hat{\boldsymbol{y}}_{n,j}^{(l)}||\Bigg), \quad (6)$$

where the samples $\hat{\boldsymbol{y}}_{n,i}^{(l)} \forall i \in \{1, \dots, K\}$ are the K elements sampled without replacement from $\{\hat{\boldsymbol{y}}_{n,1}, \dots \hat{\boldsymbol{y}}_{n,M}\}$ during repetition l. Similarly, we can write the minibatch subsampled Sinkhorn Divergence loss as:

$$\mathcal{L}_{S_c^\epsilon}(\boldsymbol{\theta}) = \frac{1}{NL}\sum_{n=1}^{N}\sum_{l=1}^{L} S_c^\epsilon(\{\tilde{\boldsymbol{y}}_{n,1}^{(l)}, \dots \tilde{\boldsymbol{y}}_{n,K}^{(l)}\}, \{\hat{\boldsymbol{y}}_{n,1}^{(l)}, \dots \hat{\boldsymbol{y}}_{n,K}^{(l)}\}). \quad (7)$$

The ES originates from the Energy Distance (Rizzo and Székely 2016), which in turn is an MMD (Sejdinovic et al. 2013) and as such has unbiased gradient estimates when used with minibatch subsampling (Genevay, Peyré, and Cuturi 2018). Fatras *et al.* (Fatras et al. 2019) studied the theoretical properties of the minibatch Sinkhorn Divergence, which was also shown to have unbiased gradients with respect to model parameters. We empirically study the impact of the values of M, K, and L on the performance of SampleNets in sect. 5.

5 Experiments and Results

Details of our experimental setup can be found in sect. A. We perform experiments to answer three questions and draw the following conclusions.

Can SampleNet estimate multimodal probability distributions? We test SampleNet on two real-world datasets with data generating distributions that are multimodal and show its ability to accurately predict these distributions with no assumptions on their parametric form.

How does SampleNet perform in comparison to distributional regression baselines? We compare SampleNet to other lightweight distributional regression methods on real-world regression datasets and on monocular depth prediction. SampleNet is shown to perform on par or better than all tested baselines.

How sensitive is SampleNet to its hyperparameters? We study the impact of the number of samples M, the minibatch sample size K, the number of repetitions L, and the regularization strength η on the performance of SampleNets when predicting multimodal distributions. We observe the performance of SampleNet to be most sensitive to the regularization strength η, which we recommend being prioritized during hyperparameter optimization. We also observe that SampleNet produces high-quality predictive distributions with a small number of samples M, increasing M, L, and K does not lead to a substantial increase in performance on most of our real-world datasets.

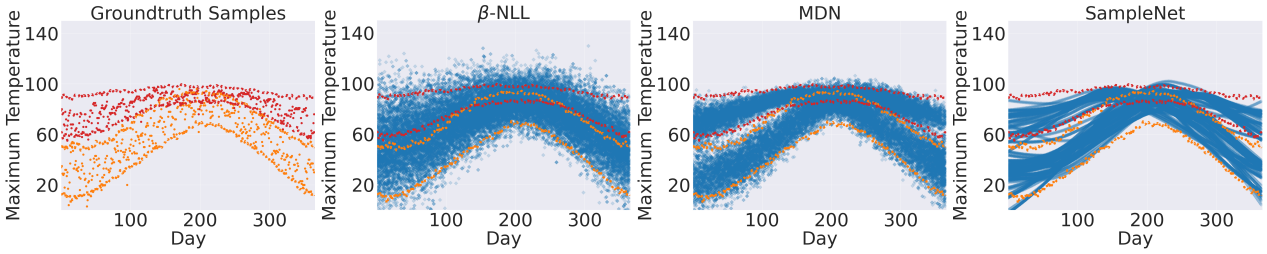


Figure 4: Scatter plots of samples (blue) from SampleNet and baselines on the Weather dataset. The test set 95% confidence intervals of maximum daily temperatures from two different weather stations is shown in red and orange. Groundtruth samples used for training are shown in the leftmost plot.

