FASTENSEMBLE: BENCHMARKING AND ACCELERATING ENSEMBLE-BASED UNCERTAINTY ESTIMATION FOR IMAGE-TO-IMAGE TRANSLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Estimating prediction uncertainty and confidence of deep learning models is crucial for mission-critical machine learning applications, such as biomedical imaging for diagnostics or therapy, and self-driving cars. However, making robust uncertainty estimation is complicated given the variety of learning objectives, data modalities, types of data corruption. Previous studies often addressed such a challenge by restricting datasets to standard ones like CIFAR or ImageNet. While convenient, it is doubtful whether the same conclusion holds for real-life datasets, in which more complicated image generation tasks are involved. This paper presents a different perspective to evaluate how confidence and uncertainty estimators behave under distribution shifts, focusing on the biomedical imaging domain. Specifically, we test a series of pair-wise cell imaging datasets using a new metric to compare existing models. In addition, we introduce FastEnsemble, a fast ensemble method which only requires less than 8% of the full-ensemble training time to generate a new ensemble member. Our experiments show that the proposed fast ensemble method is able to substantially improve the speed *vs* quality trade-off.

1 INTRODUCTION

As we deploy machine learning models to real-life production systems, an obstacle to many practitioners is to what extent we can trust the prediction results generated from millions or billions of parameters. To solve this problem, we need another layer of abstraction that takes in the model and data information and outputs the confidence interval (for regression task) or the expected error rate (for classification task). Ideally, such mechanism needs to: 1) handle both expected input (called *in-distribution* data) or unexpected input (called *out-of-distribution data*), 2) compared with the original model training time, operate efficiently enough, so that little overhead is posted, and 3) independent of modeling details, can work even for black-box models.

The algorithm that calibrates the model confidence to match the prediction accuracy is formally called *confidence calibration*. For instance, the original machine learning model may report 99.8%-confidence about its prediction, yet the actual accuracy is only 90% – far below the confidence. This disparity requires us to calibrate the confidence estimation from 99.8% to 90%. A related concept is called *uncertainty estimation*; it is meant to generate the confidence interval for model predictions, so we expect the true value falls into this interval with high probability. Predictive uncertainty also alarms the human-in-the-loop (HITL) machine learning paradigm, signaling that human intervention is needed once it raises above a threshold.

This work is motivated by the importance of uncertainty estimation in biomedical applications. In the past few years, many machine learning models have been deployed in biomedical imaging, such as the cell type identification (Christiansen et al., 2018), label-free organelle labeling (Ounkomol et al., 2018), histology virtual staining (Rivenson et al., 2019), and noninvasive cell phenotyping (Imboden et al., 2021). Among these applications, many involve image-to-image translation models. Unfortunately, uncertainty estimation in image translation has been under-explored – there exists no promising benchmarks nor systematical studies on how existing uncertainty estimation methods perform on image generation tasks. One of the main obstacles for this problem is the lack of benchmark and evaluation method – it is difficult to quantitatively evaluate uncertainty estimation

methods for image generation. To evaluate an uncertainty estimation method for classification, one can easily choose a leave-one-out set, usually a new class that is not appeared in training, and calculate the disagreement between uncertainty estimation and prediction error rate. However, this cannot be easily done in image-to-image translation. As the output space is high dimensional, uncertainty estimation cannot be easily calculated in a per-sample manner, and there could be nonuniform uncertainty for different patchs of the image. For example, it may be the case that in the same image, some cells have been seen in training but others are not, leading to nonuniform uncertainty within an image.

This paper develops the first systematic benchmark and evaluation method for uncertainty evaluation in image-to-image translation. To build this evaluation benchmark, we collect a series of phase contrast (transmitted light microscopy) and immunofluorescent images of mesenchyaml stromal cells Imboden et al. (2021) and prostatic cancer cells (LNCaP). With these microscopy data, we measure the quality of uncertainty estimation through out-of-distribution detection and distribution shift assessment. Equipped with this new benchmark, we evaluate six representative uncertainty estimation methods, including naive ensemble, snapshot ensemble, batch ensemble, SGLD, variational inference, and Monte-Carlo dropout. Our experimental results suggest that the naive ensemble consistently outperforms other more complicated algorithms on our biomedical image-to-image translation benchmark.

While the native ensemble approach provides an accurate estimate on the prediction uncertainty, a major weakness of such an approach is its computational overhead for building independent models. So it is critical to ask *can we find a near zero-cost method for uncertainty estimation?* To this end, we present our solution called FastEnsemble. This solution is inspired by the recent findings on the connectivity of local minimum of deep neural networks (Garipov et al., 2018). In particular, our goal is not to find the low-loss path connecting two distinct local minima but to find some independent low-loss paths starting from an initial solution. We search the path efficiently so that each path only takes 3% to 5% of the time to train one model from scratch. In total, gathering an ensemble of six models requires $\sim 20\%$ extra computation. Our contributions can be summarized as follows:

- We develop a new benchmark to evaluate uncertainty estimation algorithms on biomedical image generation applications. Based on that, we try to find the best solution out of six popular uncertainty estimators. Our paper is the first to study uncertainty quantification on image generation tasks systematically.
- We propose a new method that generates many independent ensemble models with a small overhead. Experimental results demonstrate that our approach can significantly speed up the running time without sacrificing the uncertainty estimation quality.

2 Related work

We have seen the active development of new efficient methods for confidence calibration and uncertainty quantification. Similar work can be roughly divided into two groups. The first group falls into the category of Bayesian learning. In this paper, we include the following approximate Bayesian methods:

Monte-Carlo dropout (MC-Dropout): Gal & Ghahramani (2016) showed that the dropout layers applied before every weight layer is mathematically equivalent to the deep Gaussian process. The most significant benefit of this solution is simplicity, meaning that the existing neural networks with dropout layer before (de)convolution layer or fully connected layer are naturally becoming a Bayesian neural network.

