Pre-Train Your Loss: Easy Bayesian Transfer Learning with Informative Prior

Ravid Shwartz-Ziv [*1] **Micah Goldblum** [*1] **Hossein Souri** [2] **Sanyam Kapoor** [3] **Chen Zhu** [1]
Yann LeCun [1 2] **Andrew Gordon Wilson** [2]

Abstract

Deep learning is increasingly moving towards a transfer learning paradigm whereby large "foundation models" are fine-tuned on downstream tasks, starting from an initialization learned on the source task. But an initialization contains relatively little information about the source task. Instead, we show that we can learn highly informative posteriors from the source task, which serves as the basis for priors that modify the whole loss surface on the downstream task. This simple modular approach enables significant performance gains and more data-efficient learning on various downstream classification and segmentation tasks, serving as a drop-in replacement for standard pre-training strategies.

1. Introduction

Transfer learning is quickly becoming a mainstream practice in deep learning. Typically, large "foundation models" are pre-trained on massive volumes of source data, and then the learned parameter vector is used as an initialization for training in a downstream task. While this approach has had great empirical success, reliance on initialization is a very limited way to perform transfer learning.

We propose to instead use a re-scaled Bayesian parameter posterior from the first task as a *pre-trained* prior for the downstream task. Since the negative log posterior on the downstream task is our loss function, this procedure reshapes our training objective on the downstream task to reflect more nuanced information that we have learned from the source task, as in Figure 2. The posterior on the source task is re-scaled before use as a prior on the downstream task to reflect the belief that the source and downstream tasks are drawn from different distributions.

[*]Equal contribution [1]New York University [2]Meta AI Research [3]University of Maryland. Correspondence to: Ravid Shwartz-Ziv <rs8020@edu.nyu>.

First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022, Baltimore, Maryland, USA, PMLR 162, 2022.

With our highly informative prior for the downstream task, we can then proceed with optimization or perform full Bayesian model averaging with the posterior on the downstream task. We find both procedures profoundly improve upon standard deep transfer learning, making use of a simple pipeline that involves easy-to-use pre-existing components. Indeed, the simplicity of the approach, combined with its promising empirical performance, is one of its greatest features — enabling its use as a drop-in replacement for standard approaches for deep transfer learning.

In this work, we demonstrate that pre-training and downstream loss surfaces are closely aligned but with key differences that make it advantageous to modify the loss surface on the downstream task. Furthermore, we show how to parameterize and re-scale pre-trained priors for optimizing downstream Bayesian inference. Finally, we observe through extensive experiments across both image classification and semantic segmentation on multiple neural architectures and on both supervised and self-supervised (SimCLR (Chen et al., 2020)) pre-training loss functions that Bayesian inference is particularly advantageous in the transfer learning setting.

2. Background

Bayesian Neural Networks. Consider a neural network f with weights w and a training dataset $\mathcal{D} = \{(x^{(i)}, y^{(i)})\}$ of independently drawn samples. In standard non-Bayesian neural network training, we minimize the negative log posterior: $-\log p(w|\mathcal{D}, f) \propto -\log p(\mathcal{D}|w, f) - \log p(w|f)$. In Bayesian modeling, we instead make predictions with a *Bayesian Model Average* (BMA) of all models weighted by their posterior probabilities: $p(y|x, \mathcal{D}, f) = \int p(y|x, w, f)p(w|\mathcal{D}, f)dw$. There are many ways to approximate this integral, such as MCMC, or variational approaches, where the most common prior is zero-mean isotropic one (Wilson and Izmailov, 2020; **?**). However, there are also specialized priors, including heavy-tailed priors (Fortuin et al., 2021; Izmailov et al., 2021b), noise-contrastive priors for high uncertainty under distribution shift (Hafner et al., 2020), and input-dependent priors for domain generalization (Izmailov et al., 2021a).

Transfer Learning. In *transfer learning*, we wish to re-

cycle the representation learned on one task to improve performance on another. Transfer learning is now widely applied in deep learning. It forms the basis for *foundation models* (Bommasani et al., 2021; Dosovitskiy et al., 2020; Dai et al., 2021; Ren et al., 2015; Chen et al., 2018; Howard and Ruder, 2018; Devlin et al., 2018), which are exceptionally large neural networks pre-trained on massive volumes of data and then fine-tuned on a downstream task. Recent work has found that self-supervised pre-training can transfer better than supervised pre-training (He et al., 2020), in line with our finding in Section 4 that self-supervised pre-trained priors transfer better.

3. Bayesian Transfer Learning with Pre-Trained Priors

To transfer knowledge acquired through pre-training to downstream tasks, we adopt a pipeline with three simple components composed of easy-to-use existing tools:

1. First, we fit a probability distribution with closed-form density to the posterior over feature extractor parameters using a pre-trained checkpoint and SWAG (Maddox et al., 2019) (Section 3.1).

2. Second, we re-scale this distribution, viewed as a prior for new tasks, to reflect the mismatch between the pre-training and downstream tasks. To this end, we tune a single scalar coefficient on held-out validation data (Section 3.2).

3. Finally, we plug the re-scaled prior into a Bayesian inference algorithm, along with a zero-mean isotropic Gaussian prior over added parameters (e.g., classification head) to solve the downstream task (Section 3.3).

The simplicity and modularity of this framework are key strengths. By carefully combining easy-to-use existing components, we will see in Section 4 that we can straightforwardly improve the default approaches to deep transfer learning. The potential for impact is significant: we can use this pipeline as a drop-in replacement for standard procedures used in deploying foundation models. At the same time, there is significant novelty: while Bayesian neural networks are typically used with simple zero-mean isotropic priors, we leverage the significant developments in self-supervised learning to produce highly informative priors.

Moreover, in the following subsections, we will gain a nuanced understanding of these three components and the key considerations for practical success. We discuss computational considerations in Appendix B. In Section4, we present our main empirical evaluations.

3.1. Learning Transferable Priors

We begin by building a probability distribution over the parameters of a feature extractor which represents knowledge we acquire from pre-training. To this end, we fit the distribution to the Bayesian posterior, or regularized loss function, on the pre-training task. This pre-training posterior will become the prior for downstream tasks.

For the prior, we choose the SWA-Gaussian (SWAG) (Maddox et al., 2019) due to its simplicity, scalability, popularity, and non-diagonal covariance. After obtaining a closed-form distribution using SWAG, we remove the head from on top of the feature extractor, and consider only the distribution's restriction to the parameters of the feature extractor.

Alignment of Pre-Training and Downstream Loss. In order to find if the covariance matrix of our learned prior is more effective than an isotropic one, we compare the alignment of the learned covariance matrix with a downstream task's test loss. We find that the test loss is far flatter in the directions of leading eigenvectors of the pre-trained covariance than in a random direction, indicating the learned prior indeed promotes directions consistent with the downstream task (See Appendix C)

Learned Covariance Outperforms Only a Learned Mean. Next, we directly compare the performance benefits of our learned covariance over an isotropic covariance with a learned mean. We show that a learned covariance consistently outperforms its isotropic counterpart, indicating that the shape of the pre-training loss surface's basin is informative for downstream tasks (Figure 3c in the appendix).

How is the Prior Best Structured? To understand the benefits of a low-rank component for capturing particularly important directions in the pre-training loss surface for transferring to downstream tasks, we check the performance of a learned prior for different ranks. We found that As the rank of the low-rank component increases from zero (diagonal covariance), the performance improves dramatically until it saturates, indicating that only a small number of dominant directions are important for the transferability of the prior (Figure 3b in the appendix).

.

3.2. Re-scaling the Prior

In transfer learning, we assume the data from the source and downstream task are drawn from *different* but related distributions. Thus, we would like to re-scale the source's posterior when we use it as a prior. To do so, we re-scale the learned Gaussian prior by multiplying its covariance matrix by a scalar value. In Figure 3d in the appendix, we explore the effect of the scaling factor. We show that, indeed, there is an optimal scaling factor where the prior is neither overly

diffuse nor concentrated around a solution inconsistent with the downstream task. Additionally, small scaling factors constrain us to poor parameters, hurting performance much more than large scaling factors, which induce a near-uniform prior, negating the benefits of transfer learning.

3.3. Bayesian Inference

After learning a prior and re-scaling it, we finally must draw samples from the downstream task's posterior over the parameters of the entire model, including additional modules, such as classification heads, which were added after pre-training specifically for a particular downstream task. We use a zero-mean isotropic Gaussian prior over these additional parameters, with a scaling that is again tuned on held-out training data from the downstream task. In our experiments, we choose stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014) and Stochastic Gradient Langevin Dynamics (SGLD) (Welling and Teh, 2011) since these samplers simultaneously provide strong performance and tractable computation costs.

In Figure 3c in the appendix, we also observe the advantage of Bayesian inference (SGHMC) over MAP estimation (SGD) on the negative log-posterior yielded by our learned prior. Bayesian inference outperforms MAP estimation across all dataset sizes we consider.

4. Experiments

We now conduct a thorough empirical evaluation on image classification and semantic segmentation. We generally consider five approaches: **(1)** Bayesian inference with learned priors, **(2)** SGD with learned priors, **(3)** SGD with standard pre-training, **(4)** Bayesian inference with non-learned zero-mean priors, **(5)** SGD with non-learned zero-mean priors.

We adopt the ResNet-50 and ResNet-101 architectures (He et al., 2016) and use a SimCLR (SSL) ResNet-50 checkpoint (Chen et al., 2020) pre-trained on the ImageNet dataset (Deng et al., 2009). For the supervised setting, we use pre-trained `torchvision` ResNet-50 and ResNet-101 backbones. We perform image classification experiments on four downstream tasks: CIFAR-10, CIFAR-100 (Krizhevsky, 2009), Oxford Flowers-102 (Nilsback and Zisserman, 2008), and Oxford-IIIT Pets (Parkhi et al., 2012). On semantic segmentation, we use a DeepLabv3+ system (Chen et al., 2018) with ResNet-50 and ResNet-101 backbone architectures, and we evaluate performance on the Pascal VOC 2012 (Everingham et al., 2010) and Cityscapes (Cordts et al., 2016) datasets. We provide a detailed description in Appendix F.1.

4.1. Performance Comparison

In Figure 1, we compare the five approaches described above across various dataset sizes. We observe the following: (i) learned priors consistently outperform SGD transfer learning which in turn outperforms non-learned priors; (ii) Using the same prior, Bayesian inference often outperforms SGD training; (iii) Bayesian inference adds more value when used with informative priors; (iv) learned priors are relatively most valuable on intermediate downstream data sizes. Point (iv) is particularly interesting: unlike standard SGD transfer learning, which involves an initialization from the pre-training task, learned priors provide an explicit inductive bias, enabling the resulting model to learn more efficiently from downstream data. Once there is a sufficient amount of downstream data, the source pre-training becomes less important, and the methods become more similar in performance, although there is still a significant gap.

Comparison between self-supervised and supervised pre-training. In Appendix E.2 we provide additional evaluations with priors generated using supervised checkpoints, which are learned with labeled data. We see here also that learned supervised priors paired with Bayesian inference consistently outperform all baselines. Notably, we also see that the SSL priors are more transferable and outperform their supervised counterparts.

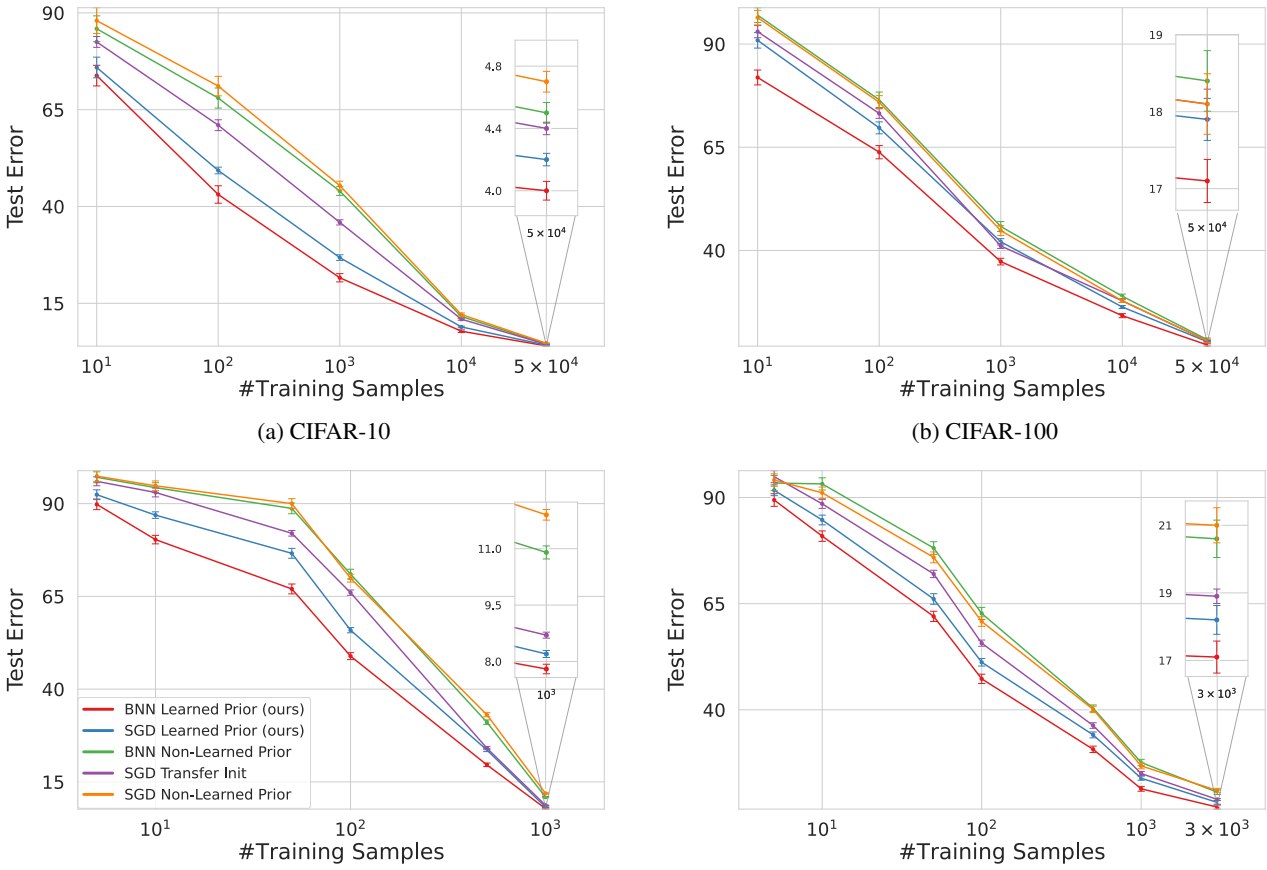
Evaluating uncertainty. We also measure predictive uncertainty via negative log test-likelihood (NLL) (See Appendix E.4). We find that Bayesian transfer learning outperforms all other methods. However, we note that even though Bayesian inference with a non-learned prior has inferior accuracy to SGD with pre-training, it has superior test-likelihood — indicating that the confidence of SGD-based transfer learners is significantly miscalibrated. We also evaluate the calibration of uncertainty using reliability diagrams (Figure 8 in the Appendix). These plots demonstrate that Bayesian inference with learned priors is the best calibrated among the methods we consider. .

4.2. Out-of-Distribution Generalization

CIFAR-10.1 (Recht et al., 2018) is a test set of 2000 natural images modeled after CIFAR-10 with the same classes and image dimensions, following a similar dataset creation process described in the original CIFAR-10 paper. Models trained on CIFAR-10 consistently perform much worse on CIFAR-10.1 despite the images being easy for humans to classify. We find that our method achieves superior performance to SGD-based transfer learning and Bayesian inference with non-learned priors on CIFAR-10.1 across training set sizes, where training sets are sampled from CIFAR-10 training data (Figure 6a).