Multimodal Datasets

To determine SampleNet's capacity to learn complex probability distributions, we perform experiments on the Weather (figure 4) and Traffic datasets (figure A.5), both of which have been demonstrated to be multimodal. More details on both datasets as well as the training hyperparameters can be found in appendix A. Since the datasets are not Gaussian, we use the ES (eq. 3) as an evaluation metric. We repeat all experiments using 5 random seeds and report the standard deviation.

Table 1 shows a quantitative comparison of SampleNet, MDN, and β-NLL. SampleNet is shown to outperform both β-NLL and MDN by a wide margin when comparing the ES on both datasets.

In addition, figures 4 and fig. A.5 qualitatively show that SampleNet predicts good approximations of the test set distributions on both datasets. On the Weather dataset, MDN is shown to be able to predict a reasonable estimate of the bimodal data-generating distribution. On the Traffic dataset, MDN suffers from mode collapse (Makansi et al. 2019), where the model uses a single Gaussian distribution to explain the data. This is evident from MDN's high value of the ES on the Traffic dataset in table 1, as well as from its samples failing to model the multiple modes of the Traffic test set distribution in figure A.5. β-NLL exhibits poor performance on both datasets as it is restricted to estimating a unimodal Gaussian distribution, which we visually verify in figures 4 and A.5. Other unimodal parametric models also fail to learn multimodal distributions, we provide their results on the Weather dataset in table A.1 of the appendix.

Real-world Regression Datasets

We evaluate the performance of SampleNet in comparison to baselines on datasets from the UCI regression benchmark (Dua and Graff 2017). We use 13 datasets that have been used in the evaluation of recent methods from literature such as β-NLL (Seitzer et al. 2022) and Student-t regression (Skafte, Jørgensen, and Hauberg 2019). Following (Seitzer et al. 2022), we also add 2 robot dynamics models datasets, ObjectSlide (Seitzer, Schölkopf, and Martius 2021) and Fetch-PickAndPlace (Seitzer et al. 2022), to our evaluation for a total of 15 datasets. For all methods, we report the mean and standard deviation of the ES in table 2 and the Gaussian NLL in table 3 computed over 20 train-test splits, similar to (Seitzer et al. 2022). We report values for the RMSE in table A.4 of the appendix. We also perform a two-sided Kolmogorov-Smirnov test with a $p \leq 0.05$ significance level and mark in bold any methods that produce statistically indistinguishable results when compared to the best-performing method in tables 2, 3, and A.4. More details on the datasets and hyperparameters can be found in appendix A.

Dataset	β-NLL	MDN	SampleNet
Weather	0.59 ± 0.03	0.21 ± 0.06	**0.16 ±0.00**
Traffic	133.88 ± 3.13	117.21 ± 4.16	**12.56 ± 0.38**

Table 1: Comparison with baselines on multimodal datasets. We report the mean ES (lower is better) ± the standard deviation over 5 random seeds.

Table A.4 shows that SampleNet as well as most other baselines perform equally well when measuring their RMSE, meaning that performance improvements on proper scoring rules can be attributed to improvements in estimating the predictive distribution. When aggregating results overall 15 datasets, SampleNet exhibits performance that is better or on par with other baselines on 15 out of 15 datasets when using the ES for evaluation (last row of table 2) and 12 out of 15 datasets when using the Gaussian NLL for evaluation (last row of table 3). In addition, SampleNet results in a statistically significant improvement in ES on 8 out of 15 datasets and a statistically significant improvement in Gaussian NLL on 2 out of 15 datasets when compared to the rest of our baselines.

Hyperparameter Sensitivity Analysis

One disadvantage of SampleNet is the introduction of 4 additional hyperparameters to optimize for: the predicted sample size M, the minibatch loss sample size K, the minibatch loss number of repetitions L, the OT regularization strength parameter η. Using the Weather and Traffic datasets, we perform a sensitivity analysis to understand the impact of varying these hyperparameters by training SampleNets over all combinations of $M = \{50, 100, 200, 400\}$, $K = \{50, 100, 200\}$, $L = \{1, 2, 3, 4\}$, and $\eta = \{0.0, 0.1, 0.5, 1.0, 5.0\}$. We use a Gaussian prior for regularization.