Stochastic gradient Langevin dynamics (SGLD): Welling & Teh (2011); Li et al. (2016) proposed a way to transform stochastic gradient optimizer to imitate the Langevin dynamics. Similar to MC-Dropout, this method does not change the architecture of neural networks as long as the stochastic gradient can be computed efficiently. Like SGD optimizer, a damping step size ϵ_t is required to guarantee that the injected Gaussian noise eventually dominates the stochastic gradient noise so that the parameter trajectory converges to the true posterior. In practice, we follow the previous implementations to turn off the noise injection at the burn-in phase, then turn on the noise injection in the sampling phase.

Stochastic variational inference (SVI) (Wainwright & Jordan, 2008; Blei et al., 2017): This method approximate the posterior by maximizing the ELBO. Unlike MC-Dropout and SGLD, to

apply SVI we need to double the number of parameters to learn both mean and standard deviation (assuming factorized Gaussian is used).

The other group we will include in the experiments is the ensemble methods. Specifically, the ones featuring low training overhead, detailed as follows

Snapshot ensemble (Huang et al., 2017a): It generates different model parameters with cyclic cosine learning rate, a checkpoint is stored whenever the learning rate drops to the minimum. Although there are other ensemble methods by the cyclic learning rate, such as the piece-wise linear rate in (Garipov et al., 2018), we only experiment with snapshot ensemble here for brevity.

Batch ensemble (Wen et al., 2020): The more recent advancement is batch ensemble. This method generates less correlated models by learning a series of rank-1 vectors $\boldsymbol{v}_i \in \mathbb{R}^d$ and $\boldsymbol{u}_i \in \mathbb{R}^d$, which are later element-wise multiplied by the weights $w_i \leftarrow w \odot (\boldsymbol{v}_i \boldsymbol{u}_i^\top)$.

Finally, we would like to address the differences between our work and Ovadia et al. (2019). In our work, we intend to dive deeper into the biomedical imaging domain (both image2image and image classification), where uncertainty estimation is critical and out-of-distribution data is abundant. In contrast, Ovadia et al. (2019) studies classification problems exclusively, including image classification, text classification and Ads-click binary classification problems. Most of the datasets studied here are originated from real applications. To our knowledge, this is the first systematical study concerning image2image. Our study is unique because in-distribution data and out-of-distribution data coexist in the same image, so the uncertainty values are directly comparable.

3 A New Uncertainty Estimation Benchmark for Image Generation Task

The predictive uncertainty originates from a lack of training data (namely *epistemic uncertainty*) or the inherent randomness in the data generation model (*aleatory uncertainty*) (Kendall & Gal, 2017; Hüllermeier & Waegeman, 2021; Abdar et al., 2021). In machine learning applications, uncertainty arises from unpredictable changes in the environment. For example, researchers may hand-pick the biomedical images in the training set uniformly, so low-quality images are cleaned up. At the same time, the model is deployed at hospitals owning different brands of microscopes in suboptimal working conditions, or there may be impurities of various shapes/components that are impossible to enumerate beforehand.

In this section, we introduce a new benchmark for evaluating uncertainty estimation methods on the image-to-image translation task. As image-to-image translation is crucial to many biomedical applications, our datasets primarily consist of microscopy cell images. Following (Ovadia et al., 2019), we investigate how different models behave under two Out of Distribution (OOD) settings: the first context is the local perturbation, meaning that the whole image is in-distribution except for some small patches. This scenario frequently happens in biomedical experiments, where impurities contaminate the cell culture. The other context is called global perturbation. The whole image distribution is drifted away from the original data generation distribution in the training set. It happens when the cells are cultured under different conditions (e.g., drug treatment, growth media changes, or image acquisition at different time points).

3.1 Out of distribution detection

The first benchmark is a collection of pairwise image to image translation tasks closely related to biomedical imaging. This work mainly used the same dataset that was tested and published in our previous study (Imboden etal). In addition, in the current study we acquired new LNCaP images for mimicking non-trivial image contaminations. To benchmark the out-of-distortion detection of our model, we tested three conditions: MSC clean (control), MSC-impurities (non-cellular objects), and MSC-LNCaP (cellular objects).

MSC-Clean (quality control): The dataset used for training purposes contains pairs of phase contrast and the respective fluorescence (IF) images of mesenchymal stromal cells (MSC). The cells were immunofluorescently stained for CD105, a surface marker, widely used to define MSC subpopulations. All images were acquired with an inverted microscope (Etaluma LS720, Lumaview

720/600-Series software) with a 20x phase contrast objective (Olympus, LCACHN 20XIPC). This is called a cleaned dataset as a quality control was performed where blurry or corrupt images were excluded.

MSC-Impurities: This dataset includes images of the same cell type (MSCs) and surface marker (CD105) as the cleaned dataset. To evaluate the impact of image impurities on the training accuracy, the MSC-Impurities dataset contains images of which 25% show artifacts. We included three different types of image artifacts: microscope slide impurities (e.g. scratches, bubbles, slide dust), fluorescent speckles and non-specific binding of the antibody.

MSC-LNCaP: This dataset is artificially created by mixing the images of MSC cells (majority) with LNCaP cells (read patch boxes in Figure 3). MSC cells and LNCaP cells are visually different, but for non-expert humans it is non-intuitive to tell them apart. So we expect this dataset to be much harder than **MSC-Impurities**.

