Bayesian Low-Rank Learning (Bella): A Practical Approach to Bayesian Neural Networks

Bao Gia Doan[*1], **Afshar Shamsi**[*2], **Xiao-Yu Guo**[1], **Arash Mohammadi**[2], **Hamid Alinejad-Rokny**[4]
Dino Sejdinovic[1], **Damien Teney**[3], **Damith C. Ranasinghe**[1], **Ehsan Abbasnejad**[1]

[1]The University of Adelaide
[2]Concordia University
[3]Idiap Research Institute
[4]University of New South Wales

{giabao.doan,xiaoyu.guo,dino.sejdinovic,damith.ranasinghe,ehsan.abbasnejad}@adelaide.edu.au; damien.teney@idiap.ch;
h.alinejad@unsw.edu.au; {afshar.shamsi,arash.mohammadi}@concordia.ca

Abstract

Computational complexity of Bayesian learning is impeding its adoption in practical, large-scale tasks, despite demonstrations of significant merits such as improved robustness and resilience to unseen or out-of-distribution inputs over their non-Bayesian counterparts. Although, Deep ensemble methods have proven to be highly effective for Bayesian deep learning, their practical application is hindered by substantial computational cost. In this study, we introduce an innovative framework to mitigate the computational burden of ensemble Bayesian deep learning. We explore a more feasible alternative, inspired by the recent success of low-rank adapters, we introduce Bayesian Low-Rank LeArning (Bella). We show, i) Bella achieves a dramatic reduction in the number of trainable parameters required to approximate a Bayesian posterior; and ii) it not only maintains, but in some instances, surpasses the performance–in accuracy and out-of-distribution generalisation–of conventional Bayesian learning methods and non-Bayesian baselines. Our extensive empirical evaluation in large-scale tasks such as `ImageNet`, `CAMELYON17`, `DomainNet`, `VQA` with CLIP, LLaVA demonstrate the effectiveness and versatility of Bella in building highly scalable and practical Bayesian deep models for real-world applications.

Code — https://bnn-bella.github.io/BNN-Bella/

1 Introduction

Bayesian deep learning (Neal 2012) provides mechanisms, for building predictive models more *robust* to adversarial attacks, resilient to unseen or out-of-distribution data and a *theoretical* framework for estimating model uncertainty (Ye and Zhu 2018; Liu et al. 2019; Izmailov et al. 2020; Chen and Ghattas 2020; Wilson et al. 2022; Doan et al. 2023, 2024). Consequently, embracing Bayesian deep learning (BDL) represents a significant stride towards building more reliable and trustworthy AI systems for various real-world applications (e.g., autonomous driving, medical image analysis, etc.). *Unfortunately, their practical use is encumbered by computational complexity*.

[*]These two authors contributed equally to this work.

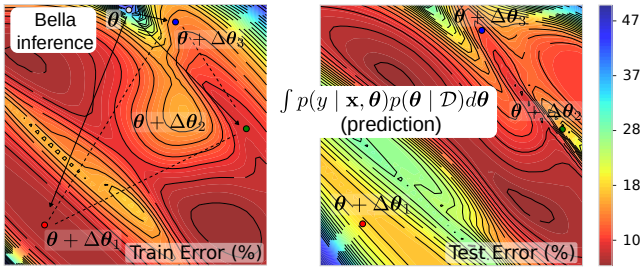

Figure 1: Train and Test Error (%) landscapes of `CAMELYON17`. Training landscape demonstrates the learned 3 particle approximation of the modes of the posterior from a pre-trained model $\boldsymbol{\theta}$—modification of parameters in the constrained region ($\Delta\boldsymbol{\theta}$) leads to approaching posterior modes in inference. Here, we observe a key benefit of our Bella approximation compared to a point estimate—while a single parameter particle (e.g. $\boldsymbol{\theta} + \Delta\boldsymbol{\theta}_1$) do not generalize well, Bayesian prediction with Equation (1), effectively an average over multiple parameter settings, leads to better performance.

Unlike traditional alternatives with point estimates—a single set of model parameters mapping inputs to outputs—Bayesian models learn the distribution of model parameters to offer a distribution over possible predictions. Consider a neural network $f(\mathbf{x}, \boldsymbol{\theta})$ with input $\mathbf{x}$ parameterized by $\boldsymbol{\theta}$ and its corresponding prior distribution $p(\boldsymbol{\theta})$. The likelihood $p(\mathcal{D}|\boldsymbol{\theta})$ is determined by f (Bishop 2006) and then, Bayesian inference seeks to use Bayes' theorem to derive a *posterior* distribution given by $p\left(\boldsymbol{\theta}|\mathcal{D}\right) = p\left(\mathcal{D}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right)/p(\mathcal{D})$ to compute the predictive distribution:

$$p(y|\mathbf{x}, \mathcal{D}) = \int p(y \mid \mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D}) d\boldsymbol{\theta}. \tag{1}$$

Despite evidence to support the adoption of Bayesian learning, their widespread use is hindered by several challenges; *a primary issue is the intractability of the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ in practical learning tasks*. The exact solution for the posterior, even for networks of moderate size, is impractical. Because of deep neural network's complexity and the high-dimensional integral of the resulting denominator. This necessitates the use of approximations.

Our work expands upon recent research demonstrating the effectiveness of ensemble methods and Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016) approaches in BDL, for example, (Seligmann et al. 2024; Lakshminarayanan, Pritzel, and Blundell 2017; Abbasnejad et al. 2020a; Doan et al. 2022; Izmailov et al. 2020; Chen and Ghattas 2020), to *provide a practical method for ensemble Bayesian deep learning for large-scale tasks*. To achieve a practical and efficient approach, we consider spawning parameter particles (models) by adding linear interpolation to a pre-trained model's parameters to build an ensemble. To further enhance the efficiency of these interpolations, inspired by the success of low-rank adapters (LORAs)(Hu et al. 2022), we propose using low-rank parameters in the interpolation process, we refer to as Bayesian Low-Rank Learning (Bella).