Dataset	Dropout	β-NLL	Student-t	MDN	SampleNet
Boston	**1.572 ± 0.297**	1.938 ± 0.344	**1.662 ± 0.346**	**1.644 ± 0.294**	**1.559 ± 0.355**
Kin8nm	0.056 ± 0.002	0.044 ± 0.002	0.048 ± 0.002	**0.041 ± 0.001**	**0.042 ± 0.001**
Power	11.317 ± 0.049	**2.282 ± 0.061**	**2.289 ± 0.049**	2.325 ± 0.063	**2.263 ± 0.056**
Yacht	0.842 ± 0.155	3.718 ± 0.57	4.67 ± 0.834	0.597 ± 0.198	**0.179 ± 0.056**†
Concrete	3.082 ± 0.257	3.416 ± 0.267	3.301 ± 0.255	**3.069 ± 0.267**	**2.823 ± 0.299**†
Wine (red)	0.359 ± 0.019	0.359 ± 0.020	**0.361 ± 0.023**	0.362 ± 0.020	**0.340 ± 0.020**
Wine (white)	0.397 ± 0.010	0.398 ± 0.009	0.394 ± 0.012	0.395 ± 0.010	**0.379 ± 0.009**
Naval	0.038 ± 1e-4	0.003 ± 0.003	0.004 ± 0.001	0.003 ± 0.003	**0.001 ± 7e-5**†
Superconductivity	6.730 ± 0.205	6.593 ± 0.294	6.597 ± 0.139	6.415 ± 0.122	**6.035 ± 0.127**†
Protein	3.017 ± 0.054	2.392 ± 0.016	2.474 ± 0.026	**2.116 ± 0.031**	**2.120 ± 0.018**
Year	48.288 ± 0.058	4.955 ± 0.264	4.784 ± 0.088	4.800 ± 0.087	**4.606 ± 0.017**
ObjectSlide	0.010±7e-5	**34e-4±7e-5**	47e-4±6e-5	**33e-4±6e-5**	**35e-4± 3e-5**†
Energy	1.319 ± 0.079	0.936 ± 0.165	0.963 ± 0.221	1.211 ± 0.223	**0.729 ± 0.068**
Carbon	0.027 ± 0.028	0.061 ± 0.005	0.006 ± 0.001	**0.004 ± 0.001**	**0.004 ± 0.001**
Fetch-Pick&Place	35e-4± 1e-5	24e-4± 1e-5	27e-4±9e-5	35e-4±2e-5	**20e-4±2e-5**†
Total Top Performance	1	2	3	6	**15**

Table 2: The mean and standard deviation of the ES over the train-test splits of the real-world regression datasets. Best performing methods are marked in bold. The † symbol indicates a SampleNet trained without OT regularization ($\eta = 0$).

Figures A.6a and A.6b show the impact of varying η on the ES for various values of M on the Weather and Traffic datasets, respectively, aggregated over all values of L and K. On the Traffic dataset, η is shown to have a minimal impact on performance. On the other hand, results on the Weather dataset show that increasing η results in a large decrease in the ES for all values of M. When noting that the Traffic dataset contains 6 times the amount of training data when compared to the Weather dataset, our proposed regularization improves the results on the smaller dataset. In addition, the Weather dataset is originally generated from two unimodal Gaussian distributions, which could benefit from our regularization loss driving the solution to optimally transport a Gaussian prior. We notice similar sensitivity to the value of η when testing on real-world datasets in Table 2, where we find no consistency in the value of η or the choice of prior among the 15 datasets when looking at the best performing SampleNet configuration.

Figure A.6c shows the impact of varying K on the ES for various values of M on Traffic datasets, aggregated over all values of L and η. For a fixed value of M, increasing K results in an improvement in the accuracy of predicted distributions as the minibatch losses in eq. 6 and eq. 7 become a more accurate approximation of the original losses in eq. 3 and eq. 4. The results in figure A.6d show a similar behavior, where the ES decreases when increasing the number of repetitions L in eq. 6 and eq. 7 for a fixed value of K on the Traffic datasets. Similar results on the Weather dataset can be found in figure A.3 of the appendix. The most important observation from figure A.3 is that the minimum achievable ES (lower point of the box plots) does not substantially decrease as we increase the number of output samples M, meaning that we can achieve similar performance using any value of M and enough hyperparameter optimization. A similar behavior is observed on UCI and dynamics models datasets, where we notice that the best performing SampleNet configurations are in table A.3 used $M < 200$ for 11 out of 15 datasets. These observations alleviate a major concern about the dependence of our sample-based approach on a large output sample size to achieve good performance. When applying SampleNet to a new problem, our sensitivity analysis indicates that the best approach is to choose the highest value of M that fits the computational memory and time constraints with K=M and L=1. Hyperparameter optimization can then be performed for η, the choice of prior, as well as any other standard neural network hyperparameters such as the learning rate. Our minibatch subsampling losses leave the option for practitioners to increase the number of output samples M if needed while maintaining adequate performance.