4.3. Semantic Segmentation with ImageNet Priors

Popular segmentation methods use ImageNet pre-trained weights as initialization for backbone parameters (Chen et al., 2018). We observe that semantic segmentation mod-



(a) CIFAR-10

(b) CIFAR-100

(c) Oxford Flowers-102

(d) Oxford-IIIT Pets

Figure 1: **Performance comparison.** Learned priors outperform non-learned priors, and Bayesian inference generally outperforms SGD training, especially with learned priors. Moreover, learned priors tend to be more data-efficient. We use a ResNet-50 and SimCLR for pre-trained priors. The horizontal axis denotes the number of downstream training samples on a logarithmic scale. Error bars correspond to a standard error over five trials.

els, which contain numerous parameters aside from the backbone, still benefit immensely from a learned prior over backbone parameters. Placing a strong prior, even if over only the backbone component of the segmentation system, enables us to more fully harness pre-training and boost performance across benchmark datasets PASCAL VOC 2012 and Cityscapes.

In Table 1, we see that Bayesian inference with learned priors achieves superior Mean-IoU to both baselines (SGD and SGLD) without learned priors. Priors learned on the SimCLR objective again outperform those learned on labeled data. We also present experiments with MAP estimates (SGD) for the loss functions induced by both learned priors in Appendix F.2. We again find that our learned priors boost the performance of MAP estimates as well and thus are preferable to standard transfer learning, even for practitioners who do not intend to perform Bayesian inference.

Dataset	Backbone	SGD Transfer	Non-Learned Prior	Learned Prior Supervised	Learned Prior SSL
PASCAL VOC 2012	ResNet-50	73.17	73.16	73.72	**74.15**
	ResNet-101	75.53	75.79	**76.27**	*
Cityscapes	ResNet-50	73.17	75.52	76.09	**76.52**
	ResNet-101	77.04	77.02	**77.63**	*

Table 1: **Bayesian inference for semantic segmentation.** The performance (Mean-IoU) of SGLD with learned and non-learned priors over ResNet-50 and ResNet-101 for segmentation tasks. *SimCLR ResNet-101 checkpoints have not been released.

5. Discussion

In this work, we fuse the benefits of transfer learning and Bayesian neural networks. We propose a framework for collecting prior knowledge from a pre-training task and utilizing it throughout the downstream process, not only for initialization. Our pipeline uses powerful model checkpoints pre-trained on a large dataset and does not require

pre-training from scratch. Our Bayesian transfer learning paradigm can provide higher accuracy across classification benchmark tasks than SGD-based transfer learning and standard BNNs.

References

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Rohitash Chandra and Arpit Kapoor. Bayesian neural multi-source transfer learning. *Neurocomputing*, 378:54–64, 2020.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Kien Do, Truyen Tran, and Svetha Venkatesh. Semi-supervised learning with variational bayesian inference and maximum uncertainty regularization. *arXiv preprint arXiv:2012.01793*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425*, 2019.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010.

Vincent Fortuin, Adrià Garriga-Alonso, Florian Wenzel, Gunnar Rätsch, Richard Turner, Mark van der Wilk, and Laurence Aitchison. Bayesian neural network priors revisited. *arXiv preprint arXiv:2102.06571*, 2021.

Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, pages 905–914. PMLR, 2020.

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*, pages 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

W Ronny Huang, Zeyad Emam, Micah Goldblum, Liam Fowl, Justin K Terry, Furong Huang, and Tom Goldstein. Understanding generalization through visualizations. *"I Can't Believe It's Not Better!" NeurIPS 2020 workshop*, 2020.

Pavel Izmailov, Patrick Nicholson, Sanae Lotfi, and Andrew Gordon Wilson. Dangers of bayesian model

averaging under covariate shift. *arXiv preprint arXiv:2106.11905*, 2021a.

Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like? *arXiv preprint arXiv:2104.14421*, 2021b.

Neal Jean, Sang Michael Xie, and Stefano Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *Advances in Neural Information Processing Systems*, 31, 2018.

Sanyam Kapoor, Theofanis Karaletsos, and Thang D Bui. Variational auto-regressive gaussian processes for continual learning. In *International Conference on Machine Learning*, pages 5290–5300. PMLR, 2021.

Alireza Karbalayghareh, Xiaoning Qian, and Edward R Dougherty. Optimal bayesian transfer learning. *IEEE Transactions on Signal Processing*, 66(14):3724–3739, 2018.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*, 2017.

Wesley Maddox, Shuai Tang, Pablo Moreno, Andrew Gordon Wilson, and Andreas Damianou. Fast adaptation with linearized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 2737–2745. PMLR, 2021.

Wesley J Maddox, Pavel Izmailov, Timur Garipov, Dmitry P Vetrov, and Andrew Gordon Wilson. A simple baseline for bayesian uncertainty in deep learning. *Advances in Neural Information Processing Systems*, 32:13153–13164, 2019.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

Pingbo Pan, Siddharth Swaroop, Alexander Immer, Runa Eschenhagen, Richard Turner, and Mohammad Emtiyaz E Khan. Continual deep learning by functional regularisation of memorable past. *Advances in Neural Information Processing Systems*, 33:4453–4464, 2020.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.

Massimiliano Patacchiola, Jack Turner, Elliot J Crowley, Michael O'Boyle, and Amos J Storkey. Bayesian meta-learning for the few-shot setting via deep kernels. *Advances in Neural Information Processing Systems*, 33: 16108–16118, 2020.

Zoe Piran, Ravid Shwartz-Ziv, and Naftali Tishby. The dual information bottleneck. *arXiv preprint arXiv:2006.04641*, 2020.

Naresh Balaji Ravichandran, Anders Lansner, and Pawel Herman. Semi-supervised learning with bayesian confidence propagation neural network. *arXiv preprint arXiv:2106.15546*, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

Ravid Shwartz-Ziv. Information flow in deep neural networks. *arXiv preprint arXiv:2202.06749*, 2022.

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)*, pages 681–688. Citeseer, 2011.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *arXiv preprint arXiv:2002.08791*, 2020.

Junyu Xuan, Jie Lu, and Guangquan Zhang. Bayesian transfer learning: An overview of probabilistic graphical models for transfer learning. *arXiv preprint arXiv:2109.13233*, 2021.

Ruqi Zhang, Chunyuan Li, Jianyi Zhang, Changyou Chen, and Andrew Gordon Wilson. Cyclical stochastic gradient mcmc for bayesian deep learning. *arXiv preprint arXiv:1902.03932*, 2019.

A. More Background

Leveraging Auxiliary Data in Bayesian Modeling. A number of works outside of deep learning have considered knowledge transfer in Bayesian modeling, especially in settings such as domain adaptation or homogeneous transfer learning in which source and target tasks contain similar feature and label spaces, but differ in their marginal distributions. For example, Xuan et al. (2021) considers methods for probabilistic graphical models, and Karbalayghareh et al. (2018) develop a theoretical framework to understand optimal Bayes classifiers given prior knowledge from other domains.

Other works develop Bayesian tools for leveraging auxiliary data or multiple domains in deep learning. Chandra and Kapoor (2020) learn from multiple domains simultaneously using a round-robin task sampling procedure and a single-layer neural network. Bayesian methods for continual learning update the posterior to accommodate new tasks without forgetting how to perform previous ones (Nguyen et al., 2017; Ebrahimi et al., 2019; Kapoor et al., 2021), or develop kernels based on neural networks trained on previous tasks for Gaussian process inference (Maddox et al., 2021; Pan et al., 2020). Semi-supervised algorithms can incorporate unlabeled data into BNN training pipelines using biologically plausible Bayesian Confidence Propagation Neural Networks (BCPNN), which model the cortex (Ravichandran et al., 2021), by perturbing weights and using consistency regularization (Do et al., 2020), or by semi-supervised deep kernel learning (Jean et al., 2018). Deep kernel learning has also been used for Bayesian meta-learning in few-shot classification and regression (Patacchiola et al., 2020).

In contrast to these works, we do not perform multi-task learning nor is our goal to harness auxiliary unlabeled downstream data; we instead approach transfer learning in which we leverage pre-training data for expressive priors that maximize performance on a single downstream task.

B. Computational Considerations

Learning the prior is a one-time cost that incurs a cost of about 30 epochs of pre-training. On the downstream task, SGD with the learned prior costs the same as standard transfer learning. We use 5 SGLD and SGHMC chains, such that Bayesian inference costs about $5\times$ standard training. We see both SGD and MCMC with pre-trained priors significantly outperform standard transfer learning. At test time, SGD with a learned prior has the same cost as standard transfer learning. We use 10 posterior samples for MCMC based Bayesian inference, incurring a $10\times$ test-time cost. We see that MCMC significantly outperforms deep ensembles, which have the same train and test-time costs.

C. Bayesian Transfer Learning with Pre-Trained Priors

The experiments in this section are primarily intended to gain conceptual insight into each step of our approach. For experiments in this section, we use a prior learned over the parameters of an ImageNet pre-trained SimCLR ResNet-50 feature extractor (Deng et al., 2009; Chen et al., 2020; He et al., 2016), and we choose CIFAR-10 and CIFAR-100 for downstream tasks (Krizhevsky, 2009). We perform Bayesian inference with stochastic gradient Hamiltonian Monte Carlo (SGHMC) (Chen et al., 2014).

C.1. Learning Transferable Priors

Alignment of Pre-Training and Downstream Loss Geometry. We compare the alignment of the learned covariance matrix with the test loss of a downstream task.

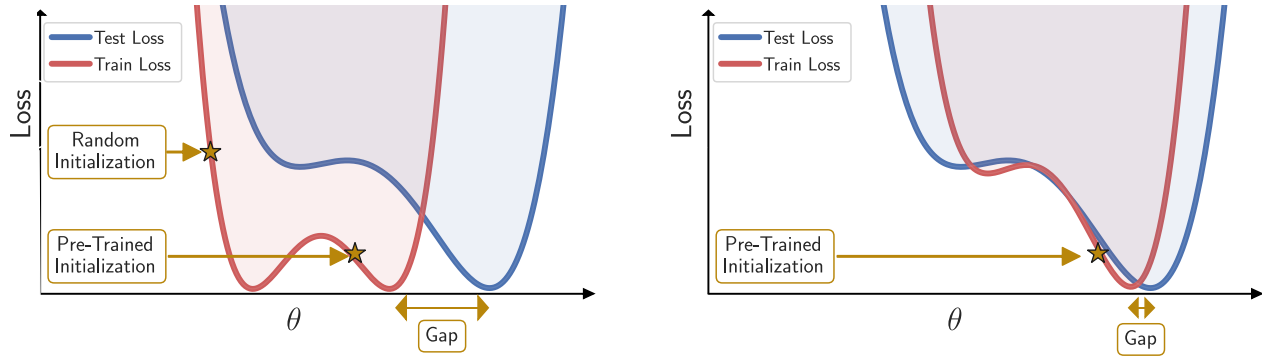
We begin by computing the top 5 leading singular vectors of a SWAG learned covariance matrix over parameters of the SimCLR ImageNet-trained ResNet-50 feature extractor. We then train a linear classifier head on CIFAR-10 training data on top of the fixed pre-trained feature extractor. Starting at the learned parameter vector, we perturb the feature extractor parameters in the direction of the singular vectors (fixing fully-connected classifier parameters), and measure the increase in test loss. We compare to the test loss when instead perturbing by each of 10 random vectors. All perturbation distances are filter normalized (as in (Li et al., 2017; Huang et al., 2020)), to account for invariance with respect to filter-wise parameter rescaling. In Figure 3a, we see the CIFAR-10 test loss is far flatter in the directions of leading eigenvectors of the pre-trained covariance than in a random direction, indicating the learned prior indeed promotes directions consistent with the downstream task.

C.2. Re-scaling the Prior

In transfer learning, we assume that the data from the source and downstream task are drawn from *different* but related distributions. Thus, we would like to re-scale the posterior from the source when we you it as a prior.

To address this consideration, we re-scale the learned Gaussian prior by multiplying its covariance matrix by a scalar value. We select the highest performing scalar value across a grid on a holdout set from the downstream task. If we do not scale the covariance enough, our prior will become concentrated around parameters which are inconsistent with the downstream task, and if we scale the covariance too much, our prior will be diffuse and assign too much mass to regions of parameter space which are again inconsistent with the downstream task.

We now put this intuition to the test. We fit the SimCLR



(a) Standard Transfer Learning

(b) Transfer Learning with Learned Priors

Figure 2: **Learned priors reshape the loss surface.** Red curves represent training loss and blue curves represent the true population loss. The typical transfer learning approach avoids poor-generalizing minima by improving initializations (left). The learned priors further improve the alignment between training and test loss, improving accuracy (right). The gap between the learned and optimal parameters is denoted by a double-sided gold arrow.

pre-training loss using a Gaussian prior with mean μ and covariance matrix Σ. Given our uncertainty regarding the strength of the relationship between the pre-training and downstream tasks, it is unclear if our learned prior would lead to over-confidence on the downstream task. To rectify this problem, we instead assign prior covariance matrix $\lambda\Sigma$ with $\lambda \geq 1$. Figure 3d shows the accuracy of our method on CIFAR-10 as a function of λ. As expected, we see that we need to make the prior more diffuse if we are to optimize performance. In fact, prior re-scaling can be the difference between strong and poor performance. We also see that there is an optimal scaling factor where the prior is neither overly diffuse nor concentrated around a solution, which is inconsistent with the downstream task. Additionally, small scaling factors constrain us to poor parameters, hurting performance much more than large scaling factors, which simply induce a near-uniform prior, negating the benefits of transfer learning.

D. Experimental Setting

We adopt the ResNet-50 and ResNet-101 architectures (He et al., 2016), and we scale the input image to 224×224 pixels to accommodate these feature extractors designed for ImageNet data. We use a SimCLR (SSL) ResNet-50 checkpoint (Chen et al., 2020) pre-trained on the ImageNet 1k dataset (Deng et al., 2009) and fit our prior to the SimCLR loss function. For the supervised setting, we use pre-trained `torchvision` ResNet-50 and ResNet-101 backbones. We perform image classification experiments on four downstream tasks: CIFAR-10, CIFAR-100 (Krizhevsky, 2009), Oxford Flowers-102 (Nilsback and Zisserman, 2008), and Oxford-IIIT Pets (Parkhi et al., 2012). On semantic segmentation, we use a DeepLabv3+ system (Chen et al., 2018) with ResNet-50 and ResNet-101 backbone architectures, and we evaluate performance on the Pascal VOC 2012 (Everingham et al., 2010) and Cityscapes (Cordts et al., 2016) datasets. All error bars represent one standard error over 5 runs. We evaluate over a variety of downstream training set sizes, as transfer learning often involves limited downstream data.

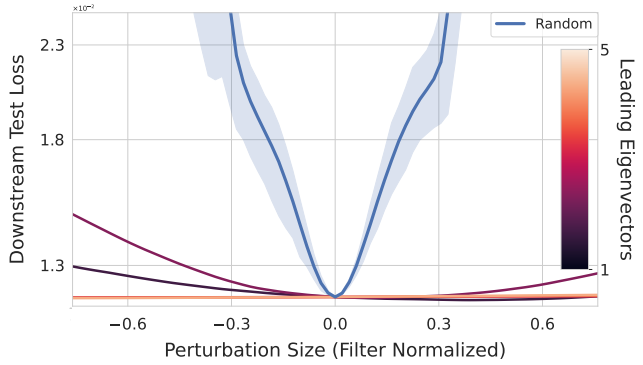
E. Classification

E.1. Implementation Details

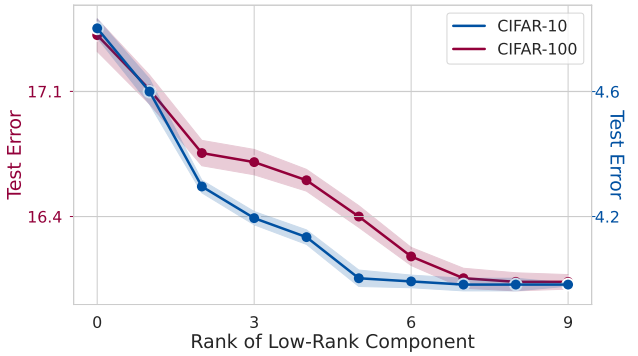
Based on the evaluation protocols in the papers introducing the datasets, we report the top-one accuracy for CIFAR-10 and CIFAR-100 and mean per-class accuracy for Oxford-IIIT Pets and Oxford 102 Flowers. For Oxford 102 Flowers, we select hyperparameters based on the validation sets specified by the dataset creators. While tuning the hyperparameters, we hold out a subset of the training set for validation on the other datasets. After selecting the optimal hyperparameters from the validation set, we retrain the model using the selected parameters on both the training and the validation images. The test results are reported.

We use ResNet-50 and ResNet-101 architectures (He et al., 2016) with supervised and SSL checkpoints (Chen et al., 2020) pre-trained on the ImageNet 1k dataset (Deng et al., 2009). We use the same hyperparameters as in Chen et al. (2020).

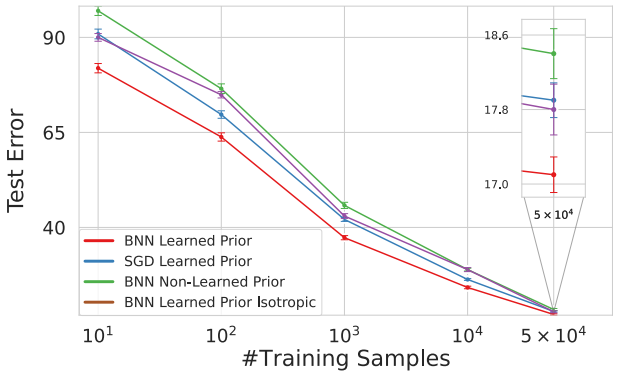
Learning the prior. In order to learn the prior, we use the SWAG algorithm (Maddox et al., 2019) with the cyclic learning rate presented in Zhang et al. (2019). We use SGD with a Nesterov momentum parameter of 0.9 for optimization. We select our initial learning rate from a logarithmic



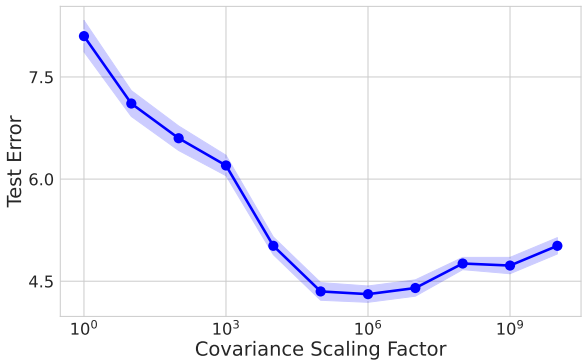
(a) Pre-training and downstream losses are aligned



(b) Optimal rank of low-rank component



(c) Importance of a learned covariance



(d) Optimal prior scaling factor

Figure 3: **Learning Transferable Priors.** Error bars corresponds to standard error over 5 trials. All experiments were performed with ResNet-50 backbone pre-trained via SimCLR on ImageNet. **(a)** Test loss (CIFAR-10) is less sensitive to perturbations in the directions of leading singular vectors of a learned prior's covariance compared to random directions. Estimated over 10 random directions. **(b)** A non-diagonal covariance is important, though performance on downstream tasks plateaus at a relatively low rank. CIFAR-100 benefits from a higher rank covariance than CIFAR-10, due to its relative complexity. **(c)** A non-diagonal covariance outperforms an isotropic covariance with learned mean on CIFAR-100. Bayesian inference provides further performance gains. **(d)** We must re-scale the posterior from the source task, since data from each task is drawn from a different distribution. If λ is too small, then we assume too much similarity between tasks. If λ is too large, we lose useful information from the source task. Downstream inference is performed on CIFAR-10.

grid between 0.005 and 0.5 and evaluate 10000 iterations for each cycle. Other hyperparameters are from Chen et al. (2020), including data augmentation which comprises color augmentation, blurring, random crops, and horizontal flips Shwartz-Ziv (2022).

Downstream classification tasks. We train for $30,000$ steps with a batch size of 128 on CIFAR-10 and CIFAR-100, 16 for Oxford Flowers-102 and 32 for Oxford-IIIT Pets. SGD and SGHMC are used with a momentum parameter of 0.9. During fine-tuning, we perform random crops with resizing to 224×224 and horizontal flips. At test time, we resize the images to 256 pixels along the shorter side and produce a 224×224 center crop. In our study, we select the learning rate and weight decay from a grid of 7 logarithmically spaced learning rates between 0.0001 and 0.1, and 7 logarithmically spaced weight decay values between $1e^{-6}$ and $1e^{-2}$, as well as without weight decay. These weight-decay values are divided by the learning rate. For the SGHMC optimizer, the temperature is selected from a logarithmic grid of 8 values between $1e^{0}$ and $1e^{-8}$, and we use predictions from five different chains for the final evaluation.

E.2. Supervised Pre-Training

As mentioned above, we conduct additional experiments with learned priors on labeled data. Figures 5 and 4 contain downstream task performance comparisons for the Resnet-50 and Resnet-101 models with learned priors starting with

`torchvision` checkpoints pre-trained on ImageNet 1k. These figures compare our method to SGD transfer learning with pre-trained initializations and Bayesian inference with non-learned Gaussian prior. Across both backbones, our method outperforms all baselines for nearly every number of train samples and for all four datasets, where the Bayesian inference model with a non-learned prior is always worse. (For full results, see tables 2, 3, 4 and 5.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	75.7	46.8	24.9	9.2	4.3
BNN Non-learned Prior	86.5	68.1	44.9	11.6	4.8
SGD Ensemble	78.7	57.2	31.6	10.9	5.2
SGD Transfer Init	81.9	62.1	36.5	11.6	4.4

Table 2: CIFAR-10 test error with `torchvision` Resnet-50 prior/initialization.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	86.8	64.9	38.8	26.1	17.2
BNN Non-learned Prior	97.0	83.6	50.8	30.1	19.2
SGD Ensemble	93.3	74.6	43.1	26.4	17.4
SGD Transfer Init	94.9	71.8	42.7	26.9	17.8

Table 3: CIFAR-100 test error with `torchvision` Resnet-50 prior/initialization.

Model/# samples	5	10	50	100	500	1000
BNN Learned Prior	90.5	88.0	77.2	52.8	20.4	8.5
BNN Non-learned Prior	97.1	94.3	88.7	71.0	35.0	10.7
SGD Ensemble	96.2	90.1	86.1	60.0	24.0	8.9
SGD Transfer Init	96.6	94.3	85.4	68.0	26.3	9.6

Table 4: Oxford-Flowers-102 test error with `torchvision` Resnet-50 prior/initialization.

E.3. Self-Supervised Pre-Training

In this section, we present additional data to Figures 1 and 6 found in the main body. The tables 10, 11, 12 and 13 are corresponding to Figure 1, while Table 14 corresponding to Figure 6.

Computation and comparison to deep ensembles. Our Bayesian model average (BMA) contains 10 samples from the posterior, and thus incurs a higher test-time cost than a single SGD-trained model. We therefore additionally compare to an equally sized ensemble of SGD-trained transfer learning models in Appendix E.3. We see that our BMA strongly outperforms the deep ensemble. In transfer learning, deep ensemble members are initialized with the same value from pre-training on the source task, and fine-tuning tends to stay in the same basin, such that the ensemble components are relatively homogenous in their predictions. Recall that SGD with our learned priors incurs essentially the same computational costs as standard transfer learning, but has much better performance.

Model/# samples	5	10	50	100	500	1000	3000
BNN Learned Prior	98.2	84.0	64.7	49.3	32.0	24.6	17.6
BNN Non-learned Prior	93.4	93.2	78.1	62.7	45.4	38.6	25.2
SGD Ensemble	96.6	86.3	72.1	55.0	37.9	26.6	19.3
SGD Transfer Init	94.5	89.9	75.4	61.1	36.1	25.9	18.6

Table 5: Oxford-IIIT Pets test error with `torchvision` Resnet-50 prior/initialization.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	73.9	43.9	20.4	7.2	3.7
BNN Non-learned Prior	82.1	68.7	36.3	16.4	4.1
SGD Transfer Init	78.6	56.8	30.8	8.9	4.3

Table 6: CIFAR-10 test error with `torchvision` Resnet-101 prior/initialization.

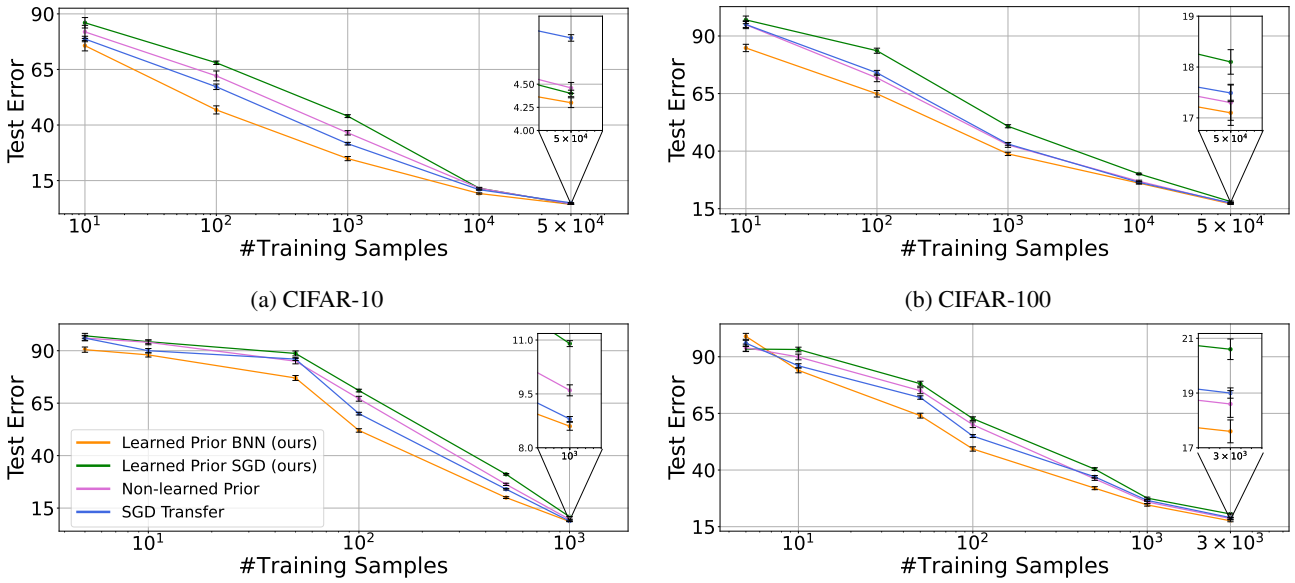
E.4. Evaluating uncertainty

In Figure 7, we present the negative log test-likelihood (NLL) for both the Bayesian methods and the SGD-based methods for 4 datasets (CIFAR-10, CIFAR-100, Oxford Flowers-102 and Oxford-IIIT Pets). In general, we can see a similar trend for all the datasets where the Bayesian transfer learning outperforms all other methods. In Figure 8 we present the reliability diagrams for CIFAR-10 and CIFAR100 for the ReSnet-18 network. A perfectly calibrated network has no difference between accuracy and confidence, represented by a dashed black line Piran et al. (2020). Under-confident predictions are those below this line, whereas overconfident predictions are those above. We can see that Bayesian inference with learned priors is the best calibrated among the methods we consider. The error bars are computed on 5 runs and the radius is a standard error.

F. Semantic Segmentation

F.1. Implementation Details

For our semantic segmentation evaluations, we use the DeepLabv3+ (Chen et al., 2018) model with ResNet-50 and ResNet-101 backbone architectures. We train our models using Pascal VOC 2012 and Cityscapes trainsets and evaluate on their *val* sets. We utilize the same hyperparameters employed in Chen et al. (2018) with a few modifications: our learning rate schedule is the *poly* policy (Chen et al., 2017) with initial value 0.01 for Pascal VOC 2012 and 0.1 for Cityscapes, our crop size is 513×513 for Pascal VOC 2012 and 768×768 for Cityscapes. Our *output stride* is 16 for both training and evaluating, and we perform random scale data augmentation during training. In our SGD and SGLD optimization, we employ Nesterov momentum, and we set the momentum coefficient to 0.9. We use batches of 16 images and weight decay with a coefficient of 10^{-4}. In addition, for the SGLD optimizer, the temperature is set to



(a) CIFAR-10

(b) CIFAR-100

(c) Oxford Flowers-102

(d) Oxford-IIIT Pets

Figure 4: **Performance comparison.** The test error of a ResNet-50 BNN equipped with a supervised pre-trained prior (red) is consistently lower than that of standard SGD-based transfer learning from the same pre-trained checkpoint (yellow), an equivalently sized ensemble of SGD-based transfer learning models (blue), or BNN with a mean zero isotropic Gaussian prior (green). The x-axis denotes the number of downstream training samples. Axes on a logarithmic scale.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	81.2	61.9	35.1	23.5	16.8
BNN Non-learned Prior	93.5	79.6	58.8	32.0	20.2
SGD Transfer Init	94.4	71.1	44.2	25.9	17.0

Table 7: CIFAR-100 test error with `torchvision` Resnet-101 prior/initialization.

Model/# samples	5	10	50	100	500	1000
BNN Learned Prior	88.0	81.0	74.6	50.7	18.9	7.1
BNN Non-learned Prior	96.0	92.8	88.6	75.1	41.1	21.7
SGD Transfer Init	95.0	90.2	80.8	59.2	22.3	8.6

Table 8: Oxford-Flowers-102 test error with `torchvision` Resnet-101 prior/initialization.

2×10^{-8}. We train all models for 30k iterations.

F.2. MAP Estimation

We also compare MAP estimates (SGD) on the loss induced by our learned priors to the baseline with only weight decay. Table 15 and 16 contain the results using ResNet-50 and ResNet-101 backbones, respectively. On both architectures and both datasets, learned priors boost performance on these experiments without using Bayesian inference at all.

Model/# samples	5	10	50	100	500	1000	3000
BNN Learned Prior	96.1	82.4	62.9	48.6	30.3	23.9	17.2
BNN Non-learned Prior	96.4	91.1	78.6	67.7	56.9	38.2	26.2
SGD Transfer Init	94.0	88.6	74.3	59.2	35.5	25.0	18.1

Table 9: Oxford-IIIT Pets test error with `torchvision` Resnet-101 prior/initialization.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	73.8 ± 3.2	43.1 ± 3.1	21.6 ± 1.6	7.8 ± 0.5	4.0 ± 0.1
SGD Learned Prior	75.9 ± 2.1	49.3 ± 1.9	26.8 ± 1.1	8.9 ± 0.3	4.2 ± 0.1
BNN Non-learned Prior	86.1 ± 3.9	68.0 ± 3.8	44.9 ± 1.7	11.6 ± 0.6	4.4 ± 0.1
SGD Ensemble	81.3 ± 2.2	57.0 ± 3.1	32.3 ± 1.4	9.4 ± 0.5	4.6 ± 0.1
SGD Transfer Init	83.6 ± 1.8	61.0 ± 2.1	36.8 ± 1.0	10.9 ± 0.4	4.4 ± 0.1

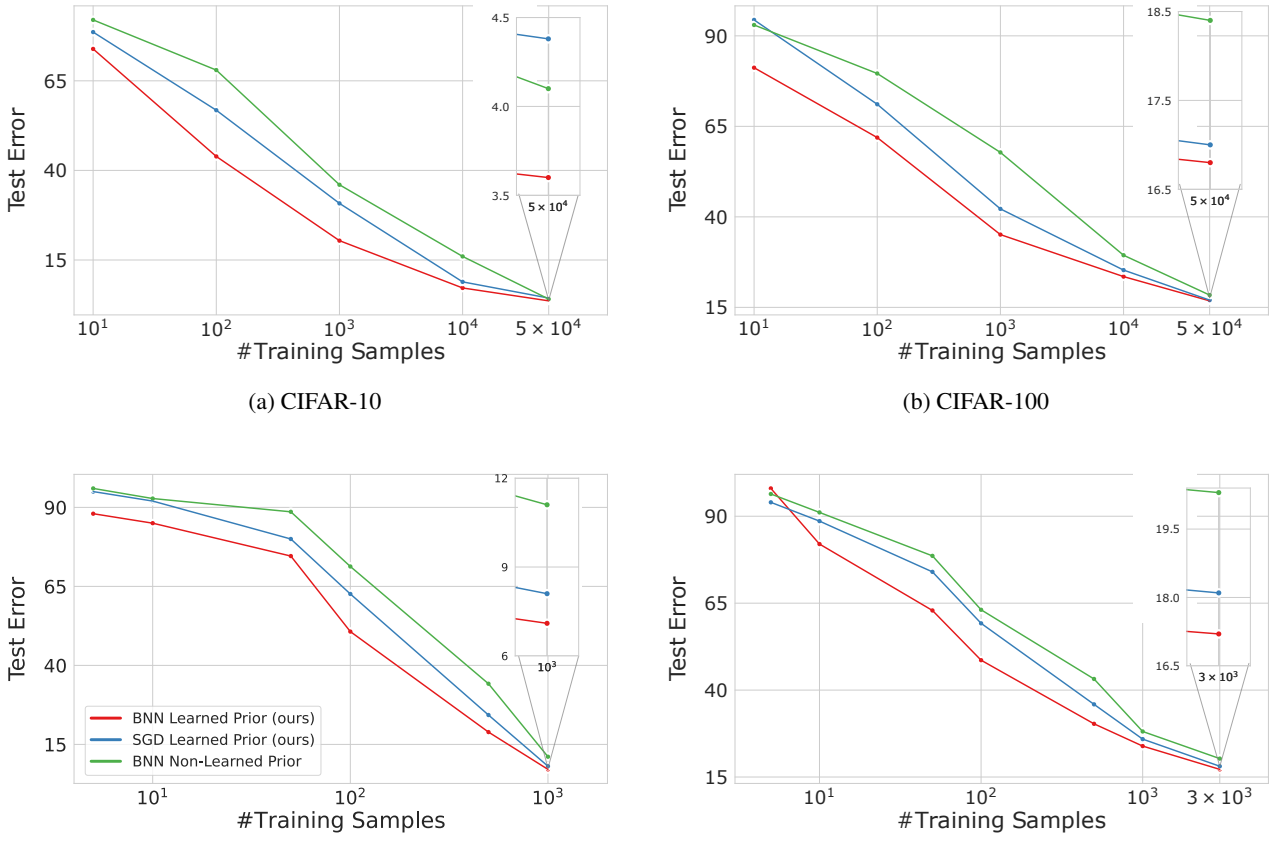
Table 10: CIFAR-10 test error with SimCLR (SSL) Resnet-50 prior/initialization.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	81.9 ± 3.7	63.8 ± 2.4	37.3 ± 1.2	24.2 ± 0.7	16.4 ± 0.4
SGD Learned Prior	90.9 ± 2.8	69.7 ± 1.9	41.1 ± 1.1	26.3 ± 0.5	17.3 ± 0.4
BNN Non-learned Prior	97.0 ± 3.9	84.0 ± 2.8	50.8 ± 1.8	28.9 ± 0.8	18.8 ± 0.6
SGD Ensemble	89.2 ± 2.0	70.9 ± 1.7	44.0 ± 1.2	26.6 ± 0.7	17.2 ± 0.4
SGD Transfer Init	93.0 ± 2.2	73.2 ± 1.8	45.5 ± 0.9	27.8 ± 0.6	17.9 ± 0.3

Table 11: CIFAR-100 test error with SimCLR (SSL) Resnet-50 prior/initialization.

Model/# samples	5	10	50	100	500	1000
BNN Learned Prior	89.8 ± 3.1	80.3 ± 1.7	67.0 ± 1.8	48.9 ± 1.4	19.6 ± 0.7	7.8 ± 0.2
SGD Learned Prior	90.2 ± 2.0	81.0 ± 1.3	67.0 ± 1.1	49.9 ± 1.1	20.6 ± 0.6	8.2 ± 0.1
BNN Non-learned Prior	97.1 ± 2.7	94.3 ± 2.0	88.7 ± 2.1	71.0 ± 2.0	35.0 ± 0.9	10.9 ± 0.3
SGD Ensemble	93.8 ± 2.0	91.2 ± 1.6	84.3 ± 1.2	56.8 ± 1.0	21.9 ± 0.3	8.5 ± 0.2
SGD Transfer Init	96.0 ± 1.8	93.0 ± 1.8	82.0 ± 1.3	66.0 ± 1.1	24.1 ± 0.6	8.7 ± 0.1

Table 12: Oxford-Flowers-102 test error with SimCLR (SSL) Resnet-50 prior/initialization.



(a) CIFAR-10

(b) CIFAR-100

(c) Oxford Flowers-102

(d) Oxford-IIIT Pets

Figure 5: **Performance comparison.** The test error of a ResNet-101 BNN equipped with supervised pre-trained prior (red) is consistently lower than that of standard SGD-based transfer learning from the same pre-trained checkpoint (yellow), or BNN with a mean zero isotropic Gaussian prior (green). The x-axis denotes the number of downstream training samples. Axes on a logarithmic scale.

Model/# samples	5	10	50	100	500	1000	3000
BNN Learned Prior	89.4 ± 2.6	80.9 ± 1.9	62.0 ± 1.8	47.3 ± 1.6	30.7 ± 1.1	21.4 ± 0.9	17.1 ± 0.7
SGD Learned Prior	91.7 ± 1.9	84.7 ± 1.7	66.1 ± 1.9	51.2 ± 1.3	34.1 ± 0.8	23.9 ± 0.6	18.2 ± 0.6
BNN NL Prior	93.4 ± 2.9	93.2 ± 2.1	78.1 ± 2.2	62.7 ± 2.1	45.4 ± 1.1	38.6 ± 1.1	25.2 ± 0.8
SGD Ensemble	92.2 ± 2.1	88.1 ± 1.4	68.3 ± 2.1	55.7 ± 1.5	35.0 ± 1.2	24.5 ± 1.0	18.1 ± 0.5
SGD Transfer Init	94.9 ± 2.1	88.5 ± 1.6	72.0 ± 1.3	55.7 ± 1.1	36.3 ± 1.0	25.0 ± 0.7	18.9 ± 0.3

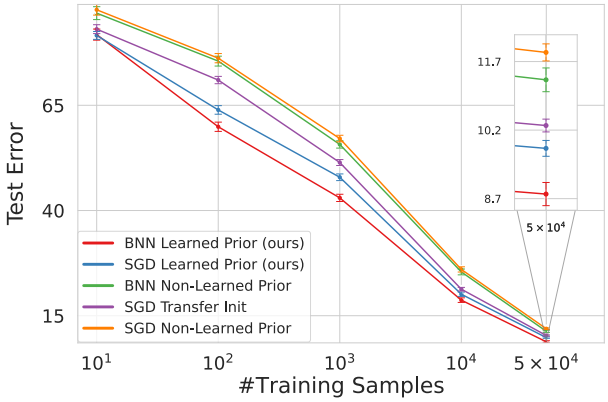
Table 13: Oxford-IIIT Pets test error with SimCLR (SSL) Resnet-50 prior/initialization.

Model/# samples	10	100	1000	10000	50000
BNN Learned Prior	81.8 ± 2.0	59.9 ± 1.7	43.0 ± 1.3	18.7 ± 0.8	8.8 ± 0.4
SGD Learned Prior	81.6 ± 1.4	63.9 ± 1.5	48.0 ± 1.2	20.1 ± 0.8	9.8 ± 0.3
BNN Non-learned Prior	86.9 ± 2.3	78.5 ± 1.8	59.0 ± 1.3	32.0 ± 1.0	11.3 ± 0.4
SGD Transfer Init	83.1 ± 1.6	69.0 ± 1.3	52.0 ± 1.1	21.3 ± 0.6	10.3 ± 0.2

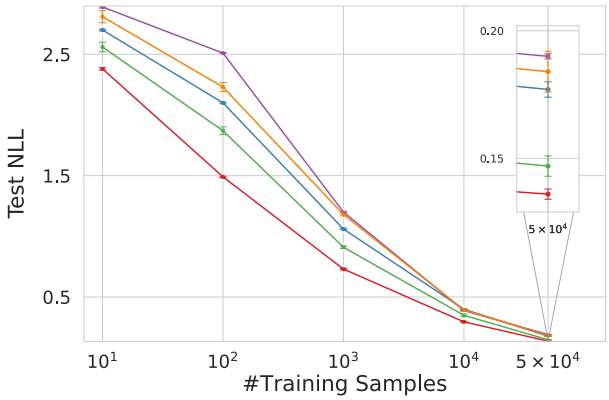
Table 14: Classification error with self-supervised pre-learning - CIFAR-10.1

Table 15: **MAP estimation for semantic segmentation (ResNet-50 backbone).** Comparing the performance (Mean-IoU) of SGD with learned priors (both `torchvision` supervised and SimCLR SSL) to a non-learned prior (weight decay with `torchvision` initialization) over DeepLabv3+ backbone parameters for downstream segmentation with SGD rather than Bayesian inference. Evaluations are conducted on *val* sets.

Dataset	Non-Learned Prior MAP	Supervised Prior MAP	SSL Prior MAP
PASCAL VOC 2012	73.17	73.27	**73.48**
Cityscapes	75.52	75.57	**76.15**

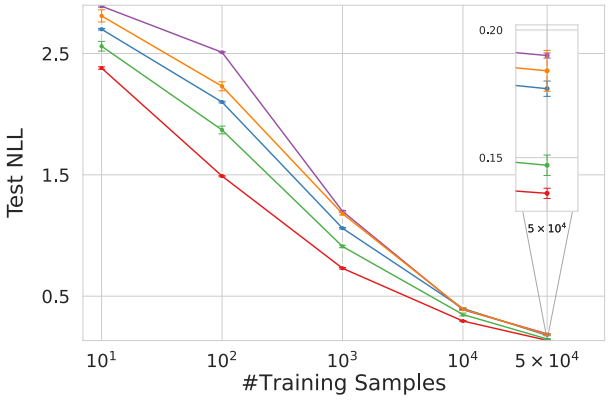


(a) Distributional shift on CIFAR-10.1.

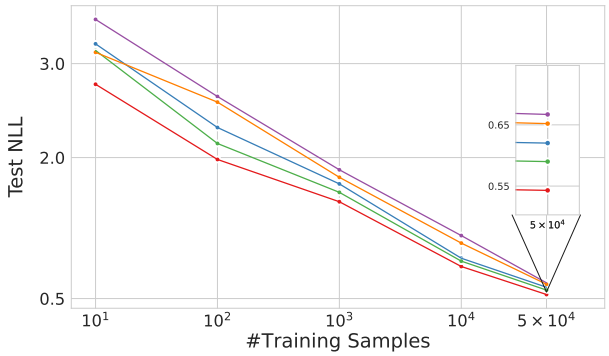


(b) Test Negative Log-Likelihood on CIFAR-10.

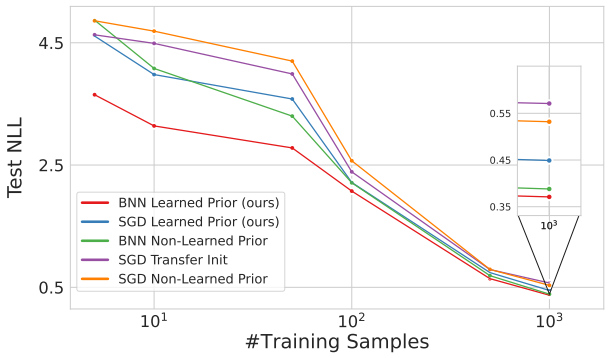
Figure 6: **Predictive likelihood and distribution shift.** The horizontal axis denotes the number of downstream CIFAR-10 training samples on a logarithmic scale. The error bars correspond to the standard error on 5 trials. All experiments were performed with ResNet-50 backbone pre-trained via SimCLR on ImageNet. **(a)** BNNs with our learned priors achieve similarly superior accuracy in OOD testing. **(b)** Bayesian inference with a learned prior achieves superior test NLL, but BNNs with a non-learned prior outperforms all other methods despite often performing worst in terms of accuracy.



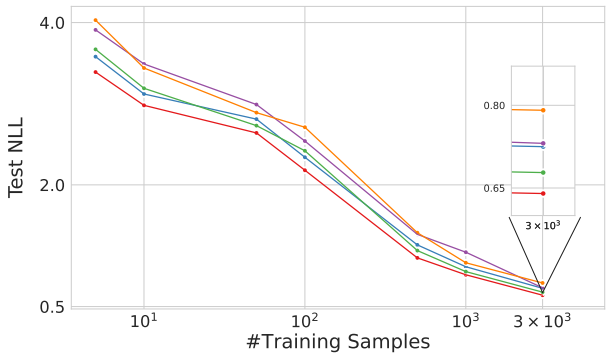
(a) CIFAR-10



(b) CIFAR-100

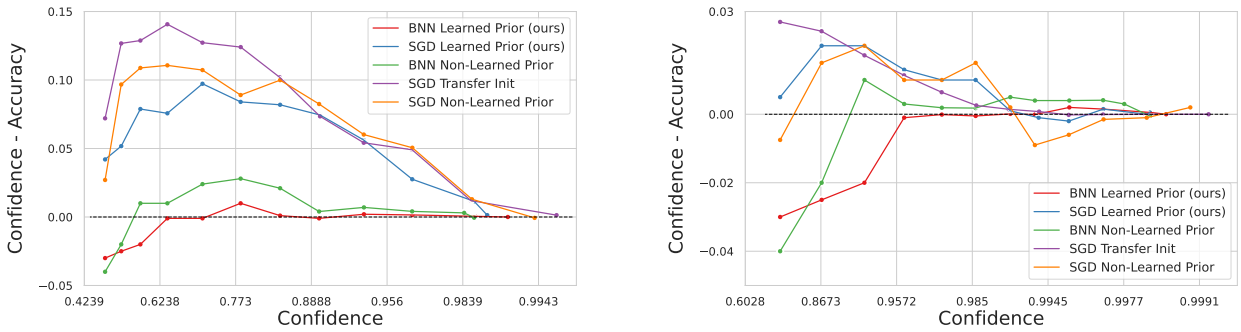


(c) Oxford Flowers-102



(d) Oxford-IIIT Pets

Figure 7: **Predictive likelihood.** The horizontal axis denotes the number of downstream CIFAR-10 training samples on a logarithmic scale. All experiments performed with ResNet-50 backbone.



(a) CIFAR-100

(b) CIFAR-10

Figure 8: **Reliability diagrams** - Bayesian transfer learning is capable of significantly improving calibration over standard transfer training (SGD Init) as well as non-learned prior BNN. The error bars are computed on 5 runs.

Table 16: **MAP estimation for semantic segmentation (ResNet-101 backbone).** Comparing the performance (Mean-IoU) of SGD with `torchvision` learned priors to a non-learned prior (weight decay with `torchvision` initialization) over DeepLabv3+ backbone parameters for downstream segmentation with SGD rather than Bayesian inference. Evaluations are conducted on *val* sets.

Dataset	Non-Learned Prior MAP	Supervised Prior MAP
PASCAL VOC 2012	75.53	**75.89**
Cityscapes	77.04	**77.27**