To validate this, we compare Bella to the full-parameter alternative in terms of (1) accuracy, (2) out-of-distribution generalization, (3) adversarial robustness, and (4) alignment of uncertainty estimation with human confidence. In summary, our findings indicate that this approach achieves comparable performance to full SVGD or ensemble methods at a fraction of the computational cost, with the exception of adversarial robustness. For example, Figure 1 illustrates the application of our proposed approach of Bayesian inference in an application with distribution shifts (`CAMELYON17` (Koh et al. 2021) benchmark). The particles (*i.e.*, parameter samples from the posterior) obtained using Bella prove effective in improving generalization, improving the common single particle fine-tuning and comparable with full ensemble-based baselines at a fraction of the cost. Our key contributions are:

- We propose a new Bayesian learning framework, Bella, for SVGD approximation of a posterior—exploiting the availability of pre-trained models, we spawn particles or models by linear interpolation of a constrained set of model parameters for a SVGD approximation of a posterior.
- Our approach more efficiently captures the complexity and multi-modality of the solution space compared to current SVGD but at a fraction of the cost—we observe on-par performance with full SVGD in uncertainty estimation, performance improvement, and robustness but only use less than 0.3% of parameters.
- We demonstrate Bella performs on par or better than baselines—ensembles or current SVGD implementations. Bella consistently outperforms the non-BNN counterparts on 8 datasets; image classification (`ImageNet`, `CIFAR 10/100`), Out-of-distribution (`DomainNet`, `Camleyon17`, `CIFAR-10-C`), and `VQA` as well as *adversarial robustness*. Notably, it sets a new state-of-the-art for the `CAMELYON17` (see Appendix for more details) according to the leaderboard (Challenge 2024).
- Bella employed with the multi-modal model LlaVa (Liu et al. 2023b), leads to improved performance and uncertainty estimation highly correlated with human confidence.

2 Related Work

Parameter-Efficient Fine Tuning. In contrast to fine-tuning all parameters, recent research proposed inserting *adapters* in between existing neural layers to reduce the number of trainable parameters and, subsequently, the compute time (GPU consumption) (Houlsby et al. 2019; Rebuffi, Bilen, and Vedaldi 2017; Lin, Madotto, and Fung 2020). Hu et al. (Hu et al. 2022) use a bottleneck structure to impose a low-rank constraint on the weight updates, named LoRA. The key functional difference is that LoRA can be merged with the main weights during inference, thus avoiding the introduction of any latency whilst significantly reducing the number of parameters.

Fine-Tuning Approaches and Bayesian Deep Learning. Previous research investigating the application of fine-tuning approaches for BDLs have predominantly focused on large language models (LLMs), e.g. (Fan et al. 2020; Zhang et al. 2021; Yang et al. 2024). Notably, marking a departure from the conventional methods relying primarily on tuning the network's parameters, these studies chose to define priors and approximate posterior over low rank *attention weights*. Concurrently, Yang et al. (Yang et al. 2024) introduced the concept of Laplace LoRA to incorporate Bayesian concepts to enhance the calibration of fine-tuned LLMs. However, Laplace's method (Daxberger et al. 2021) relies on a Gaussian approximation of the posterior distribution. Whilst this can be effective for unimodal and symmetric distributions, similar to Dusenberry et al. (2020); Krueger et al. (2017); Vadera et al. (2022), the approach does not fully encapsulate the intricacies of more complex posteriors, particularly in neural networks where the posterior has multimodality and asymmetry (Izmailov et al. 2021). Deep ensembles (Lakshminarayanan, Pritzel, and Blundell 2017) typically perform better in practice compared with variational and Laplace methods, due to their ability to capture multiple modes. When employing fine-tuning, a direct application of ensembling for LoRAs was considered in (Wang, Aitchison, and Rudolph 2024). Some interpretations of deep ensembles suggest that they approximate gradient flows in function spaces and that building desirable properties into an ensemble (such as repulsive behavior), is possible (Wild et al. 2023). SVGD (Liu and Wang 2016), can be viewed in a similar vein. However, while these more sophisticated, repulsive, ensembling approaches are highly impractical in the pre-training phase, we argue that their expressivity can be brought to bear precisely in tandem with low-rank fine-tuning, which is the viewpoint we adopt in this contribution.

Building upon these foundations, our research represents a pioneering effort to apply the principles of repulsive ensemble-based low-rank fine-tuning to computer vision. In particular, we bridge pre-training and fine-tuning phases with recent conjectures on mode connectivity (Ainsworth, Hayase, and Srinivasa 2023). Our methodology not only capitalizes on the efficiency of fine-tuning techniques, e.g. (Ding et al. 2023, 2022; Dettmers et al. 2023) but also innovatively addresses the scalability challenges inherent to BDLs. Overall, our work sets a new precedent in applying Bayesian approaches to computer vision tasks by offering a scalable and efficient framework for enhancing model performance.

3 Stein Variational Gradient Descent Primer

Bayesian inference techniques have been integral to the development of neural networks, with a rich history underscored by previous works (Neal et al. 2011; Neal 2012; MacKay 1991; Welling and Teh 2011; Blei, Kucukelbir, and McAuliffe 2017). Variational Inference (VI) (Blundell et al. 2015; Blei, Kucukelbir, and McAuliffe 2017) and Markov Chain Monte Carlo (MCMC) (Neal et al. 2011; Welling and Teh 2011) are two primary approximate Bayesian inference frameworks. The former substitutes the true posterior with a tractable alternative while the latter involves sampling. However, accurately computing the posterior with either MCMC or VI becomes computationally infeasible when dealing with large-scale networks containing millions of parameters. Although approximations can be obtained more efficiently with VI (Blundell et al. 2015), VI is also demonstrably too restrictive to resemble the *multi-modality* of the true posterior and suffers from mode collapse (Izmailov et al. 2021).

Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016), among other ensemble-based approaches, is an alternative approximate Bayesian technique that combines the strengths of MCMC and VI by transporting a set of parameter *particles* to fit the true posterior distribution, while encouraging diversity among the particles, by incorporating a repulsive term in the parameter updates. This diversity prevents the mode collapse and enables learning multiple models to represent various patterns in the data. Using n samples from the posterior (*i.e.* parameter particles), SVGD modifies the gradient descent as:

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_i - \epsilon_i \hat{\boldsymbol{\phi}}^*(\boldsymbol{\theta}_i) \quad \text{with}$$
$$\hat{\boldsymbol{\phi}}^*(\boldsymbol{\theta}) = \sum_{j=1}^{n} \left[k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}_j} \log p(\boldsymbol{\theta}_j | \mathcal{D}) - \frac{\gamma}{n} \nabla_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}) \right].$$

Here, $\boldsymbol{\theta}_i$ is the ith particle, $k(\cdot, \cdot)$ is a kernel function that measures the similarity between particles and γ is a hyperparameter. Notably, the kernel function encourages the particles to be dissimilar in order to capture more diverse samples from the posterior and γ controls the trade-off between the diversity of the samples versus the minimization of the loss.