Monocular Depth Prediction

Following our own recommendations on setting hyperparameters, we use SampleNet to estimate predictive distributions on the large-scale computer vision task of monocular depth prediction. We use the NYUv2 dataset (Nathan Silberman and Fergus 2012) for training and testing. We modify the same base depth prediction architecture used by β-NLL in (Seitzer et al. 2022) to generate samples and train using the loss in 5. We compare the performance of SampleNet on deterministic metrics and proper scoring rules to three baselines β-NLL, Student-t regression, and MDN. More details about the experimental setup and metrics can be found in appendix A.

Our compute resources allow us to set a maximum value of $M = 25$ when K=M and L=1 before running out of

Dataset	Dropout	β-NLL	Student-t	MDN	SampleNet
Boston	$\mathbf{2.634 \pm 0.353}$	2.768 ± 0.325	$\mathbf{2.589 \pm 0.362}$	$\mathbf{2.781 \pm 0.651}$	$\mathbf{2.413 \pm 0.273}$
Kin8nm	-0.795 ± 0.057	-1.174 ± 0.035	-1.087 ± 0.038	$\mathbf{-1.242 \pm 0.028}$	-1.213 ± 0.025
Power	4.751 ± 0.002	$\mathbf{2.839 \pm 0.048}$	$\mathbf{2.833 \pm 0.037}$	$\mathbf{2.859 \pm 0.038}$	$\mathbf{2.851 \pm 0.037}$
Yacht	1.799 ± 0.113	3.465 ± 0.258	5.692 ± 2.59	$\mathbf{0.214 \pm 0.423}$	$\mathbf{0.033 \pm 0.188}^{\dagger}$
Concrete	3.392 ± 0.305	3.314 ± 0.144	3.172 ± 0.122	$\mathbf{3.320 \pm 0.589}$	$\mathbf{3.060 \pm 0.138}^{\dagger}$
Wine (red)	$\mathbf{0.980 \pm 0.062}$	1.117 ± 0.027	156.548 ± 386.887	1.226 ± 0.806	$\mathbf{0.945 \pm 0.068}$
Wine (white)	1.109 ± 0.038	1.076 ± 0.033	$\mathbf{1.058 \pm 0.026}$	2.560 ± 6.358	$\mathbf{1.045 \pm 0.024}$
Naval	-1.819 ± 0.071	$\mathbf{-10.045 \pm 0.481}$	-7.873 ± 0.659	-9.974 ± 0.401	$\mathbf{-10.224 \pm 0.146}^{\dagger}$
Superconductivity	5.501 ± 0.238	$\mathbf{3.721 \pm 0.197}$	4.260 ± 2.465	3.983 ± 0.443	$\mathbf{3.602 \pm 0.041}^{\dagger}$
Protein	10.928 ± 0.208	3.105 ± 0.313	4.996 ± 2.640	$\mathbf{2.912 \pm 0.106}$	$\mathbf{2.845 \pm 0.007}$
Year	6.197 ± 0.001	$\mathbf{3.510 \pm 0.060}$	1496.605 ± 1308.447	$\mathbf{3.579 \pm 0.047}$	$\mathbf{3.531 \pm 0.005}$
ObjectSlide	-2.287 ± 0.148	$\mathbf{-4.996 \pm 1.015}$	-4.010 ± 0.347	-3.461 ± 2.054	$-3.016 \pm 0.309^{\dagger}$
Energy	3.763 ± 0.070	2.323 ± 0.390	2.450 ± 0.562	2.844 ± 1.239	$\mathbf{2.001 \pm 0.299}$
Carbon	-3.967 ± 0.043	$\mathbf{-6.854 \pm 9.193}$	$\mathbf{-10.589 \pm 0.856}$	39.275 ± 47.959	-7.461 ± 3.283
Fetch-Pick&Place	-12.703 ± 0.337	-14.613 ± 0.383	-14.078 ± 0.184	-13.092 ± 0.372	$\mathbf{-14.800 \pm 0.088}^{\dagger}$
Total Top Performance	2	7	4	7	**12**