Among those datasets, **MSC-Clean** is the one to train the U-Net (Ronneberger et al., 2015) model to be experimented later. At this moment, the model hasn't encountered the OOD patches in the subsequent two datasets. After that, we apply the model on **MSC-Impurities** and **MSC-LNCaP** to collect the uncertainty of each pixel. Finally, we examine whether the model assigns high uncertainties inside the bounding boxes and low uncertainties outside the bounding boxes. To this end, we encapsulate this problem by the ranking problem. Specifically, we leverage two commonly used metrics in information retrieval, $\mathrm{Precision@}k$ and $\mathrm{Recall@}k$, to compare different methods. Here we treat pixels inside bounding boxes as positive instances $\mathcal{S}_1$ (and vice versa); we then rank the pixels by the uncertainty values in descending order. The top-k highest uncertainty pixels $\mathcal{S}_2$ are selected. Then we have

$$\mathrm{TP@}k = |\mathcal{S}_1 \cap \mathcal{S}_2|, \quad \mathrm{Precision@}k = \frac{\mathrm{TP@}k}{k}, \quad \mathrm{Recall@}k = \frac{\mathrm{TP@}k}{|\mathcal{S}_1|}, \tag{1}$$

here TP means number of true positives. We illustrate this idea in Figure 1.

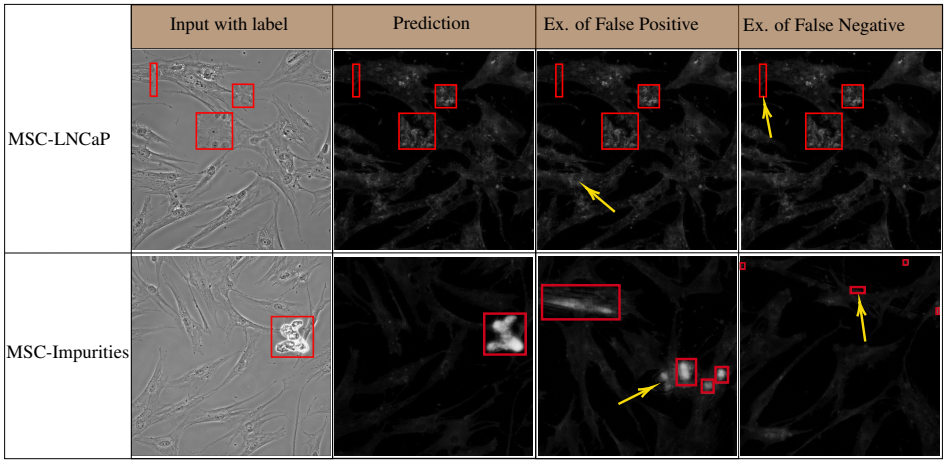

Figure 1: Image samples from the **MSC-LNCaP** and **MSC-Impurities** datasets and the corresponding uncertainty estimation generated by the ensemble method. The first row highlights bounding boxes (drawn to highlight the ground truth inaccessible to models) in an image from **MSC-LNCaP**. It is difficult even for humans to notice the out-of-distribution LNCaP cells surrounded by MSC cells without expertise. The second row is generated from **MSC-Impurities** data. This is an easier task because impurities usually are easily distinguishable from the cells.

Moreover, by changing k, we can plot the ROC curve to compare different methods visually. The experimental results are displayed in Figure 2. In this comparison, we include six popular uncertainty estimation methods, including the naive ensemble, snapshot ensemble, batch ensemble, SGLD, SVI, and MC-Dropout. Details of these methods can be found in related work.

From this figure, we can observe that the naive ensemble method outperforms all other methods, sometimes with a significant margin (**MSC-LNCaP**). We remark that this finding supports a sim-

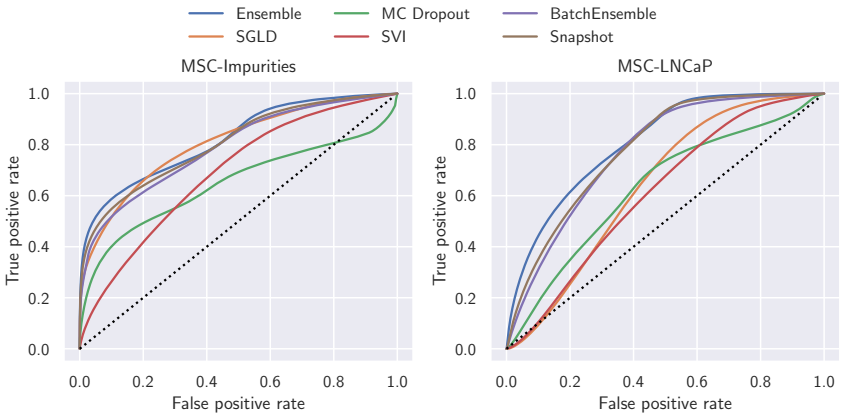

Figure 2: Comparison of some widely used ensemble methods and Bayesian inference algorithms. Notice the naive ensemble method performs similarly to batch ensemble or SGLD in **MSC-Impurities** data and significantly better in **MSC-LNCaP** data. In practice, we want to control the false positive rate to a small value, so we mainly look at the AUC when false positive ≤ 0.2.

ilar conclusion in Ovadia et al. (2019), where the authors found that the simple ensemble method outperforms other Bayesian methods in image recognition datasets. Our experiment further indicates that existing fast ensemble methods (BatchEnsemble, Snapshot Ensemble) cannot close the gap concerning OOD robustness.

3.2 Distribution shift assessment

In this experiment, we show that ensemble method is more robust even under large perturbations. Previous benchmark datasets are mostly in-distribution except for small patches labeled by bounding boxes, a more challenging case where the testing samples are different from the whole training set remains to be investigated. To evaluate different algorithms in this condition, we introduce a new dataset called **LNCaP-Density**.

LNCaP-Density: In contrast to **MSC-Clean** and **MSC-Impurities**, the images used for this dataset are of an LNCaP cell type. LNCaP cells are androgen-sensitive human prostate adenocarcinoma cells. In this dataset, time-lapse phase contrast images of 12 different fields of view (FOVs) were acquired over a period of 72 hours. Cell density increases significantly over the time period due to cell division and growth. We did some manual sorting work to distribute all images into four subsets: namely VSparse ("very sparse"), Sparse, Dense, and VDense. Figure 3 gives some samples in each groups.

We train two models: model A is trained using the most sparsely populated cells (VSparse), and model B is trained with the most densely populated cells (VDense). After completing training, we run the predictions on all groups (VSparse, Sparse, Dense, VDense). The Pearson correlation between prediction and ground truth is calculated as the metric. We plot the histogram in Figure 4.

Similar to the previous local perturbation benchmark, from Figure 4, we can see naive ensemble is still the best performing method in nearly every case. But the gap between batch ensemble / snapshot ensemble is small. Moreover, we generally find the ensemble-based methods more stable than Bayesian methods by comparing the error bar length. This finding aligns well again with Ovadia et al. (2019).

4 Accelerating Ensemble Method

In the previous section, we tested three Bayesian methods and three ensemble methods on two OOD benchmarks. Our investigation reveals that the most robust uncertainty estimator is the naive ensemble aggregation, despite the Bayesian methods being more theoretically principled.
include a problem statement We hypothesize that the power of the Bayesian method is restricted

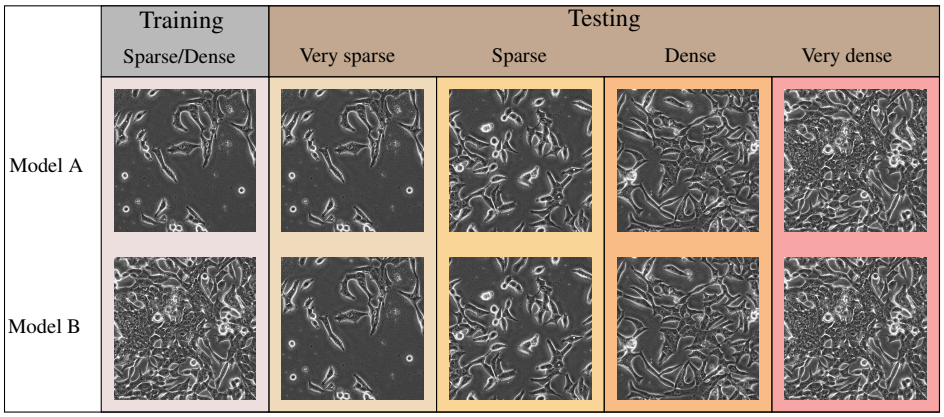

Figure 3: Samples from the **LNCaP-Density** dataset and illustration of the distribution shift experiment. Model A is trained with the "very sparse" subset of **LNCaP-Density**, and Model B is trained with the "very dense" subset. Both models are then tested with all subsets of varying densities.

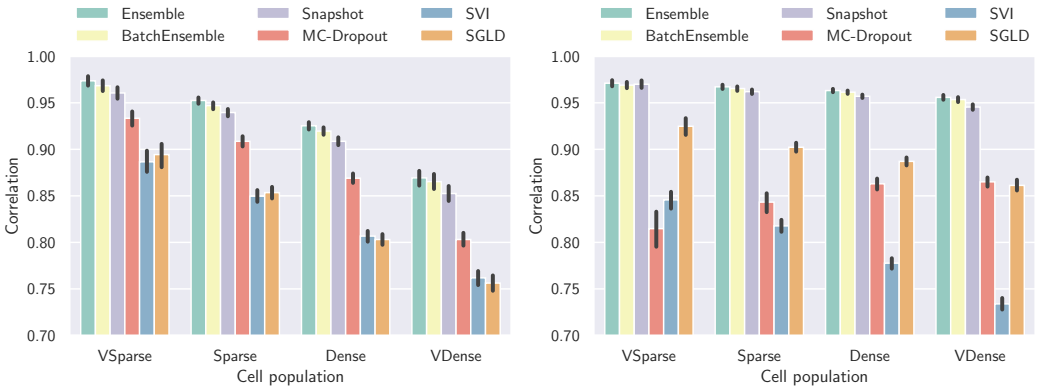

Figure 4: Comparing our method with other ensemble or Bayesian methods under a distribution shift setting. *Left*: Training on the "very sparse" subset and evaluation on each subset (Model A of Fig. 3). *Right*: Training on the "very dense" subset (Model B) and evaluation on each subset. The error bar is computed over all images. We can see the correlation drops more quickly for the "very sparse" training set (*Left*); this is because the "very sparse" subset contains mostly dark backgrounds and so less meaningful information can be extracted.

by choice of prior distributions and approximate inference. On the other hand, the training cost of the naive ensemble method makes the deployment prohibitive to large-scale databases. Training an ensemble of K models will increase the computational cost by K times. As we have seen in the previous experiments, current fast ensemble methods are not meant for robust uncertainty estimation.

In the following sections, we introduce a simple yet effective ensemble method called FastEnsemble. Our approach is inspired by the recent findings mode connectivity of local minimum Garipov et al. (2018): we first find a seed model w_0, then explore along the "loss valley" by adding a bias term $\|w - w_0\|_1$ to the classification or regression loss. On convergence, we expect the new model w' to be as good as w_0, but show enough independence. Our idea contrasts to snapshot ensemble or batch ensemble, where the former is controlled by a cyclically climbing up and decaying learning rate. The latter takes no direct measure to achieve this.

4.1 FastEnsemble Algorithm

Denote the loss function of data pair (x_i, y_i) as ℓ; $f(\cdot; w)$ is the neural network parameterized by w. Our algorithm has two stages: in the initial stage, we train a "seed model" to convergence following

the usual routine, the model is denoted as w_0. Then in the next stage, we augment the loss function ℓ by a series of ℓ_1 distances defined over model set $\mathcal{M}$.

$$\ell^+(w) = \ell\Big(f\big(x_i; w\big), y_i\Big) - \frac{\lambda}{|\mathcal{M}|} \cdot \sum_{w_{\text{anchor}} \in \mathcal{M}} \|w - w_{\text{anchor}}\|_1 \,. \tag{2}$$

Previous finding (Garipov et al., 2018) suggests that the low loss area (Figure 5) is connected. Once we train the seed model to a low loss, we can generate many good and independent models by simultaneously minimizing the training loss and maximizing the distance between the new model and all existing ones in $\mathcal{M}$. The algorithm in pseudo-code is shown in Algorithm 1.

Initialize: N: number of ensemble models parameterized by w_i; $\ell(\hat{y}, y)$: the loss function; λ: the hyperparameter to be tuned. $k_1 \gg k_2, k_3$: number of iterations for training seeding model, training sub-models and fintuning sub-models.
▷ *Train the seeding model* ◁
for all $i \in \{0 \dots k_1 - 1\}$ **do**
    Run one step of optimizer and learning rate scheduler.
Initial model list $\mathcal{M} = \{w_0\}$.
▷ *Train the rest $N-1$ models* ◁
for all $n \in \{1 \dots N - 1\}$ **do**
    for all $i \in \{0 \dots k_2 - 1\}$ **do**
        ▷ *Quick training* ◁
        Minimize $\ell^+(w)$ in Eq. (2).
    for all $i \in \{0 \dots k_3 - 1\}$ **do**
        Minimize $\ell(\hat{y}, y)$. ▷ *Finetuning*
    Append to model list $\mathcal{M} = \mathcal{M} + w_n$.

Algorithm 1: Algorithm of FastEnsemble

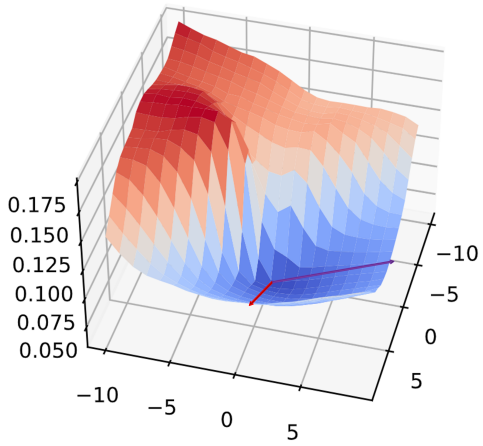

Figure 5: Loss landscape around a local minimum. There are multiple directions (in red arrows) we can choose to escape the local minimum while staying in the low loss "valley".

Notice in this algorithm, we choose the number of iterations $k_1 \gg k_2, k_3$ so that compared to the one-time seed model training, the rest $N-1$ ensemble members only takes $\frac{k_2+k3}{k_1} \approx 3{\sim}8\%$ overhead. That makes our new training overhead considerably cheaper than in snapshot ensemble. Our algorithm introduces a hyperparameter λ, which controls the trade-off between model accuracy and inter-model independence. In other words, a larger λ causes a lower model correlation (due to longer distances between $\mathcal{M}$), but the individual model performs worse than before.

Dataset	Naive	MC-Dropout	SGLD	SVI	BatchEnsemble	Snapshot	FastEnsemble
	Measured by AUC (controlling FPR$\leq$ 0.2)						
MSC-Impurities	**0.112**	0.074	0.099	0.050	0.098	0.002	0.108
MSC-LNCaP	0.082	0.035	0.021	0.023	0.059	0.001	**0.090**
	LNCaP-Desity(Model A), measured by mean Pearson correlation						
Very dense	**0.869**	0.803	0.756	0.762	0.865	0.853	0.865
Dense	**0.925**	0.869	0.803	0.807	0.919	0.909	0.923
Sparse	**0.952**	0.909	0.853	0.849	0.947	0.939	0.950
Very sparse	**0.974**	0.933	0.894	0.887	0.968	0.960	0.971
	LNCaP-Density(Model B), measured by mean Pearson correlation						
Very dense	**0.869**	0.803	0.756	0.762	0.865	0.852	0.865
Dense	**0.925**	0.869	0.803	0.807	0.919	0.908	0.922
Sparse	**0.952**	0.909	0.853	0.850	0.947	0.939	0.950
Very sparse	**0.973**	0.933	0.894	0.887	0.968	0.960	0.971

Table 1: Revisiting experiments in Section 3 with our proposed FastEnsemble. For clarity, the first place is marked in **bold font**, the second place is in red, the third place is in blue.

We repeat all the experiments in Section 3 again with our proposed method, then make some comparisons in AUROC or Pearson correlation measures. The results are displayed in Table 1. The naive

ensemble method is still better than the others except MSC-LNCaP dataset; this indicates that current fast ensemble models are still sacrificing accuracy for the speed. Among all efficient methods, our FastEnsemblesurpasses all others in the MSC-LNCaP dataset and ranked second on all other datasets.

5 CLASSIFICATION BENCHMARK AND CALIBRATION ROBUSTNESS

In this section, we intend to show that the proposed FastEnsemble method can also work on regular classification tasks. In particular, we first run on CIFAR10 and CIFAR100 as two standard datasets, then we move to out-of-distribution robustness on CIFAR10-C (Hendrycks & Dietterich, 2019) and CIFAR100-C. Finally, we focus on biomedical imaging datasets, Camelyon17 (Bandi et al., 2018) and RxRx1 (Taylor et al., 2019), as two larger-scale real applications.

We measure three things: accuracy, log-likelihood, and confidence calibration. Confidence is defined as the probability in the model output (values after sigmoid or softmax function). As the size of the deep learning model grows, the model can easily fit the training set to a low NLL loss by generating probabilities closer to one-hot distribution, which implicitly hurts the confidence estimation (Guo et al., 2017). We quantify the miscalibration level by the expected calibration error (ECE) (Naeini et al., 2015):

$$\text{ECE} = \int_0^1 w(p) \cdot \big|\text{Acc}(p) - p\big| \, \mathrm{d}p. \tag{3}$$

In this equation p is the confidence output from Softmax; $w(p)$ is percent of data having confidence p ; $\text{Acc}(p)$ is the accuracy as a function of confidence. In practice, the integration (3) is computed by confidence binning $\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \big|\text{Acc}(B_m) - \text{Conf}(B_m)\big|$, in which $B_m = (^{(m-1)}/_M, ^m/_M]$ is the m-th bin between $[0, 1]$.

For the network architecture and training configurations, we mostly follow the previous literature. Specifically:

- CIFAR: This configuration applies to all CIFAR based datasets. We train with AdamW optimizer for 200 epochs, batch size is 128. We adopt the linear learning rate scheduler, the initial learning rate is 1.0×10^{-3}.
- Camelyon17: This is a collection of tissue slides under microscopy, in which training and testing distributions differ due to patient population or in slide staining and image acquisition. We follow the configuration in WILDS benchmark (Koh et al., 2021). The model architecture is a ImageNet-1k pretrained DenseNet121 (Huang et al., 2017b), finetuned with momentum SGD and batch size $= 32$.
- RxRx1: Similar to Camelyon17, there is a distribution shift due to the batch effect. We choose ImageNet-1k pretrained ResNet50 (He et al., 2016) to initialize the model, finetuned with Adam and batch size $= 72$.

More experiment details can be found in Appendix. First, we explore the accuracy and ECE under distribution shift. The results can be found in Figure 6.

The figure shows that the naive ensemble method is still the best choice considering the best accuracy and calibration in all cases. But our approach is on par with the naive ensemble; both are significantly better than batch ensemble and snapshot ensemble. Next, we repeat the same routine to all six datasets to compare accuracy, log-likelihood, as well as ECE. We repeated the experiments three times by changing random seeds. Finally, we report the mean measures and standard deviations. From Table 2, we can conclude that our method is the closest to naive ensemble in terms of accuracy, and often has the lowest calibration error on the datasets we tested.

6 CONCLUSION

In this paper, we consider the problem of robust uncertainty estimation and calibration under various distribution shifts. Our focus is on the applications in biomedical imaging, where the batch effect (e.g., cell-cell phenotype variation, batch-to-batch inconsistency, imaging condition differences) is a dominating reason behind the training-testing mismatch. Since this application is vital in healthcare and fundamental research, it is essential to create an efficient method to estimate how reliable

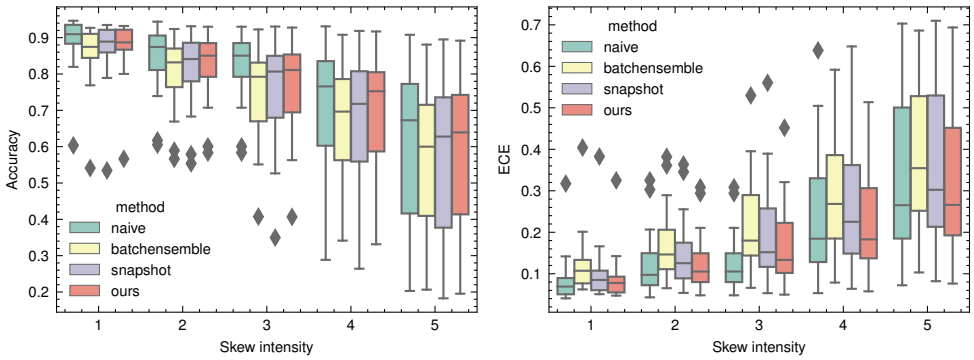

Figure 6: CIFAR10-C: accuracy and ECE (the lower the better) degrade as image skewness intensifies. The box plot is made by aggregating the measurements over 15 kinds of corruptions made by Hendrycks & Dietterich (2019).

Single			Naive			Batch			Snapshot			Ours		
ACC	NLL	ECE	ACC	NLL	ECE	ACC	NLL	ECE	ACC	NLL	ECE	ACC	NLL	ECE
CIFAR10+VGG16:														
92.89	0.492	0.058	94.64	0.306	0.041	92.79	0.566	0.061	93.62	0.375	0.049	93.24	0.308	0.047
0.10	0.031	0.002	0.06	0.018	0.001	0.08	0.019	0.001	0.21	0.029	0.001	0.19	0.015	0.002
CIFAR100+VGG16:														
68.65	2.496	0.236	75.15	1.516	0.173	68.44	2.954	0.252	70.52	1.775	0.198	71.14	1.326	0.149
0.10	0.137	0.004	0.07	0.105	0.003	0.16	0.074	0.003	0.39	0.126	0.015	0.29	0.103	0.012
Camelyon17+DenseNet121:														
84.99	0.397	0.083	85.96	0.347	0.066	84.39	0.399	0.081	Failure			87.71	0.305	0.048
1.06	0.038	0.012	0.04	0.004	0.003		—					0.33	0.016	0.009
RxRx1+ResNet50:														
25.82	7.908	0.469	34.80	5.638	0.370	30.31	7.409	0.450	19.36	5.556	0.265	31.08	6.490	0.407
0.27	0.025	0.002	0.07	0.195	0.012	0.51	0.272	0.012	0.06	0.167	0.019	0.39	0.091	0.003
Test-only datasets using models trained from CIFAR10 and CIFAR100														
CIFAR10-C+VGG16:														
86.84	0.980	0.110	89.21	0.671	0.084	85.91	1.180	0.121	87.00	0.810	0.102	87.38	0.623	0.089
0.56	0.105	0.006	0.27	0.052	0.003	0.13	0.018	0.001	0.21	0.061	0.003	0.31	0.312	0.002
CIFAR100-C+VGG16:														
55.81	4.117	0.335	63.45	2.670	0.256	55.15	5.054	0.359	58.36	3.035	0.282	58.96	2.249	0.220
0.28	0.286	0.006	0.36	0.246	0.009	0.14	0.100	0.003	0.35	0.224	0.019	0.31	0.139	0.007

Table 2: Experiment results on distribution shifted or clean datasets. Mean values are in normal font. Standard deviations are computed over three independent runs, and we display them in gray color. The metrics are NLL/ACC/ECE. Notice that CIFAR10-C and CIFAR100-C are test-only datasets; we evaluate them using the same model checkpoint acquired from CIFAR10 and CIFAR100. In Camelyon17+DenseNet121 combination, we found the snapshot ensemble method failed to converge in all three trials. The reason is that when the learning rate spikes at the beginning of the second cycle, the optimizer makes an unnecessarily big step to drive the model out of the low loss area.

the machine learning predictions are. Our general conclusion aligns well with previous findings revealing that naive ensemble performs better in most cases for both image-to-image translation and classification tasks. A specific contribution of this work is the presented large-scale, systematic studies in the experimental biology imaging domain. Beyond the calibration error that has been extensively studied in the classification task, this work presents a comprehensive benchmarking results of the uncertainty estimation in image generation tasks. More importantly, we proposed a fast ensemble method that provides uncertainty assessments comparable to those of naive ensemble, but with substantially reduced training overhead.

ETHICS STATEMENT

The microscopy images analyzed in this work were obtained using commercially available cells. The experimental procedure was conducted following the ethic guidelines in the experimental biology field. Since no human or animal subjects were involved in this study, no special protocols or ethic approval were required.

REPRODUCIBILITY STATEMENT

The experimental settings to help reproduce the results are discussed in the beginning of each subsection. More details, such as network architecture, hyper-parameters, training conditions, etc. are listed in Appendices. Source code and datasets will be publicly available soon.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 2021.

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, et al. From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. *IEEE transactions on medical imaging*, 38(2):550–560, 2018.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Eric M Christiansen, Samuel J Yang, D Michael Ando, Ashkan Javaherian, Gaia Skibinski, Scott Lipnick, Elliot Mount, Alison O'Neil, Kevan Shah, Alicia K Lee, et al. In silico labeling: predicting fluorescent labels in unlabeled images. *Cell*, 173(3):792–803, 2018.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8803–8812, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017a.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017b.

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

Sara Imboden, Xuanqing Liu, Brandon S Lee, Marie C Payne, Cho-Jui Hsieh, and Neil YC Lin. Investigating heterogeneities of live mesenchymal stromal cells using ai-based label-free imaging. *Scientific Reports*, 11(1):1–11, 2021.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *arXiv preprint arXiv:1703.04977*, 2017.

Pang Wei Koh, Shiori Sagawa, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pp. 5637–5664. PMLR, 2021.

Chunyuan Li, Changyou Chen, David Carlson, and Lawrence Carin. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Chawin Ounkomol, Sharmishtaa Seshamani, Mary M Maleckar, Forrest Collman, and Gregory R Johnson. Label-free prediction of three-dimensional fluorescence images from transmitted-light microscopy. *Nature methods*, 15(11):917–920, 2018.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *arXiv preprint arXiv:1906.02530*, 2019.

Yair Rivenson, Hongda Wang, Zhensong Wei, Kevin de Haan, Yibo Zhang, Yichen Wu, Harun Günaydın, Jonathan E Zuckerman, Thomas Chong, Anthony E Sisk, et al. Virtual histological staining of unlabelled tissue-autofluorescence images via deep learning. *Nature biomedical engineering*, 3(6):466–477, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

J. Taylor, B. Earnshaw, B. Mabey, M. Victors, and J. Yosinski. Rxrx1: An image set for cellular morphological variation across many experimental batches. In *International Conference on Learning Representations (ICLR)*, 2019.

Martin J Wainwright and Michael I Jordan. Introduction to variational methods for graphical models. *Foundations and Trends in Machine Learning*, 1:1–103, 2008.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688. Citeseer, 2011.

Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. *arXiv preprint arXiv:2002.06715*, 2020.

A Dataset information

MSC-Clean : This dataset contains 595 training images and 77 testing images (not used in this paper). Images are stored in three channel png format with 1024x1024 pixels.

MSC-Impurities : This dataset contains 491 images (all of them are used for testing). Images are stored in three channel png format with 1024x1024 pixels. Each image file contains $1\sim3$ bounding boxes.

MSC-LNCaP : This dataset contains 77 images (all for testing). Images are stored in three channel png format with 1024x1024 pixels. Although this data is considerably smaller than MSC-Impurities, it contains more bounding boxes, typically $4\sim8$. We include some samples for each data in Figure 7 and Figure 8.

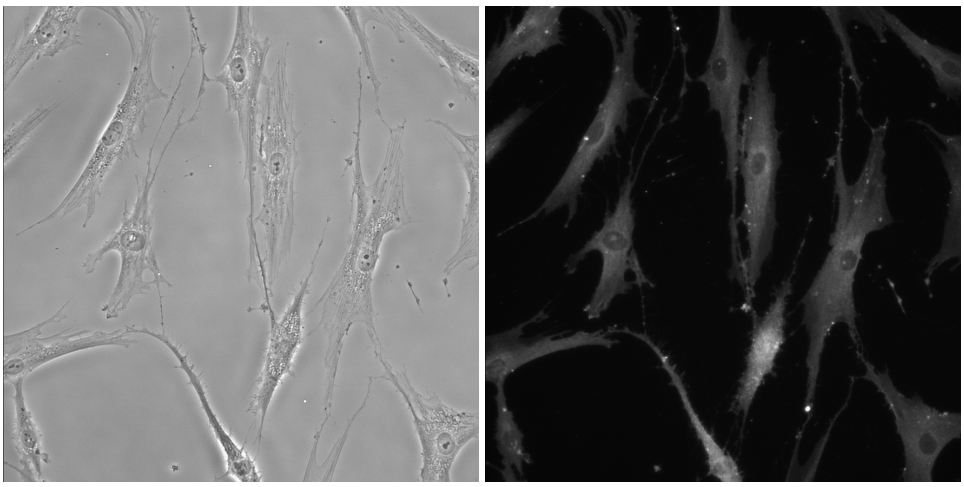

Figure 7: Source and target images of **MSC-Clean** paired dataset.

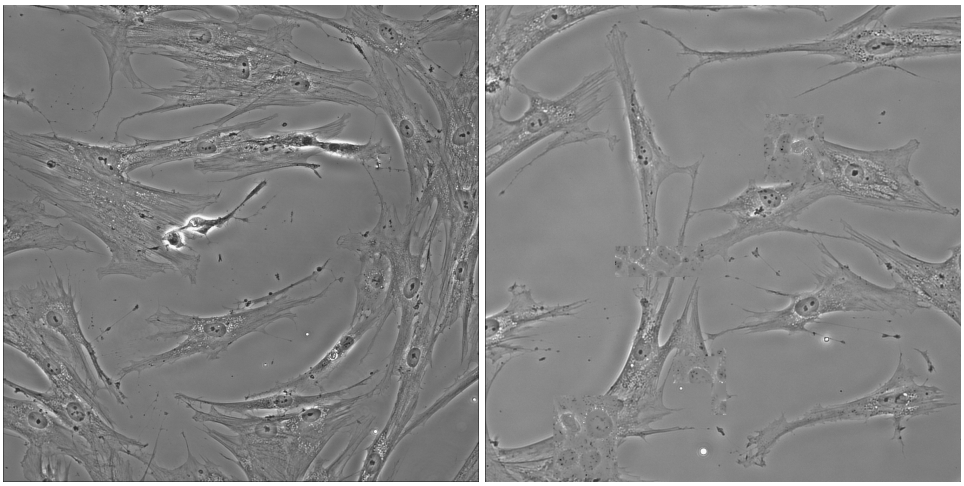

Figure 8: Left: sample from **MSC-Imurities** dataset, we can observe the impurity area near the center of image. Right: sample from **MSC-LNCaP** dataset. Although not very visible, there is a small patch in the bottom left (where the LNCaP cells are near-round, but normal MSC cells are slim).

Finally, we have the LNCaP-density dataset, it consists of four subsets: Very sparse (41 images), Sparse (181 images), Dense (275 images), Very dense (180 images). The samples are shown in Figure 3.

B NETWORK ARCHITECTURES AND HYPER-PARAMETERS

Our paper doesn't feature network architecture innovations. All network architectures are publicly available from websites.

Image to image translation tasks. We use U-Net (Ronneberger et al., 2015) publicly available at `https://github.com/phillipi/pix2pix`. Here we choose *unet-256* configuration, with channel multiplier ngf $= 64$ and batch normalization. Dropout is disabled except for MC-Dropout method.

CIFAR10/CIFAR100/CIFAR10-C/CIFAR100-C. We choose the standard VGG-16 architecture publicly available at `https://github.com/kuangliu/pytorch-cifar`. For CIFAR100/CIFAR100-C, we increase the last fc-layer to $d_{out} = 100$. We train the network for 200 epochs, using AdamW optimizer and learning rate 1.0×10^{-3}. Momentum is set to $\beta_1 = 0.5$, $\beta_2 = 0.999$.

Camelyon17/RxRx1 : We download the data with scripts from `https://github.com/p-lambda/wilds`. We strictly follows the official training scripts and hyperaprameters, which can be found here: `https://github.com/p-lambda/wilds/blob/main/examples/configs/datasets.py`.

Next, we release the training protocols of ensemble methods and Bayesian methods.

Naive ensemble We train six models independently with different random seeds. The prediction results are generated by a simple average. The total computational budget is $6B$. Where B is the budget to train one model from scratch.

Our method We first train a standard checkpoint with budget B, then use $\frac{k_2+k_3}{k_1} B \times 5$ to get the rest 5 models. In total, it costs $\frac{k_1+5(k_2+k_3))}{k_1} B$. Most typical choices are $k_1 = 200$, $k_2 = k_3 = 6$.

BatchEnsemble We replicate the batch ensemble code from official repository at `https://github.com/google/edward2/blob/main/edward2/tensorflow/layers/convolutional.py#L560`, and extend it to support ConvTranspose2d layer. We match the training budget of our method by increasing the training time proportionally.

MC-Dropout We use the dropout rate equaling to $p = 0.5$. The computational budget is B.

SGLD We first train the model until convergence (burn-in phase), at this stage, we don't inject Gaussian noise. At inference time, we train the model for one epoch after each sampling, the learning rate is 1000x smaller than the training stage. No preconditioning technique is applied. We remark that although training budget is only B, the inference budget is much higher than other methods.

SVI We copied the implementation of MFVI from pyro (`https://pyro.ai/examples/svi_part_i.html`) and a 3rd-party implementation `https://github.com/kumar-shridhar/PyTorch-BayesianCNN`. The prior follows iid $\mathcal{N}(0, 0.02)$. For fair comparison, we increase the training time to match our method.