4 Bayesian Low-Rank Learning (Bella)

The problem with the current SVGD and other ensemble-based methods in large deep neural networks is its huge computational cost. This renders it infeasible to train efficiently and to scale to a sufficient number of parameter particles for accurately approximating the posterior distribution, which currently remains coarse. In this work, we propose to capitalize on the low-rank representations of fine-tuning in order to construct a practical and scalable variant of SVGD. We note that while our approach may not fully capture the diversity of the multi-modal posterior, a recent conjecture (Ainsworth, Hayase, and Srinivasa 2023) suggests that these modes might result from parameter permutations in neural networks, leading to models that are functionally equivalent. Building on this idea, Bella could serve as a practical alternative with sufficient theoretical justification for ensemble methods.

Consider any dense layer, for which there is a fixed pre-trained weight matrix $\boldsymbol{\theta}_0 \in \mathbb{R}^{d_1 \times d_2}$ with d_1, d_2 the corresponding numbers of hidden units. We consider n low-rank perturbations of $\boldsymbol{\theta}_0$ as

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}_i = \boldsymbol{\theta}_0 + \mathbf{B}_i \mathbf{A}_i\,, \quad i = 1, \ldots, n. \tag{2}$$

where $\mathbf{B}_i \in \mathbb{R}^{d_1 \times r}$, $\mathbf{A}_i \in \mathbb{R}^{r \times d_2}$ are the low-dimensional update parameters, and $r \ll d_1, d_2$ is the update's rank. Now Bella proceeds as the joint SVGD on $(\mathbf{A}_i, \mathbf{B}_i)$, with updates

$$\mathbf{A}_i = \mathbf{A}_i - \epsilon_i \sum_{j=1}^{n} \hat{\boldsymbol{\phi}}_j^*(\mathbf{A}_i), \quad \mathbf{B}_i = \mathbf{B}_i - \epsilon_i \sum_{j=1}^{n} \hat{\boldsymbol{\phi}}_j^*(\mathbf{B}_i)$$

$$\text{with} \quad \hat{\boldsymbol{\phi}}_j^*(\mathbf{B}_i) = k_{i,j} \nabla_{\mathbf{B}_i} p\left(y \mid \mathbf{x}, \boldsymbol{\theta}_0 + \mathbf{B}_i \mathbf{A}_i\right) - \frac{\gamma}{n} \nabla_{\mathbf{B}_i} k_{i,j},$$
$$\hat{\boldsymbol{\phi}}_j^*(\mathbf{A}_i) = k_{i,j} \nabla_{\mathbf{A}_i} p\left(y \mid \mathbf{x}, \boldsymbol{\theta}_0 + \mathbf{B}_i \mathbf{A}_i\right) - \frac{\gamma}{n} \nabla_{\mathbf{A}_i} k_{i,j}\,,$$

where we denote $k_{i,j} = k\left(\mathbf{B}_j \mathbf{A}_j, \mathbf{B}_i \mathbf{A}_i\right)$. Here, we have placed a zero-mean Gaussian prior on $(\mathbf{A}_i, \mathbf{B}_i)$, but other choices are possible. Note that the kernel function on $(\mathbf{A}_i, \mathbf{B}_i)$ is given by $k\left(\boldsymbol{\theta}_0 + \mathbf{B}_j \mathbf{A}_j, \boldsymbol{\theta}_0 + \mathbf{B}_i \mathbf{A}_i\right)$, which ensures that the similarity is computed on the original parameter space, and in the commonly used case of shift-invariant kernels, this simplifies to $k\left(\mathbf{B}_j \mathbf{A}_j, \mathbf{B}_i \mathbf{A}_i\right)$. Further simplifications are obtained for specific kernel functions – in particular, in the case of Gaussian RBF, while the naive implementation would require the cost of $O(r d_1 d_2)$ for a single kernel evaluation, we can bring it down to $O(r^2(d_1 + d_2))$ using standard trace manipulation, as described in the Appendix. This procedure can be repeated across all dense layers.

Bella introduces a significant improvement in the efficiency of model training and execution. By utilizing the same pre-trained weights $\boldsymbol{\theta}_0$ across all parameter particles but allowing for individual low-rank adaptations $\Delta\boldsymbol{\theta}_i$, we achieve a balance between parameter sharing and the diversity necessary for effective learning. Bella significantly reduces the parameter space from the full matrix's $d_1 d_2$ to just $r(d_1 + d_2)$, thereby enhancing both efficiency and scalability. This setup not only reduces the computational burden during training but also streamlines the process at inference time. The heavy lifting is done once by loading the large base model $\boldsymbol{\theta}_0$, and the lightweight low-rank adapters $\Delta\boldsymbol{\theta}_i$ can be dynamically applied with minimal overhead in order to approximate the posterior predictive distribution as $p(y^* \mid \mathbf{x}^*, \mathcal{D}) \approx \frac{1}{n} \sum_{i=1}^{n} p(y^* \mid \mathbf{x}, \boldsymbol{\theta}_0 + \Delta\boldsymbol{\theta}_i)$. This approach is particularly advantageous in large-scale models, where the weight matrices $\boldsymbol{\theta}_0$ are of substantial dimensions.

5 Empirical Experiments and Results

In this section, we provide an in-depth overview of our experimental setup, detailing the methodology, equipment, and procedures employed in the implementation of Bella. We aim to compare Bella with established ensemble-based methods and SVGD, both of which have demonstrated effectiveness in BDLs (Seligmann et al. 2024; Lakshminarayanan, Pritzel, and Blundell 2017).

5.1 Experimental Set-up

Datasets. In this research, we have employed a variety of datasets, each selected for their relevance and contribution, they include `CIFAR-10`, `CIFAR-100` (Krizhevsky, Hinton

et al. 2009), CIFAR-10-C (Hendrycks and Dietterich 2019), STL-10 (Coates, Ng, and Lee 2011), CAMELYON17 (Bandi et al. 2018), ImageNet (Russakovsky et al. 2015), and DomainNet (Peng et al. 2019). We also consider VQA v2 dataset utilized for Visual Question Answering (VQA). Additional details are in the Appendix.

Neural Architecture. In our experiments, we employed the CLIP ViT-B/32 model (Ilharco et al. 2021), a pre-trained variant utilizing contrastive supervision from image-text pairs, as initially introduced in the seminal CLIP research (Radford et al. 2021). We conducted end-to-end fine-tuning of the image encoder, adjusting all model parameters, a strategy typically yielding higher accuracy than training only the final linear layer. For both our ensemble-based methods, we consider conventional ensemble and SVGD. We use the average logits (unnormalized outputs) to produce the output (Gontijo-Lopes, Dauphin, and Cubuk 2021). For VQA task, we employ the SoTA LLaVA-1.5-7B (Liu et al. 2023b) to showcase the effectiveness of Bella on large-scale network architecture. Detailed hyper-parameters are in the Appendix.

5.2 Cost Efficiency

Cost comparisons (in terms of trainable parameters, memory and storage) are shown in Table 1. The performance comparison of our Bella models with their respective base models, as well as with Vision Bayesian Lora (VBL)—a derived Laplace's approximation from (Yang et al. 2024) for vision tasks—is shown in Table 2. In particular, the gray columns Table 1 indicate the models used to generate the results in Table 2.

Impressively, across a spectrum of benchmark computer vision datasets such as CIFAR-10, CIFAR-100, Camelyon17, and ImageNet, the Bella models demonstrate superior performance. This is achieved with only a fraction of cost (trainable parameters) as shown in Table 1, underscoring the models' proficiency in parameter efficiency without compromising on accuracy. Further, the Bella models surpass the performance of *Single* models, while employing a comparable level of computational resources—see the Memory Consumption of models used, in Table 1 reporting approximately 4.5 GB for Bella compared to 5.05 GB for Single. The results across the benchmarks attest to the efficacy of our methodology.

Notably, in Table 2 with Bella SVGD, with 1.6% of the trainable parameters in comparison to the Single baseline, leads to approximately 2% and 10% increase in performance on the OOD tasks of CAMELYON17 and DomainNet, respectively; whilst achieving comparable performance with the current SVGD implementation (SVGD baseline model).

Along with Table 2, Table 1 demonstrates the primary benefit of our approach—*the significant reduction in memory and storage needs*. For example, as seen in Table 1, we obtain approximately a 5× reduction in model size—that is from 2,222 MB for a 5 particle SVGD baseline model to just 439 MB for a Bella SVGD model with 5 particles. This efficiency not only reduces the demands on GPUs but also minimizes potential I/O bottlenecks.

Moreover, the reduced GPU demand, as shown in Table 1, facilitates larger mini-batch sizes during training to speedup the training process. More crucially, it enables one to enhance the Bayesian posterior's parameter particles to over 100, a feat previously unattainable with current SVGD implementations. Significantly, constructing a 100 parameter Bayesian approximation consumes only 5.19 GB memory compared to current SVGD implementations for Bayesian models (SVGD base) needing over 6 GB for *even a mere 3 particle approximation*.

Interestingly, our results for Bella models are on-par with SVGD and ensemble approximations of the posterior. This provides empirical evidence that with constrained model parameters, it is still possible to reach the diverse modes of the posterior. Further, the results support recent conjectures on mode connectivity (Gueta et al. 2023).

5.3 Out-Of-Disrtibution (OOD) Datasets

Assessing the robustness of machine learning systems to unseen conditions is crucial, particularly regarding their ability to generalize to out-of-distribution (OOD) data. We evaluate robustness to OOD by utilizing multiple OOD benchmarks and estimating accuracy under various noise levels (distribution shifts) to further assess the performance of different baselines in these challenging scenarios.

First, we use the DomainNet dataset, which is one of the most diverse domain adaptation datasets and spans a wide range of visual styles, from real images to abstract art. This variety provides a challenging test bed for algorithms aiming to bridge different visual domains. Our study involves training the CLIP network on the 'Real' subset of DomainNet and evaluating its generalization across various domains. Additionally, we use CIFAR-10-C, a corrupted version of CIFAR-10, to further assess the generalization and robustness of models trained on CIFAR-10 datasets. The results for both datasets are presented in Table 3.

Models	Bseline Models (SVGD)				Bella Models (SVGD)				Single
	$n=3$	$n=5$	$n=20$	$n=40$	$n=3$	$n=5$	$n=20$	$n=100$	$n=1$
Trainable Parameters	340M	567M	1.76B	3.51B	1.10M	1.84M	7.37M	36.86M	113M
Memory Consumption (RAM in GB)	6.71	8.35	26.08	48.45	4.48	4.50	4.63	5.19	5.05
Storage Consumption (MB)	1321	2222	8868	17735	436	439	460	572	433

Table 1: Computational cost to train different models based on CLIP architecture for different datasets. Notably, with SVGD Baseline Models, we can only train up to n=40 particles on a A6000 48 GB GPU, while we can increase to more than 100 parameter particles with our Bella method with negligible increase of GPU consumption. The grey columns correspond to costs (parameters) of models in Table 2.

Datasets	Bella Models		Baseline Models (Base)			
	Ensemble (n=5)	SVGD (n=5)	Ensemble (n=5)	SVGD (n=5)	Single	VBL
CIFAR10	97.32 ± 0.37%	**97.57** ± 0.38%	97.26 ± 0.28%	97.56 ± 0.31%	*96.86* ± 0.43%	94.04 ± 0.37%
CIFAR100	86.02 ± 0.48%	**87.63** ± 0.46%	86.65 ± 0.24%	87.03 ± 0.29%	*85.14* ± 1.3%	85.01 ± 1.21%
CAMELYON17	93.11 ± 1.36%	93.61 ± 1.28%	93.29 ± 0.94%	**93.98** ± 1.23%	*90.25* ± 2.31%	91.58 ± 1.38%
DomainNet	80.34 ± 2.36%	81.41 ± 3.06%	82.96 ± 2.11%	**83.66** ± 1.76%	*69.75* ± 4.86%	80.60 ± 2.37%
ImageNet	77.29	78.24	78.93	**79.36**	*76.87*	-

Table 2: Comparing Bella models with their baseline (*Base*) counterparts on *vision benchmarks*. Bella models are well-performing and on par with their baseline Ensemble, SVGD, and VBL whilst consuming only a fraction of cost. The best-performing results are in **bold**, the second-best in underline, and the least favorable are in *italic* for emphasis.

Furthermore, the STL-10 dataset, which has significant label overlap with CIFAR-10, serves as a relevant OOD test case for CIFAR-10 models (see the Appendix).

Results from Table 3 and the Appendix show that Bella models, demonstrating significantly better efficiency, achieve competitive performance compared to more resource-intensive implementations with Ensemble and SVGD baselines across both DomainNet and CIFAR-10-C benchmarks. All Bayesian approximations, including our scalable and efficient method, outperform the single model baseline.

5.4 Comparing Uncertainty Estimations

Bayesian models capable of providing a theoretical basis for measuring *model uncertainty*. Also known as epistemic uncertainty, refers to uncertainty stemming from limitations in our knowledge or understanding of the underlying data generating process or the model itself. One of the ways to quantify model uncertainty is through mutual information estimates, following (Gal 2016).

Mutual Information (MI). This is the mutual information between the output prediction and the posterior over model parameters $\boldsymbol{\theta}$, and can be used as a measure of epistemic (model) uncertainty. It can be expressed as: $\text{MI}(\boldsymbol{\theta}, y \mid \mathcal{D}, \mathbf{x}) = H[p(y \mid \mathcal{D}, \mathbf{x})] - \mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})} H[p(y \mid \boldsymbol{\theta}, \mathbf{x})]$ If the parameters at input are well defined (e.g., data seen during training), then we would gain little information from the obtaining label, or the MI measured will be low.

We employ MI to measure uncertainty to investigate whether the Bella approximations of the posterior leads to uncertainty estimates commensurate with those obtained from SVGD baselines. This provides empirical evidence of a functional equivalence of the Bella approximations of the posterior to that obtained from the current computationally intensive implementation of SVGD.

Datasets. We utilize the CIFAR-10-C task, featuring corrupted images, to examine the uncertainty of model predictions trained on the standard CIFAR-10 dataset. Additionally, we assess the uncertainty measures on the CAMELYON17 dataset, which is characterized by inherent dataset shifts within itself.

Results. Figure 2 demonstrates the effectiveness of our approach to estimate uncertainty. Our Bella perform similarly to the SVGD base model, with a slightly better uncertainty on misclassified images of CAMELYON17 and corrupted CIFAR-10-C datasets (under brightness corruption with the maximum intensity), see details in the Appendix.

5.5 Robustness against Adversarial Examples

In this section, we examine the resilience of our proposed Bella against adversarial attacks, specifically employing the L_∞ Fast Gradient Sign Method (FGSM) across various attack budgets as detailed in Figure 3. This analysis aims to benchmark the robustness of our method in comparison to traditional models under adversarial conditions. We employ the robustness benchmark (Papernot et al. 2018) to deploy the attack on CIFAR-10 test set and report results in Figure 3.

The findings presented in Figure 3 reveal that conventional models such as SVGD and Ensemble exhibit just slightly greater resistance to adversarial attacks. We attribute this enhanced robustness to the broader diversity in model parameters, which stems from their capacity to adjust the entire network's parameters, unlike the Bella models.

Significantly, despite operating within the same computational constraints as a singular network model, our Bella demonstrates enhanced efficacy in mitigating adversarial attacks, thereby bolstering its robustness.

5.6 Ablation Studies

This section undertakes a series of ablation studies to examine the effects of various components within Bella. Our analysis includes an exploration of the training costs associated with

Models	DomainNet					Avg.	CIFAR-10-C			Avg.
	Real	Clip-Art	Infograph	Paint	Sketch		Gaussian Blur	Pixelate	Spatter	
Single base	74.61	55.29	31.81	53.87	43.84	51.89	90.58	77.94	92.14	78.53
VBL	82.97	59.94	26.56	52.81	46.96	53.85	89.34	76.43	89.21	70.76
Ensemble Bella	82.70	61.22	28.15	55.32	51.58	55.60	93.30	85.86	94.83	82.24
Ensemble base	85.07	65.40	36.07	57.90	54.22	59.73	91.92	86.49	93.45	84.61
SVGD Bella	84.47	63.67	32.22	56.83	54.86	58.41	94.05	88.70	94.85	83.77
SVGD base	85.42	65.53	36.58	58.18	55.47	60.24	93.02	86.75	95.19	86.84

Table 3: The out-of-distribution generalization performance of Bella, measured by accuracy (for number of particles $n = 5$). Each column represents a specific shift, either real (DomainNet) or artificial (CIFAR-10-C).

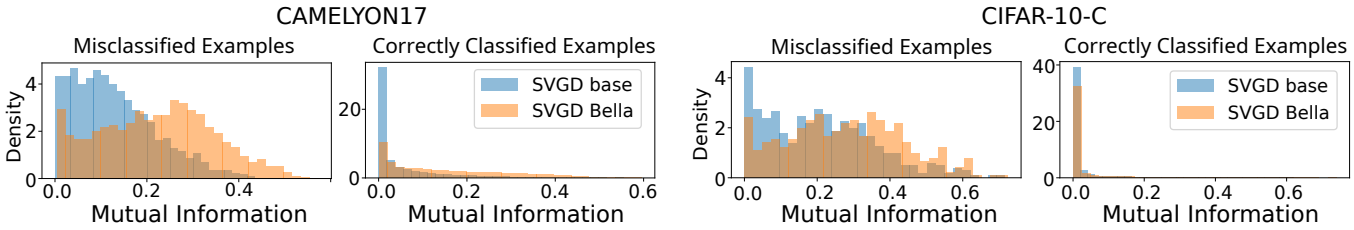

Figure 2: Evaluation of uncertainty estimations using Mutual Information on CAMELYON17 and CIFAR-10-C datasets. ↑ MI for Misclassified Examples is better—denoted by the distribution shifting →. In contrast, ↓ MI for Correctly Classified Examples is better—denoted by the distribution shifting ←.

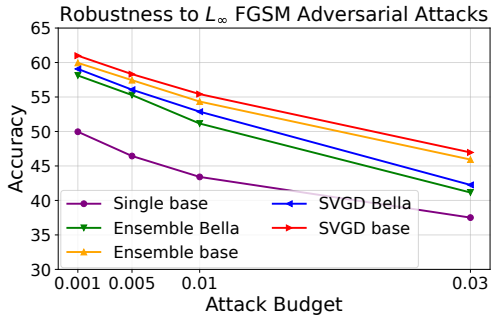

Figure 3: Comparison of model robustness to L_∞ FGSM adversarial attacks across varied attack budgets on the CIFAR-10 dataset.

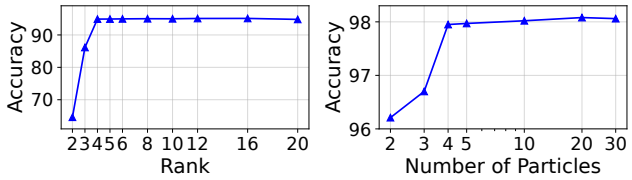

Figure 4: The impact of ranks on CAMELYON17 performance (left), as well as the impact of the number of parameter particles on CIFAR-10 (right) on Bella performance.

different ranks and their consequent influence on model performance. Given that Bella incorporates multiple parameter particles, we also delve into how varying the number of these particles affects Bella's efficacy. Additionally, we explore the application of low-rank adapters across different layers and assess their impact. Further details on other studies are in the Appendix. We show in the Appendix that we achieve state-of-the-art performance on CAMELYON17.

Ablations on rank r. As outlined in Section 4, we substitute the network's extensive full matrix with low-rank matrices defined by the 'rank' parameter (r). This section aims to assess how this parameter influences Bella's performance.

To assess the influence of rank, we conduct an analysis on the challenging, large-scale, CAMELYON17 task. The results, reported in Figure 4 (left), elucidate the relationship between rank size and performance. While utilizing a smaller rank considerably reduces the parameter space, it also restricts Bella's learning capacity, hindering its ability to achieve optimal performance. In contrast, increasing the rank to 4 significantly enhances performance. However, performance tends to plateau at a rank of 16, indicating a saturation point.

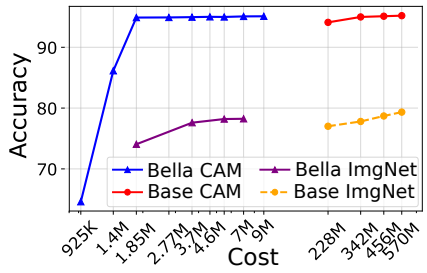

Figure 5: Bella achieves similar Accuracy with full SVGD (Base) only with a fraction of cost.

Ablations on the number of particles n. In this section, we delve into the influence of the quantity of parameter particles on the performance of Bella. The findings, depicted in the right in Figure 4, reveal an improvement in Bella's performance with an increase in the number of particles.

This outcome is both intuitive and insightful, as a larger ensemble of parameter particles enhances the approximation of the Bayesian posterior more effectively.

Ablations on the number of trainable parameters. We will compare the number of trainable parameters between Bella and full SVGD in model performance. Critically, higher number of trainable parameters means higher cost to train. To show generalization, we also employ another large-scale challenging dataset Imagenet in this experiment.

The plot in Figure 5 for the CAMELYON17 dataset begins with Bella at $r = 2$ (comprising 5 particles) and concludes with full SVGD (SVGD base) using the same number of particles. As r increases, the accuracy of Bellas improves, eventually plateauing at approximately 95.08%, which is comparable to the accuracy of full SVGD (95.21%). This highlights the efficiency and advantages of our proposed Bella. Remarkably, Bella achieves performance on par with full SVGD models despite utilizing a significantly smaller pool of trainable parameters—approximately 0.3% for $r = 4$—compared to the more parameter-intensive alternatives.

Calibration Study. Table 4 presents a comparison of the Bella models (SVGD and ensemble) against baseline models, including Variational Inference (VI) (Kim and Hospedales 2023) and Stochastic-Gradient Langevin Dynamic (SGLD) (Welling and Teh 2011), using several key metrics: Expected Calibration Error (ECE), Maximum Calibration Error (MCE), Brier score, and Area Under the Receiver Operating Characteristic curve (AUROC). Lower ECE and MCE values indicate that the model's probability esti-

mates are more reliable. A lower Brier score reflects better accuracy in the model's probability predictions, while a higher AUROC demonstrates superior discrimination between classes. Across the datasets, the Bella models demonstrate performance comparable to the baseline models. This suggests that the Bella models, while utilizing far fewer parameters for training, offer a strong alternative to traditional baseline approaches while maintaining competitive calibration and Discriminative capability.

5.7 Generalization to a Visual Question Answer

In this section, we extend the application of our Bella to another challenging vision task, Visual Question Answering (VQA), as detailed in (Antol et al. 2015). We leverage the state-of-the-art, pre-trained, large multi-modal model LlaVA (Liu et al. 2023b,a) for this purpose. Utilizing LlaVA transforms VQA into a process where an image and a natural-language question are inputs, and the model generates a free-form, open-ended text answer. Answering questions in VQA requires various intelligence capabilities, including not only image recognition but also complex reasoning. For this task, we employ `VQA v2` (Antol et al. 2015) dataset containing 204,721 images, more than 1 Billion (1B) questions and 10B ground-truth answers in total. There are three main types of answers: Yes/No, a Number, and Other.

Model. We employed our proposed Bella on top of LlaVA-1.5-7B (Liu et al. 2023b,a). Further details about the dataset, model and metrics are deferred to the Appendix.

Accuracy. The primary outcomes for Yes/No and Number answer queries are detailed in Table 5. We chose these question types to ensure a fair comparison and avoid semantic mismatches in open-ended answers in Others. A distinctive feature of Bella is its reduced uncertainty estimates for correct predictions, coupled with increased uncertainty estimates for incorrect ones. Moreover, it surpasses the Single base model regarding Accuracy and Exact Match metrics.

Data.	Metric	Bella		Base				
		Ens.	SVGD	Single	VI	SGLD	Ens.	SVGD
CIFAR10	ECE ↓	1.4	1	8.1	0.99	0.94	2.5	**0.55**
	MCE ↓	65	26	35	33	24	23	**21**
	Brier ↓	0.40	**0.32**	0.49	0.43	0.36	0.49	0.33
	AUROC ↑	99.98	**99.99**	99.78	99.93	99.94	**99.99**	**99.99**
CIFAR100	ECE ↓	5.2	4.9	6.5	4.2	3.8	5.1	**3.3**
	MCE ↓	19	**15**	86	47	41	48	46
	Brier ↓	0.23	0.19	0.23	0.2	0.19	0.19	**0.18**
	AUROC ↑	**99.88**	**99.88**	99.71	99.87	99.83	**99.88**	**99.88**
CAMELYON	ECE ↓	5.5	5.3	6.4	5	5.2	5.3	**4.9**
	MCE ↓	31	29	34	36	31	30	**21**
	Brier ↓	5.2	5.1	5.5	5.4	5.4	5.1	**4.5**
	AUROC ↑	98.4	98.7	98.12	98.6	98.6	98.5	**98.81**
Domain Net	ECE ↓	5.9	5.8	7.9	5.1	4.6	5.2	**4.9**
	MCE ↓	40	37	61	40	41	37	**34**
	Brier ↓	5.3	5.2	6.3	5	4.4	5	**4.7**
	AUROC ↑	98.31	98.46	98.1	98.99	**99.21**	98.88	99.01

Table 4: Calibration comparison of Bella models (SVGD and Ensemble) with baseline methods across multiple datasets.

Models	Ent. Corrects (↓)	Ent. Incorrect (↑)	Acc. (↑)	Match (↑)
VQA: Evaluation of Yes/No Questions				
Single base	0.3336	0.5920	91.59	86.74
Ens. Bella	0.3438	0.5935	91.20	86.21
SVGD Bella	**0.3245**	**0.5950**	**92.46**	**87.83**
VQA: Evaluation of Number Questions				
Single base	0.4148	0.9911	58.69	49.26
Ens. Bella	0.4248	**0.9929**	57.86	48.68
SVGD Bella	**0.4059**	0.9816	**60.19**	**50.99**

Table 5: LLaVA-VQA Results: evaluating the accuracy and entropy of correct vs. incorrect predictions (Ens: Ensemble).

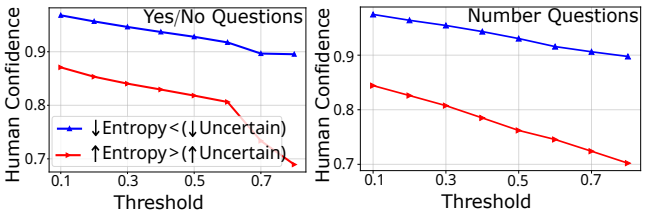

Figure 6: Correlation between model certainty and human confidence. This negative correlation suggests that the model's entropy can serve as a reliable gauge of certainty, mirroring human judgment.

Notably, our method is efficient, particularly in contrast to the resource-intesive baseline Bayesian models tailored for this task (*e.g.* (Abbasnejad et al. 2019, 2020b,a)). The computational requirements for using these baselines render the application of full SVGD or ensemble alternatives impractical, thereby highlighting the impact on practical applications by harnessing the effectiveness of Bella.

Model Uncertainty and Human Confidence. Further, we investigate the relationship between model uncertainty attained from Bella and human confidence, facilitated by multiple annotators in the VQA dataset. For this, we measure the model disagreement by measuring the correlation between predictive entropy and human annotations of the answers.

In Figure 6 we show the correlation between the entropy of the model outputs and human confidence levels. By setting a specific threshold for the model's entropy, we effectively bifurcate our predictions into two distinct categories: those with lower entropy fall into the 'Low Entropy' group, signaling reduced uncertainty within the model's assessments, while predictions exceeding the threshold are allocated to the 'High Entropy' group, indicative of greater uncertainty. Intriguingly, our observations reveal a consistent negative correlation between the model's entropy levels and human confidence across the different query types (Yes/No and Number questions). This pattern suggests that the model's entropy can serve as a reliable gauge of certainty, mirroring human judgment in its response to varying levels of uncertainty.

6 Conclusion

We present Bella, an innovative efficient Bayesian Neural Network (BNN) approximation using a base pre-trained model. Bella achieves remarkable compatibility with full-rank BNN methods like SVGD and Ensembles, surpassing single-network solutions in classification, OOD generalization, and uncertainty quantification. This approach enables

scalable, reliable BNN implementations, demonstrating the benefits of simple and efficient training over traditional single fine-tuning techniques.

References

Abbasnejad, E.; Abbasnejad, I.; Wu, Q.; Shi, J.; and Hengel, A. v. d. 2020a. Gold seeker: Information gain from policy distributions for goal-oriented vision-and-langauge reasoning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Abbasnejad, M. E.; Shi, J.; van den Hengel, A.; and Liu, L. 2020b. GADE: A Generative Adversarial Approach to Density Estimation and its Applications. *Int. J. Comput. Vis.*, 128(10): 2731–2743.

Abbasnejad, M. E.; Shi, Q.; van den Hengel, A.; and Liu, L. 2019. A Generative Adversarial Density Estimator. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*, 10782–10791.

Ainsworth, S.; Hayase, J.; and Srinivasa, S. 2023. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *International Conference on Learning Representations (ICLR)*.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. VQA: Visual Question Answering. In *International Conference on Computer Vision (ICCV)*.

Bandi, P.; Geessink, O.; Manson, Q.; Van Dijk, M.; Balkenhol, M.; Hermsen, M.; Bejnordi, B. E.; Lee, B.; Paeng, K.; Zhong, A.; et al. 2018. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE Transactions on Medical Imaging*.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*.

Blundell, C.; Cornebise, J.; Kavukcuoglu, K.; and Wierstra, D. 2015. Weight uncertainty in neural network. In *International Conference on Machine Learning (ICML)*.

Challenge, G. 2024. Camelyon17 Leaderboard. https://camelyon17.grand-challenge.org/evaluation/challenge/leaderboard/. Accessed: 08-Mar-2024.

Chen, P.; and Ghattas, O. 2020. Projected Stein variational gradient descent. *Advances in Neural Information Processing Systems (NeurIPS)*.

Coates, A.; Ng, A.; and Lee, H. 2011. An Analysis of Single Layer Networks in Unsupervised Feature Learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Daxberger, E.; Kristiadi, A.; Immer, A.; Eschenhagen, R.; Bauer, M.; and Hennig, P. 2021. Laplace Redux–Effortless Bayesian Deep Learning. In *NeurIPS*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLORA: efficient finetuning of quantized LLMs. In *International Conference on Neural Information Processing Systems (NIPS)*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; Yi, J.; Zhao, W.; Wang, X.; Liu, Z.; Zheng, H.-T.; Chen, J.; Liu, Y.; Tang, J.; Li, J.; and

Sun, M. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*.

Ding, N.; Qin, Y.; Yang, G.; Wei, F.; Yang, Z.; Su, Y.; Hu, S.; Chen, Y.; Chan, C.-M.; Chen, W.; et al. 2022. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint arXiv:2203.06904*.

Doan, B. G.; Abbasnejad, E. M.; Shi, J. Q.; and Ranasinghe, D. C. 2022. Bayesian Learning with Information Gain Provably Bounds Risk for a Robust Adversarial Defense. In *International Conference on Machine Learning (ICML)*.

Doan, B. G.; Nguyen, D. Q.; Montague, P.; Abraham, T.; De Vel, O.; Camtepe, S.; Kanhere, S. S.; Abbasnejad, E.; and Ranasinghe, D. C. 2024. Bayesian Learned Models Can Detect Adversarial Malware for Free. In *European Symposium on Research in Computer Security (ESORICS)*, 45–65.

Doan, B. G.; Yang, S.; Montague, P.; De Vel, O.; Abraham, T.; Camtepe, S.; Kanhere, S. S.; Abbasnejad, E.; and Ranashinghe, D. C. 2023. Feature-Space Bayesian Adversarial Learning Improved Malware Detector Robustness. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 37(12): 14783–14791.

Dusenberry, M.; Jerfel, G.; Wen, Y.; Ma, Y.; Snoek, J.; Heller, K.; Lakshminarayanan, B.; and Tran, D. 2020. Efficient and scalable bayesian neural nets with rank-1 factors. In *International Conference on Machine Learning (ICML)*.

Fan, X.; Zhang, S.; Chen, B.; and Zhou, M. 2020. Bayesian attention modules. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gal, Y. 2016. *Uncertainty in deep learning*. Ph.D. thesis, University of Cambridge.

Gontijo-Lopes, R.; Dauphin, Y.; and Cubuk, E. D. 2021. No one representation to rule them all: Overlapping features of training methods. *arXiv preprint arXiv:2110.12899*.

Gueta, A.; Venezian, E.; Raffel, C.; Slonim, N.; Katz, Y.; and Choshen, L. 2023. Knowledge is a Region in Weight Space for Fine-tuned Language Models. *arXiv preprint arXiv:2302.04863*.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-Efficient Transfer Learning for NLP. *arXiv:1902.00751*.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations (ICLR)*.

Ilharco, G.; Wortsman, M.; Wightman, R.; Gordon, C.; Carlini, N.; Taori, R.; Dave, A.; Shankar, V.; Namkoong, H.; Miller, J.; Hajishirzi, H.; Farhadi, A.; and Schmidt, L. 2021. OpenCLIP.

Izmailov, P.; Maddox, W. J.; Kirichenko, P.; Garipov, T.; Vetrov, D.; and Wilson, A. G. 2020. Subspace inference for Bayesian deep learning. In *Uncertainty in Artificial Intelligence*.

Izmailov, P.; Vikram, S.; Hoffman, M. D.; and Wilson, A. G. 2021. What Are Bayesian Neural Network Posteriors Really Like? In *International Conference on Machine Learning (ICML)*.

Kim, M.; and Hospedales, T. 2023. BayesDLL: Bayesian Deep Learning Library. *arXiv preprint arXiv:2309.12928*.

Koh, P. W.; Sagawa, S.; Marklund, H.; Xie, S. M.; Zhang, M.; Balsubramani, A.; Hu, W.; Yasunaga, M.; Phillips, R. L.; Gao, I.; Lee, T.; David, E.; Stavness, I.; Guo, W.; Earnshaw, B. A.; Haque, I. S.; Beery, S.; Leskovec, J.; Kundaje, A.; Pierson, E.; Levine, S.; Finn, C.; and Liang, P. 2021. WILDS: A Benchmark of in-the-Wild Distribution Shifts. In *International Conference on Machine Learning (ICML)*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto. Technical report.

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

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

Lin, Z.; Madotto, A.; and Fung, P. 2020. Exploring Versatile Generative Language Model Via Parameter-Efficient Transfer Learning. In *Findings of the Association for Computational Linguistics (EMNLP)*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2023a. Improved Baselines with Visual Instruction Tuning. *arXiv preprint arXiv:2310.03744*.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023b. Visual Instruction Tuning. In *Advanced in Neural Information Processing Systems (NeurIPS)*.

Liu, Q.; and Wang, D. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in Neural Information Processing Systems (NeurIPS)*.

Liu, X.; Li, Y.; Chongruo, W.; and Cho-Jui, H. 2019. ADV-BNN: Improved Adversarial Defense Through Robust Bayesian Neural Network. In *International Conference on Learning Representations (ICLR)*.

MacKay, D. 1991. Bayesian Model Comparison and Backprop Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Neal, R. M.; et al. 2011. MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11).

Papernot, N.; Faghri, F.; Carlini, N.; Goodfellow, I.; Feinman, R.; Kurakin, A.; Xie, C.; Sharma, Y.; Brown, T.; Roy, A.; Matyasko, A.; Behzadan, V.; Hambardzumyan, K.; Zhang, Z.; Juang, Y.-L.; Li, Z.; Sheatsley, R.; Garg, A.; Uesato, J.; Gierke, W.; Dong, Y.; Berthelot, D.; Hendricks, P.; Rauber, J.; and Long, R. 2018. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. *arXiv preprint arXiv:1610.00768*.

Peng, X.; Bai, Q.; Xia, X.; Huang, Z.; Saenko, K.; and Wang, B. 2019. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision (CVPR)*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*.

Rebuffi, S.-A.; Bilen, H.; and Vedaldi, A. 2017. Learning multiple visual domains with residual adapters. *arXiv:1705.08045*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*.

Seligmann, F.; Becker, P.; Volpp, M.; and Neumann, G. 2024. Beyond deep ensembles: A large-scale evaluation of bayesian deep learning under distribution shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 36.

Vadera, M. P.; Cobb, A. D.; Jalaian, B.; and Marlin, B. M. 2022. Impact of parameter sparsity on stochastic gradient mcmc methods for bayesian deep learning. *arXiv preprint arXiv:2202.03770*.

Wang, X.; Aitchison, L.; and Rudolph, M. 2024. LoRA ensembles for large language model fine-tuning.

Welling, M.; and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML)*, 681–688. Citeseer.

Wild, V. D.; Ghalebikesabi, S.; Sejdinovic, D.; and Knoblauch, J. 2023. A Rigorous Link between Deep Ensembles and (Variational) Bayesian Methods. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Wilson, A. G.; Izmailov, P.; Hoffman, M. D.; Gal, Y.; Li, Y.; Pradier, M. F.; Vikram, S.; Foong, A.; Lotfi, S.; and Farquhar, S. 2022. Evaluating approximate inference in Bayesian deep learning. In *NeurIPS 2021 Competitions and Demonstrations Track*, 113–124. PMLR.

Yang, A. X.; Robeyns, M.; Wang, X.; and Aitchison, L. 2024. Bayesian low-rank adaptation for large language models. In *International Conference on Learning Representations (ICLR)*.

Ye, N.; and Zhu, Z. 2018. Bayesian adversarial learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhang, S.; Fan, X.; Chen, B.; and Zhou, M. 2021. Bayesian attention belief networks. In *International Conference on Machine Learning (ICML)*.