Table 3: The mean and standard deviation of the Gaussian NLL over the train-test splits of the real-world regression datasets. Best performing methods are marked in bold. For computing the Gaussian NLL for SampleNet, we use the empirical mean and variance of the predicted samples.

	Deterministic Metrics		Scoring Rules		Average Var	
Method	RMS↓	SI log↓	NLL↓	ES↓	All	Border
β-NLL (Seitzer et al. 2022)	0.3825	11.346	7.237 ± 6.85e-1	0.2296 ± 1.43e-3	0.0304	0.0533
Student-t (Skafte, Jørgensen, and Hauberg 2019)	0.3868	11.494	7.187 ± 1.56e-1	**0.2274** ± 7.09e-4	0.0410	0.0731
MDN (Bishop 1994)	0.3922	11.722	8.380 ± 9.59e-1	0.2321 ± 2.72e-4	0.0445	0.0749
SampleNet†	0.3795	11.332	**1.397** ± 7.05e-4	0.2371 ± 5.43e-4	0.0157	0.0340
SampleNet	**0.3757**	**11.195**	**1.398** ± 1.22e-3	0.2354 ± 4.01e-3	0.0115	0.0261

Table 4: Results on the NYUv2 dataset. SampleNet trained with ($\eta = 0$) is marked with †.

memory. For these values of M, L, and K, we train SampleNets with $\eta = \{0.0, 0.1, 0.5, 1.0\}$ using a Gaussian prior, and show the results for the best performing configuration ($\eta = 1.0$) as well as the unregularized variant ($\eta = 0.0$) in Table 4. Using SampleNet leads to a significant reduction in Gaussian NLL on the test set compared to other baselines while maintaining a comparable value of the ES, mainly because SampleNet generates much sharper distributions. The sharpness property can be quantitatively shown in the last two columns of table 4, where distributions predicted from SampleNet are shown to have a much lower average variance when compared to those from baselines. Qualitative evidence of sharper distributions is also provided in figure A.7 of the appendix. SampleNet is also shown to outperform all baselines on standard deterministic depth prediction metrics, meaning that any improvement in predictive distributions does not come at the cost of accuracy on the original depth prediction problem.

Our findings align well with those in previous literature, where Gaussian parametric distributions predicted from neural networks are found to have high variance if learned on high variance training data using the NLL (Harakeh and Waslander 2021). Due to discontinuities of depth values at object boundaries in images, we hypothesize that the baselines learn high variance distributions to accommodate for their inability to model multiple depth modes. SampleNet seems to resolve this issue, with the regularized configuration leading to even sharper distributions when compared to the unregularized configuration (last two rows of table 4).

6 Conclusion

We present SampleNet, a scalable approach to model uncertainty in deep neural networks without imposing a parametric form on the predictive distribution. Our simple solution provides an easy-to-implement tool for enabling the estimation of regression predictive distributions on real-world problems. One limitation of SampleNet is only accurately predicts the uncertainty of in-distribution data. Modifying SampleNet's regularization to output high uncertainty values on out-of-distribution data (e.g. as in noise-contrastive priors (Hafner et al. 2020)) would be an interesting avenue for future work. Example code can be found at: https://samplenet.github.io/.

Acknowledgements

A. H. was supported by the DEEL project and the IVADO postdoctoral funding. L. P. is supported by the Canada CIFAR AI Chairs Program. The work was also supported by

the National Science and Engineering Research Council of Canada under the Discovery Grant Program.

References

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

Bouchacourt, D.; Mudigonda, P. K.; and Nowozin, S. 2016. Disco nets: dissimilarity coefficients networks. In *Advances in Neural Information Processing Systems*, volume 29.

Cui, H.; Radosavljevic, V.; Chou, F.-C.; Lin, T.-H.; Nguyen, T.; Huang, T.-K.; Schneider, J.; and Djuric, N. 2019. Multimodal trajectory predictions for autonomous driving using deep convolutional networks. In *2019 International Conference on Robotics and Automation (ICRA)*, 2090–2096.

Cui, P.; Hu, W.; and Zhu, J. 2020. Calibrated reliable regression using maximum mean discrepancy. In *Advances in Neural Information Processing Systems*, 17164–17175.

Cuturi, M. 2013. Sinkhorn distances: lightspeed computation of optimal transport. In *Advances in neural information processing systems*, volume 26.

Dua, D.; and Graff, C. 2017. UCI machine learning repository. http://archive.ics.uci.edu/ml. Accessed: 2023-02-21.

Fatras, K.; Zine, Y.; Flamary, R.; Gribonval, R.; and Courty, N. 2019. Learning with minibatch Wasserstein: asymptotic and gradient properties. In *arXiv preprint arXiv:1910.04091*.

Feydy, J.; Séjourné, T.; Vialard, F.-X.; Amari, S.-i.; Trouve, A.; and Peyré, G. 2019. Interpolating between optimal transport and MMD using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, 2681–2690.

Gal, Y.; and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International conference on machine learning*, 1050–1059.

Genevay, A.; Peyré, G.; and Cuturi, M. 2018. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, 1608–1617.

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

Gneiting, T.; and Raftery, A. E. 2007. Strictly proper scoring rules, prediction, and estimation. In *Journal of the American statistical Association*, 359–378.

Gneiting, T.; Stanberry, L. I.; Grimit, E. P.; Held, L.; and Johnson, N. A. 2008. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. In *Test*, 211.

Hafner, D.; Tran, D.; Lillicrap, T.; Irpan, A.; and Davidson, J. 2020. Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, 905–914.

Harakeh, A.; and Waslander, S. L. 2021. Estimating and evaluating regression predictive uncertainty in deep object detectors. In *International Conference on Learning Representations*.

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

Hyndman, R. J. 1996. Computing and graphing highest density regions. In *The American Statistician*, 120–126.

Kendall, A.; and Gal, Y. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, volume 30.

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

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

Makansi, O.; Ilg, E.; Cicek, O.; and Brox, T. 2019. Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7144–7153.

Meyer, G. P.; and Thakurdesai, N. 2020. Learning an uncertainty-aware object detector for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10521–10527.

Mohamed, S.; and Lakshminarayanan, B. 2016. Learning in implicit generative models. In *arXiv preprint arXiv:1610.03483*.

Nathan Silberman, P. K., Derek Hoiem; and Fergus, R. 2012. Indoor segmentation and support inference from RGBD Images. In *ECCV*.

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

Rizzo, M. L.; and Székely, G. J. 2016. Energy distance. In *wiley interdisciplinary reviews: Computational statistics*, 27–38. Wiley Online Library.

Seitzer, M.; Schölkopf, B.; and Martius, G. 2021. Causal influence detection for improving efficiency in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34.

Seitzer, M.; Tavakoli, A.; Antic, D.; and Martius, G. 2022. On the Pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. In *International Conference on Learning Representations*.

Sejdinovic, D.; Sriperumbudur, B.; Gretton, A.; and Fukumizu, K. 2013. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. In *The Annals of Statistics*, 2263–2291.

Si, P.; Kuleshov, V.; and Bishop, A. 2022. Autoregressive quantile glows for predictive uncertainty estimation. In *International Conference on Learning Representations*.

Skafte, N.; Jørgensen, M.; and Hauberg, S. 2019. Reliable training and estimation of variance networks. In *Advances in Neural Information Processing Systems*, volume 32.

Stirn, A.; and Knowles, D. A. 2020. Variational variance: simple, reliable, calibrated heteroscedastic noise variance parameterization. In *arXiv preprint arXiv:2006.04910*.

Tagasovska, N.; and Lopez-Paz, D. 2019. Single-model uncertainties for deep learning. In *Advances in Neural Information Processing Systems*, volume 32.

Villani, C. 2009. *Optimal transport: old and new*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer.