Incorporating Self-Supervised Learning into Deep Bayesian Active Learning

Candidate Number: 1072250

1 Introduction

Deep learning (DL) has achieved strong performance in computer vision, yet its reliance on large labelled datasets limits applicability in domains where annotation is costly or requires expert knowledge. *Active learning* (AL) mitigates this issue by selecting the most informative unlabelled samples for annotation, reducing labelling effort [5]. However, classical AL struggles with high-dimensional image data and deep models, largely due to poor uncertainty estimation under data scarcity [13].

The paper *Deep Bayesian Active Learning with Image Data* addressed these challenges by combining Bayesian convolutional neural networks with principled acquisition functions, enabling uncertainty-aware AL for image data. [9]. This project first replicates their key findings to validate the effectiveness of Bayesian DL in AL framework. Then, we extend their work in two directions. First, we designed a new baseline using a Bayesian neural network with both analytical and approximate matrix normal inference. Second, we incorporate self-supervised learning (SSL) with rotation prediction on unlabelled pool data before AL, using pretrained weights as prior distribution mean. We demonstrate that SSL learnt useful unlabelled prior representation by improving the baselines. Our novel extension synergises two effective methods, AL and SSL, to address the problem of limited labels in Bayesian settings, an intersection that is currently under-explored.

2 Backgrounds

Active learning (AL) reduces labelling cost by iteratively selecting informative unlabelled samples using an acquisition function [5, 12]. Early AL methods focused on uncertainty or margin-based sampling in low-dimensional models [15, 10], but these approaches scale poorly to high-dimensional data due to unreliable uncertainty estimation in deep models [13]. Bayesian deep learning offers a robust solution by modelling predictive uncertainty. For image data, [6, 8] showed that Bayesian convolutional neural networks can be efficiently implemented via Monte Carlo dropout. Building on this, [9] combined BCNNs with acquisition functions such as BALD, achieving significant gains in annotation efficiency on vision benchmarks. Later work improved uncertainty estimation using batch acquisition strategies [11] and Bayesian ensemble methods [16].

Meanwhile, self-supervised learning (SSL) exploits unlabelled data through pretext tasks such as transformation prediction and contrastive learning [1]. Recent studies have begun integrating SSL with Bayesian models [17] and active learning pipelines [14]. These advances motivate combining Bayesian uncertainty estimation with self-supervised pretraining to further reduce labelling requirements in high-dimensional vision tasks.

3 Methodology: *Deep Bayesian Active Learning with Image Data*

Gal et al. formulate AL for image classification using Bayesian convolutional neural networks (BCNNs) [9]. A prior distribution $p(\boldsymbol{\omega})$ is placed over the L convolutional weight matrices $\boldsymbol{\omega} = \{\mathbf{W}_1, \ldots, \mathbf{W}_L\}$, with a categorical likelihood $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\omega}) = \mathrm{softmax}(f^{\boldsymbol{\omega}}(\mathbf{x}))$ where $f^{\boldsymbol{\omega}}(\mathbf{x}) \in \mathbb{R}^C$. As exact posterior is intractable, an approximate *Bernoulli* variational inference (VI) is used: $p(\boldsymbol{\omega} \mid \mathbf{X}, \mathbf{Y}) \approx q(\mathbf{W}_i) := \mathbf{M}_i \cdot \mathrm{diag}([z_{i,j}]_{j=1}^{K_i})$ where $z_{i,j} \sim \mathrm{Bernoulli}(1 - p_i)$ for $i = 1, ..., L$, $j =$

$1, ..., K_{i-1}$, K_l the size of the l-th filter, and $\mathbf{M}_i$ are variational parameters [6]. Approximate Bernoulli VI is equivalent to implementing MC dropout, applied at both training and test time. The predictive distribution is obtained by averaging predictions across multiple stochastic forward passes by MC integration: $p(y = c \mid \mathbf{x}, \mathbf{X}, \mathbf{Y}) \approx \frac{1}{T}\sum_{t=1}^{T} p(y = c \mid \mathbf{x}, \hat{\boldsymbol{\omega}}_t)$, where $\hat{\boldsymbol{\omega}}_t \sim q(\boldsymbol{\omega})$ are sampled weights after dropout. Maximising the evidence lower bound (ELBO) is equivalent to training the model under cross entropy loss with L2 regularisation on weights [7].

AL proceeds iteratively by training the BCNN on the current labelled set, evaluating acquisition functions on the unlabelled pool using MC estimates, and retrieving the highest-scoring samples for annotation. The paper considers 5 acquisition functions. We show a detailed derivation of their MC estimates in Appendix B. We reproduce experiments on the MNIST classification benchmark.

4 Mandatory Extensions

We implement a Bayesian neural network (BNN) with inference only on the last weight layer as a new baseline. We place a matrix normal prior $p(\mathbf{W}) = \mathcal{MN}_{K\times C}(\mathbf{0},\, s^{-1}I_K,\, \Sigma)$, where $\Sigma \in \mathbb{R}^{C\times C}$ captures output correlations and s controls prior strength. Using a Gaussian likelihood $p(\mathbf{y} \mid \mathbf{x}, \mathbf{W}) = \mathcal{N}(\mathbf{y}; f^{\mathbf{W}}(\mathbf{x}), \Sigma)$, where $f^{\mathbf{W}}(\mathbf{x}) = \mathbf{W}^\top \varphi(\mathbf{x}) \in \mathbb{R}^C$ and φ is frozen feature extractor, the dataset likelihood factorises as $p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) = \mathcal{MN}_{N\times C}(\Phi(\mathbf{X})\mathbf{W},\, I_N,\, \Sigma)$.

Analytical Posterior. By *Corollary 1*, the posterior has a close-form distribution given by $p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y}) = \mathcal{MN}_{K\times C}(M_{\text{post}},\, U_{\text{post}},\, V_{\text{post}})$, where

$$V_{\text{post}} = \Sigma, \quad U_{\text{post}} = (sI_K + \Phi(\mathbf{X})^\top \Phi(\mathbf{X}))^{-1}, \quad M_{\text{post}} = U_{\text{post}}\Phi(\mathbf{X})^\top \mathbf{Y}. \tag{1}$$

By *Proposition 2*, the predictive distribution is given by $p(\mathbf{y}^* \mid \mathbf{x}^*, \mathbf{X},\, \mathbf{Y}) = \mathcal{N}(\mathbf{y}^* \,; \mu^*,\, \Sigma^*)$ where

$$\mu^* = M_{\text{post}}^\top \varphi(\mathbf{x}^*), \quad \Sigma^* = \big(1 + \varphi(\mathbf{x}^*)^\top U_{\text{post}}\, \varphi(\mathbf{x}^*)\big)\Sigma. \tag{2}$$

We use *predictive covariance* as the acquisition function. Since Σ^* is a scalar multiple of Σ, acquisition reduces to ranking by the scalar $q = \varphi(\mathbf{x}^*)^\top U_{\text{post}} \varphi(\mathbf{x}^*)$, equivalent to maximising log determinant $\log|\Sigma^*|$ which is large for informative unlabelled data [2].

Approximate VI. An alternative way is to approximate the posterior with a variational matrix normal distribution $p(\mathbf{Y} \mid \mathbf{X}, \mathbf{W}) \approx q(\mathbf{W}) := \mathcal{MN}_{K\times C}\big(M_q,\, U_q,\, V_q\big)$, and maximise a variational evidence lower bound (ELBO) given by *Proposition 3*:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(M_q, U_q, V_q) = -\tfrac{1}{2}\,\text{tr}\Big[\Sigma^{-1}\Big((Y - \Phi M_q)^\top (Y - \Phi M_q) + \big(\text{tr}(U_q \Phi^\top \Phi)\big)\, V_q \\ + \; s\big(\text{tr}(U_q)\, V_q + M_q^\top M_q\big)\Big)\Big] \\ + \tfrac{1}{2}\Big[KC(1 + \log(2\pi)) + C\log|U_q| + K\log|V_q|\Big] \; + \; \text{const.}\end{aligned} \tag{3}$$

where we wrote $\Phi = \Phi(\mathbf{X})$. Optimisation is performed by minimising $-\mathcal{L}_{\text{ELBO}}$, with several approaches to ensure numerical stability detailed in Appendix A.4.

The predictive mean and covariance mirror the analytical case in (2), where $\mu^* = M_q^\top \varphi(\mathbf{x}^*)$, $\Sigma^* = \Sigma + \varphi(\mathbf{x}^*)^\top U_q\, \varphi(\mathbf{x}^*) V_q$ (same derivation as Proposition 2). We use $\log|\Sigma^*|$ for acquisition.

For both inference methods, we treat MNIST as a regression task in $\mathbb{R}^{10}$, and evaluate models using RMSE between model outputs $\hat{\mathbf{y}} = f^{\mathbf{W}}(\mathbf{x})$ and one-hot target vectors $\mathbf{y}$.

5 Novel Extensions: SSL with Deep Bayesian Active Learning

Given the immense potential for SSL to learn meaningful prior from unlabelled data [1], we extend the methodology in Section 3 and 4 by pretraining their baseline models with a rotation prediction task. We hypothesise that by using the pretrained weights as the *mean of prior distributions*, deep Bayesian models can infer more informative posterior and predictive, therefore improving AL efficiency.

Rotation Prediction Pretraining: We augment the unlabelled dataset with 4 arbitrary rotations from $[0°, 90°, 180°, 270°]$. For the BCNN model (Section 3), we replace the final FC layer with a temporary 4-way head, and pretrain it deterministically. The pretrained weights $\mathbf{W}_{\text{SSL}_i}$ for $i =$

$1, ..., L-1$ are assigned to the mean of Gaussian prior: $p(\mathbf{W}_i) = \mathcal{N}(\mathbf{W}_{\text{SSL}_i}, l^2 I)$ where l is a hyperparameter. The original head is reinstalled and initialised. For BNN model (Section 4), a similar rotation head is used. After pretraining, a linear ridge regression between learnt features and one-hot outputs $\mathbf{W}_{\text{SSL}} = (\Phi^T\Phi + \alpha I_K)^{-1}\Phi^T\mathbf{Y}$ is fit to give he prior of the final weight matrix as $\mathcal{MN}_{K\times C}(\mathbf{W}_{\text{SSL}},\ s^{-1}I_K,\ \Sigma)$.

BNN models: conjugacy is preserved and the posterior remains matrix normal with closed-form solution given by *Proposition 1*: $p(\mathbf{W} \mid \mathbf{X}, \mathbf{Y}) = \mathcal{MN}_{K\times C}\big(M_{\text{post}},\ U_{\text{post}},\ V_{\text{post}}\big)$, where

$$V_{\text{post}} = \Sigma, \quad U_{\text{post}} = \big(sI_K + \Phi(\mathbf{X})^\top\Phi(\mathbf{X})\big)^{-1}, \quad M_{\text{post}} = U_{\text{post}}\big(s\mathbf{W}_{\text{SSL}} + \Phi(\mathbf{X})^\top\mathbf{Y}\big). \tag{4}$$

The predictive covariance retains the same form as in the non-SSL case and is used for acquisition.

BCNN models: the non-zero prior mean leads to a modified ELBO. As proven in Proposition 4, to maximising ELBO is now equivalent to minimising a cross-entropy loss with L2 regularisation anchored at the SSL weights $\mathbf{W}_{\text{SSL}}$:

$$\mathcal{L}(\boldsymbol{\omega}) = \frac{1}{N}\sum_{n=1}^{N}\text{CE}\left(f^{\boldsymbol{\omega}}(\mathbf{x}_n), \mathbf{y}_n\right) + \sum_{i=1}^{L}\frac{\lambda_i}{2}||\mathbf{W}_i - \mathbf{W}_{\text{SSL}_i}||^2 \tag{5}$$

where $\lambda_i = \frac{1-p_i}{Nl^2}$ aligns with the original paper [9]. All other aspects in AL remain unchanged. We have thus shown a natural way to incorporate SSL prior into deep Bayesian AL workflow.

6 Experiments

Experiments are conducted to (i) reproduce Sections 5.1–5.2 of [9], (ii) compare analytical and variational inference for the BNN last-layer model (Section 4), and (iii) evaluate the effect of self-supervised learning (SSL) on both BNN and BCNN baselines (Section 5). All experiments use MNIST (60k/10k train/test). To reduce acquisition cost, acquisition functions are evaluated on a uniformly subsampled pool of 2,000 unlabelled points. A validation set of 100 samples is used for hyperparameter tuning. BCNNs are trained for classification, while BNNs for regression in $\mathbb{R}^{10}$.

All models share fixed architectures. The BCNN matches the same configurations as [9]. The BNN uses two hidden layers of sizes 512 and 256, with ReLU activations and dropout ($p = 0.5$) for regularisation only. Active learning follows the original protocol: starting from 20 labelled samples, acquiring 10 points per iteration over 100 iterations.

Reproducing paper: *Deep Bayesian Active Learning with Image Data* We replicate Section 5.1 by comparing five acquisition functions over three runs. Results (Figure 1, Table 1) largely match the original paper, except for mean standard deviation (MSD), which in our experiments outperforms random sampling. Inspection of acquisition score distributions (Figure 3) suggests MSD produces highly non-uniform scores, indicating behaviour distinct from random selection. Section 5.2 is reproduced by evaluating deterministic CNNs (no MC dropout at test time); results (Figure 2) closely match the original findings, confirming the importance of epistemic uncertainty in AL. All results are averaged across 3 experimental runs. ± 1 sd from mean accuracy is shaded where appropriate.

Comparing BNN Inference Methods We compare analytical posterior inference against variational inference (VI) using RMSE. Analytical models are trained for 50 epochs (lr 10^{-3}), while VI models optimise the ELBO for 800 epochs (lr 5×10^{-3}). As shown in Figure 4, VI converges with slightly higher RMSE possibly due to optimisation error, while closely tracking the analytical solution as we expect them to converge to the same error. Results show averaged 5 experimental runs.

Comparing SSL Effects on BNN and BCNN Baselines Both BNN and BCNN models are pretrained on rotation prediction for 5 epochs (lr 10^{-3}), and SSL weights are used as prior means at each AL iteration. We evaluate BALD for BCNNs and predictive covariance for BNNs for acquisitions, and average results over 5 experimental runs. Results (Figures 5a and 5b) show consistent improvements with SSL across both baselines. For BNNs, SSL reduces RMSE at larger label sizes, mitigating underfitting. For BCNNs, SSL improves early-stage accuracy, demonstrating that SSL learnt useful prior from rotation prediction to enhance posterior inference and AL efficiency.

7 Conclusion

This project successfully reproduces most results in *Deep Bayesian Active Learning with Image Data*, validating the effectiveness of Bayesian models in active learning (AL) framework and the importance of uncertainty-driven acquisition functions. We further developed a Bayesian neural network baseline and demonstrate expected alignments between analytical and variational posterior approximations under a generic matrix normal formulation. Most notably, our novel extension to integrate self-supervised learning (SSL) with deep Bayesian AL showed further improvements to the baseline, providing a practical mechanism to incorporate prior information learnt from unlabelled data and reducing annotation costs. With more time, future work may explore alternative SSL objectives, such as SimCLR-based contrastive learning [4] or image restoration autoencoders [3], to further strengthen prior representations. The synergy between SSL, Bayesian deep learning and AL offers significant promise for effective learning under limited labelled data regimes.

Code Availability and Declaration

PyTorch implementation of our experiments can be found in the GitHub repository:

`https://github.com/lukyllukanoate/UDL-Mini-Project.git`

Models were trained and evaluated using a T4 GPU on Google Colab. Codes were written independently and no external repositories were referenced.

References

[1] Mohammed Majid Abdulrazzaq et al. Bayesian deep learning and a probabilistic perspective of generalization. *Mathematics*, 12(5):758, 2024.

[2] T. Tony Cai, Tengyuan Liang, and Harrison H. Zhou. Law of log determinant of sample covariance matrix and optimal estimation of differential entropy for high-dimensional gaussian distributions. *arXiv preprint arXiv:1309.0482*, 2013.

[3] Liang Chen, Paul Bentley, Kensaku Mori, Kazunari Misawa, Michitaka Fujiwara, and Daniel Rueckert. Self-supervised learning for medical image analysis using image context restoration. *Medical Image Analysis*, 58:101539, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. *Journal of Artificial Intelligence Research*, 4:129–145, 1996.

[6] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015.

[7] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Insights and applications. *arXiv preprint arXiv:1506.02142*, 2015.

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016.

[9] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR, Aug 2017.

[10] Jose Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1861–1869. PMLR, 2015.

[11] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7026–7037, 2019.

[12] Simon J. D. Prince. A tutorial on active learning. *Technical Report, Department of Computer Science, University College London*, 2004.

[13] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *arXiv preprint arXiv:2009.00236*, 2020.

[14] Mrinank Sharma, Tom Rainforth, Yee Whye Teh, and Vincent Fortuin. Incorporating unlabelled data into bayesian neural networks. *arXiv preprint arXiv:2304.01762*, 2023.

[15] Simon Tong. *Active Learning: Theory and Applications*. PhD thesis, Stanford University, Stanford, CA, USA, 2001. Ph.D. thesis, AAI3028187.

[16] Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, volume 33 of *Advances in Neural Information Processing Systems*, 2020.

[17] Yan-Xue Wu, Fan Min, Gong-Suo Chen, Shao-Peng Shen, Zuo-Cheng Wen, and Xiang-Bing Zhou. Self-supervised class-balanced active learning with uncertainty-mastery fusion. *Knowledge-Based Systems*, 300:112192, 2024.

A Derivations for Mandatory Extension

In this section, we detail our derivation for the close-form analytical posterior, and mean field variational inference (MFVI) solution for the posterior of the last layer weights in a Bayesian neural network (hierarchical parametrised basis function). **Our derivation is more general than the minimal requirement**, as it is also applicable to our novel extensions in Section 5.

We first give the definition of *matrix normal distributions* and some important properties that are used for our derivation.

Definition 1. *Let $\mathbf{X} \in \mathbb{R}^{n\times p}$ have the matrix-normal distribution with mean $\mathbf{M} \in \mathbb{R}^{n\times p}$ and row/column covariances $\mathbf{U} \in \mathbb{R}^{n\times n}$, $\mathbf{V} \in \mathbb{R}^{p\times p}$, denoted by $\mathcal{MN}_{n\times p}\big(\mathbf{M},\ \mathbf{U},\ \mathbf{V}\big)$. The probability density function is*

$$p(\mathbf{X} \mid \mathbf{M}, \mathbf{U}, \mathbf{V}) = \frac{\exp\big(-\frac{1}{2}\operatorname{tr}\left[\mathbf{V}^{-1}(\mathbf{X}-\mathbf{M})^T\mathbf{U}^{-1}(\mathbf{X}-\mathbf{M})\right]\big)}{(2\pi)^{\frac{np}{2}}\,|\mathbf{V}|^{\frac{n}{2}}\,|\mathbf{U}|^{\frac{p}{2}}}. \tag{6}$$

Lemma 1. *The standard matrix normal identities involving expectations are stated below*[1]*: If $\mathbf{X} \sim \mathcal{MN}_{n\times p}(\mathbf{M}, \mathbf{U}, \mathbf{V})$, then for appropriately dimensioned matrices $\mathbf{A}, \mathbf{B}$:*

$$\mathbb{E}\big[\mathbf{X}^\top\,\mathbf{A}\,\mathbf{X}\big] = \mathbf{V}\operatorname{tr}\big(\mathbf{U}\,\mathbf{A}^\top\big) + \mathbf{M}^\top\,\mathbf{A}\,\mathbf{M} \tag{7}$$

$$\mathbb{E}\big[\mathbf{X}\,\mathbf{B}\,\mathbf{X}^\top\big] = \mathbf{U}\operatorname{tr}\big(\mathbf{B}^\top\mathbf{V}\big) + \mathbf{M}\,\mathbf{B}\,\mathbf{M}^\top \tag{8}$$

A.1 Analytical Posterior Derivation

Proposition 1. *Let $W \in \mathbb{R}^{K\times C}$. Assume the prior distribution*

$$p(W) = \mathcal{MN}_{K\times C}\big(W \mid M_0,\ sI_K,\ \Sigma\big),$$

and the likelihood shares the same row covariance

$$p(Y \mid X, W) = \mathcal{MN}_{N\times C}\big(Y \mid \Phi(X)W,\ I_N,\ \Sigma\big).$$

Then the posterior distribution is given by

$$p(W \mid X, Y) = \mathcal{MN}_{K\times C}\big(M_{\text{post}},\ U_{\text{post}},\ \Sigma\big),$$

where

$$U_{\text{post}} = \big(sI_K + \Phi(X)^\top\Phi(X)\big)^{-1}, \qquad M_{\textit{post}} = U_{\text{post}}\big(sM_0 + \Phi(X)^\top Y\big).$$

Proof. For notational simplicity, denote $\Phi := \Phi(X)$. Up to proportionality constants, the prior density according to Equation (6) is

$$p(W) \propto \exp\left(-\frac{1}{2}\operatorname{tr}\big[\Sigma^{-1}(W-M_0)^\top(sI_K)(W-M_0)\big]\right).$$

Similarly, the likelihood satisfies

$$p(Y \mid X, W) \propto \exp\left(-\frac{1}{2}\operatorname{tr}\Big[\Sigma^{-1}\big(Y-\Phi W\big)^\top\big(Y-\Phi W\big)\Big]\right).$$

The posterior, following Bayes' Rules, is therefore

$$\begin{aligned} p(W \mid X, Y) &\propto p(Y \mid X, W)\cdot p(W) \\ &\propto \exp\left(-\frac{1}{2}\operatorname{tr}\Big(\Sigma^{-1}\Big[s(W-M_0)^\top(W-M_0) + \big(Y-\Phi W\big)^\top\big(Y-\Phi W\big)\Big]\Big)\right) \end{aligned}$$

Expanding and grouping in terms of W, the trace term simplifies to

$$\operatorname{tr}\Big(\Sigma^{-1}\Big[W^\top(sI_K+\Phi^\top\Phi)W - W^\top(sM_0+\Phi^\top Y) - (sM_0^\top + Y^\top\Phi)W + Y^\top Y + M_0^\top M_0\Big]\Big). \tag{9}$$

[1] Source: `https://en.wikipedia.org/wiki/Matrix_normal_distribution`

We now assume the posterior takes the conjugate matrix normal form

$$W \mid X, Y \sim \mathcal{MN}_{K\times C}\big(M_{\text{post}},\ U_{\text{post}},\ V_{\text{post}}\big),$$

whose density according to Equation (6) is proportional to

$$\exp\Big(-\frac{1}{2}\operatorname{tr}\Big(V_{\text{post}}^{-1}(W - M_{\text{post}})^\top U_{\text{post}}^{-1}(W - M_{\text{post}})\Big)\Big).$$

Expanding the quadratic form, the trace term becomes

$$\operatorname{tr}\Big(V_{\text{post}}^{-1}\big[W^\top U_{\text{post}}^{-1} W - W^\top U_{\text{post}}^{-1} M_{\text{post}} - M_{\text{post}}^\top U_{\text{post}}^{-1} W + M_{\text{post}}^\top U_{\text{post}}^{-1} M_{\text{post}}\big]\Big). \tag{10}$$

Comparing terms between Equation (9) and Equation (10), we choose

$$V_{\text{post}}^{-1} = \Sigma^{-1} \quad \Longrightarrow \quad V_{\text{post}} = \Sigma.$$

Matching the quadratic terms in W, we obtain

$$U_{\text{post}}^{-1} = sI_K + \Phi^\top \Phi \quad \Longrightarrow \quad U_{\text{post}} = \big(sI_K + \Phi^\top \Phi\big)^{-1}.$$

Matching the linear terms gives

$$U_{\text{post}}^{-1} M_{\text{post}} = sM_0 + \Phi^\top Y,$$

which implies

$$M_{\text{post}} = U_{\text{post}}(sM_0 + \Phi^\top Y) = \big(sI_K + \Phi^\top \Phi\big)^{-1}(sM_0 + \Phi^\top Y).$$

This completes the derivation. □

Corollary 1. *Let $W \in \mathbb{R}^{K\times C}$. Assume the prior distribution has zero mean*

$$p(W) = \mathcal{MN}_{K\times C}\big(W \mid \mathbf{0},\ sI_K,\ \Sigma\big),$$

and the likelihood is the same as Proposition 2, then the posterior is given by

$$p(W \mid X, Y) = \mathcal{MN}_{K\times C}\big(U_{\text{post}}\Phi(X)^\top Y,\ U_{\text{post}},\ \Sigma\big),$$

where $U_{\text{post}} = \big(sI_K + \Phi(X)^\top \Phi(X)\big)^{-1}$.

Proof. From Proposition 2, setting $M_0 = \mathbf{0}$ gives the result. □

A.2 Predictive Distribution from Matrix Normal Conjugate

Proposition 2. *Let the posterior distribution of the weight matrix be*

$$p(W \mid X, Y) = \mathcal{MN}_{K\times C}\big(M_{\text{post}},\ U_{\text{post}},\ \Sigma\big),$$

and assume the likelihood

$$p(y^* \mid x^*, W) = \mathcal{N}\big(W^\top \phi(x^*),\ \Sigma\big).$$

Then the predictive distribution satisfies

$$p(y^* \mid x^*, X, Y) = \mathcal{N}\big(\mu^*,\ \Sigma^*\big),$$

where

$$\mu^* = M_{\text{post}}^\top \phi(x^*), \qquad \Sigma^* = \big(1 + \phi(x^*)^\top U_{\text{post}} \phi(x^*)\big)\Sigma.$$

Proof. For notational simplicity, denote $\phi^* := \phi(x^*)$.

We compute the predictive mean and covariance by moment matching.

Predictive mean. By definition,

$$\mu^* = \mathbb{E}_{p(y^*|x^*,X,Y)}[y^*] = \int p(y^* \mid x^*, X, Y)\, y^*\, dy^*.$$

Using marginalisation over W,

$$p(y^* \mid x^*, X, Y) = \int p(y^* \mid x^*, W)\, p(W \mid X, Y)\, dW,$$

which gives (by interchanging integral on y^* and W):

$$\mu^* = \int \left(\int p(y^* \mid x^*, W)\, y^*\, dy^* \right) p(W \mid X, Y)\, dW.$$

From the Gaussian likelihood,

$$\mathbb{E}_{p(y^*|x^*,W)}[y^*] = W^\top \phi^*.$$

Therefore,

$$\mu^* = \int W^\top \phi^*\, p(W \mid X, Y)\, dW = \left(\int W^\top p(W \mid X, Y)\, dW \right) \phi^*.$$

Taking expectation under the matrix normal posterior,

$$\mathbb{E}_{p(W|X,Y)}[W] = M_{\text{post}},$$

hence

$$\mu^* = M_{\text{post}}^\top \phi^*.$$

Predictive covariance. We compute

$$\text{Var}[y^*] = \mathbb{E}[y^* y^{*\top}] - \mu^* \mu^{*\top}.$$

First,

$$\begin{aligned} \mathbb{E}[y^* y^{*\top}] &= \int p(y^* \mid x^*, X, Y)\, y^* y^{*\top}\, dy^* \\ &= \int \left(\int p(y^* \mid x^*, W)\, y^* y^{*\top}\, dy^* \right) p(W \mid X, Y)\, dW. \end{aligned}$$

From the Gaussian likelihood,

$$\mathbb{E}_{p(y^*|x^*,W)}[y^* y^{*\top}] = \Sigma + W^\top \phi^* \phi^{*\top} W.$$

Thus,

$$\mathbb{E}[y^* y^{*\top}] = \Sigma + \int W^\top \phi^* \phi^{*\top} W\, p(W \mid X, Y)\, dW.$$

Using the standard matrix normal identity from Identity (7):

$$\mathbb{E}[W^\top A W] = \Sigma\, \text{tr}(U_{\text{post}} A) + M_{\text{post}}^\top A M_{\text{post}},$$

with $A = \phi^* \phi^{*\top}$, and noticing $\text{tr}(U_{\text{post}} \phi^* \phi^{*\top}) = \text{tr}(\phi^{*\top} U_{\text{post}} \phi^*)$ has scalar inside trace, we obtain

$$\mathbb{E}[y^* y^{*\top}] = \Sigma + \phi^{*\top} U_{\text{post}} \phi^*\, \Sigma + M_{\text{post}}^\top \phi^* \phi^{*\top} M_{\text{post}}.$$

Subtracting

$$\mu^* \mu^{*\top} = M_{\text{post}}^\top \phi^* \phi^{*\top} M_{\text{post}},$$

we conclude

$$\text{Var}[y^*] = \Sigma + \phi^{*\top} U_{\text{post}} \phi^*\, \Sigma = \big(1 + \phi^{*\top} U_{\text{post}} \phi^*\big) \Sigma.$$

This completes the derivation. □

A.3 Approximate Matrix Normal Variational Inference

In this section, we provide the derivation of variational ELBO objective that fullfill the mean field variational inference (MFVI) requirements for posterior. We provide the full derivation in *matrix normal* setting with correlation among outputs.

Proposition 3. *Assume the same prior / likelihood setup as before with a zero-mean prior on the last-layer weight matrix* $W \in \mathbb{R}^{K\times C}$*:*

$$p(W) = \mathcal{MN}_{K\times C}\big(0,\ sI_K,\ \Sigma\big), \qquad p(Y \mid X, W) = \mathcal{MN}_{N\times C}\big(Y \mid \Phi(X)W,\ I_N,\ \Sigma\big),$$

and approximate the posterior with a matrix-normal variational distribution

$$q(W) = \mathcal{MN}_{K\times C}\big(M_q,\ U_q,\ V_q\big).$$

Using the shorthand $\Phi := \Phi(X)$*, the variational ELBO* $\mathcal{L}(M_q, U_q, V_q)$ *(up to an additive constant independent of the variational parameters) can be written as*

$$\begin{aligned}\mathcal{L}(M_q, U_q, V_q) = -\tfrac{1}{2}\,\mathrm{tr}\Big[\Sigma^{-1}\Big((Y-\Phi M_q)^\top(Y-\Phi M_q) + \big(\mathrm{tr}(U_q\Phi^\top\Phi)\big)\,V_q \\ +\ s\big(\mathrm{tr}(U_q)\,V_q + M_q^\top M_q\big)\Big)\Big] \\ +\ \tfrac{1}{2}\Big[KC(1+\log(2\pi)) + C\log|U_q| + K\log|V_q|\Big]\ +\ \textit{const.}\end{aligned} \tag{11}$$

(Here "const." denotes terms not depending on M_q, U_q, V_q*.)*

Proof. By definition

$$\begin{aligned}\mathcal{L}(M_q, U_q, V_q) &= \int q(W)\log p(Y \mid X, W)\, dW\ -\ \mathrm{KL}(\ q(W)\ ||\ p(W)\ ) \\ &= \mathbb{E}_q[\log p(Y \mid X, W)] + \mathbb{E}_q[\log p(W)] - \mathbb{E}_q[\log q(W)].\end{aligned} \tag{12}$$

We derive the ELBO by evaluating each of the three expectations in Equation (12) separately. We make use of the density in Definition 1 and the properties in Lemma 1 throughout the proof. Also write $\Phi := \Phi(X)$ for convenience.

1. Term $\mathbb{E}_q[\log p(Y \mid X, W)]$.

By Equation (6), the log-likelihood (up to constants) is

$$\log p(Y \mid X, W) = -\tfrac{1}{2}\,\mathrm{tr}\Big[\Sigma^{-1}(Y-\Phi W)^\top(Y-\Phi W)\Big] + \text{const.}$$

Hence

$$\mathbb{E}_q[\log p(Y \mid X, W)] = -\tfrac{1}{2}\,\mathrm{tr}\Big[\Sigma^{-1}\,\mathbb{E}_q\big[(Y-\Phi W)^\top(Y-\Phi W)\big]\Big] + \text{const.}$$

Expand the inner expectation and re-factorise:

$$\begin{aligned}\mathbb{E}_q\big[(Y-\Phi W)^\top(Y-\Phi W)\big] &= (Y-\Phi M_q)^\top(Y-\Phi M_q) \\ &\quad +\ \mathbb{E}_q\big[W^\top\Phi^\top\Phi W\big] - M_q^\top\Phi^\top\Phi M_q.\end{aligned}$$

Applying identity (7) with $A = \Phi^\top\Phi$ and $X = W$ gives

$$\mathbb{E}_q\big[W^\top\Phi^\top\Phi W\big] = V_q\,\mathrm{tr}\big(\Phi^\top\Phi\,U_q\big) + M_q^\top\Phi^\top\Phi M_q.$$

Therefore the difference reduces to the scalar factor times V_q:

$$\mathbb{E}_q\big[(Y-\Phi W)^\top(Y-\Phi W)\big] = (Y-\Phi M_q)^\top(Y-\Phi M_q) + \big(\mathrm{tr}(U_q\Phi^\top\Phi)\big)\,V_q.$$

Substituting into the expectation of the log-likelihood yields

$$\mathbb{E}_q[\log p(Y \mid X, W)] = -\tfrac{1}{2}\,\mathrm{tr}\Big[\Sigma^{-1}\Big((Y-\Phi M_q)^\top(Y-\Phi M_q) + \big(\mathrm{tr}(U_q\Phi^\top\Phi)\big)\,V_q\Big)\Big] + \text{const.} \tag{13}$$

2. Term $\mathbb{E}_q[\log p(W)]$.

The prior log-density (up to constants) is

$$\log p(W) = -\tfrac{1}{2}\,\mathrm{tr}\big[\Sigma^{-1}W^\top(sI_K)W\big] + \text{const.} = -\tfrac{s}{2}\,\mathrm{tr}\big[\Sigma^{-1}W^\top W\big] + \text{const.}$$

Thus

$$\mathbb{E}_q[\log p(W)] = -\tfrac{s}{2}\,\mathrm{tr}\Big[\Sigma^{-1}\,\mathbb{E}_q[W^\top W]\Big] + \text{const.}$$

Use identity (7) with $A = I_K$:

$$\mathbb{E}_q[W^\top W] = V_q\,\mathrm{tr}(U_q) + M_q^\top M_q.$$

Hence

$$\mathbb{E}_q[\log p(W)] = -\tfrac{1}{2}\,\mathrm{tr}\Big[\,s\Sigma^{-1}\,\big(\mathrm{tr}(U_q)\,V_q + M_q^\top M_q\big)\Big] + \text{const.} \tag{14}$$

3. Term $\mathbb{E}_q[\log q(W)]$. The exact log-likelihood of the approximate posterior, following Equation (6), is given by

$$\log q(W) = -\tfrac{1}{2}\,\mathrm{tr}\big[\,V_q^{-1}(W-M_q)^\top U_q^{-1}(W-M_q)\big] - \tfrac{KC}{2}\log(2\pi) - \tfrac{K}{2}\log|V_q| - \tfrac{C}{2}\log|U_q|.$$

Hence

$$\begin{aligned}\mathbb{E}_q[\log q(W)] = -\tfrac{1}{2}\,\mathrm{tr}\Big[\,V_q^{-1}\,\mathbb{E}_q\big[(W-M_q)^\top U_q^{-1}(W-M_q)\big]\Big]\\ -\tfrac{KC}{2}\log(2\pi) - \tfrac{K}{2}\log|V_q| - \tfrac{C}{2}\log|U_q|.\end{aligned}$$

Notice that $W - M_q \sim \mathcal{MN}_{K\times C}\big(\,0,\ U_q,\ V_q\,\big)$, using identity (7) with $X = W - M_q$, $M = 0$ and $A = U_q^{-1}$,

$$\mathbb{E}_q\big[(W-M_q)^\top U_q^{-1}(W-M_q)\big] = V_q\,\mathrm{tr}\big(U_q^{-1\top}U_q\big) + 0 = V_q\,\mathrm{tr}(I_K) = K\,V_q.$$

Hence

$$\begin{aligned}\mathbb{E}_q[\log q(W)] &= -\tfrac{1}{2}\,\mathrm{tr}\big[\,V_q^{-1}(KV_q)\big] - \tfrac{KC}{2}\log(2\pi) - \tfrac{K}{2}\log|V_q| - \tfrac{C}{2}\log|U_q|\\ &= -\tfrac{1}{2}\,K\,\mathrm{tr}(I_C) - \tfrac{KC}{2}\log(2\pi) - \tfrac{K}{2}\log|V_q| - \tfrac{C}{2}\log|U_q|\\ &= -\tfrac{KC}{2} - \tfrac{KC}{2}\log(2\pi) - \tfrac{K}{2}\log|V_q| - \tfrac{C}{2}\log|U_q|.\end{aligned}$$

Equivalently,

$$\mathbb{E}_q[\log q(W)] \;=\; -\tfrac{KC}{2}\big(1+\log(2\pi)\big) \;-\; \tfrac{K}{2}\log|V_q| \;-\; \tfrac{C}{2}\log|U_q|. \tag{15}$$

4. Combine terms into the ELBO. Insert expressions (13), (14) and (15) into the ELBO definition (12) Collecting the trace-terms gives exactly the expression displayed in (11), up to an additive constant that does not depend on M_q, U_q, V_q. This completes the derivation. □

A.4 Numerical Stability for ELBO optimisation

Optimising the ELBO for a matrix-normal variational posterior involves log-determinants, matrix inverses, and trace terms over positive-definite covariance matrices, which are numerically sensitive. To ensure stable optimisation and gradient flow, we adopt several numerical safeguards.

First, all variational covariance matrices are parametrised via *Cholesky factorisation*. Specifically, the row and column covariances are represented as

$$U_q = L_U L_U^\top \qquad \text{and} \qquad V_q = L_V L_V^\top,$$

where L_U and L_V are lower-triangular matrices with strictly positive diagonals enforced via softplus transforms. This guarantees positive definiteness throughout optimisation.

Second, matrix inverses required by the ELBO (e.g. in quadratic and trace terms) are never computed explicitly. Instead, we rely on Cholesky-based solves and inverses, which are numerically more stable and better conditioned for automatic differentiation.

Finally, log-determinant terms appearing in the KL divergence are computed using numerically stable `slogdet` operations rather than explicit determinants, avoiding overflow or underflow. Where necessary, small diagonal jitter terms are added prior to Cholesky or log-determinant computation to prevent failures due to near-singular matrices.

B Monte Carlo Acquisition Functions for Bayesian Active Learning

In this section, we explain how Monte Carlo (MC) sampling is used to compute the different acquisition functions for AL. We recapitulate the BALD derivation provided by the original author in [9], and extend it with other acquisition functions.

Let $p(w \mid \mathcal{D})$ denote the posterior over model parameters after observing dataset $\mathcal{D} = (X,\ Y)$. For a given input x, the posterior predictive distribution is

$$p(y = c \mid x, \mathcal{D}) = \int p(y = c \mid x, W)\, p(w \mid \mathcal{D})\, dw.$$

In practice, this integral is approximated using Monte Carlo sampling:

$$\hat{p}_c^{(t)} \ := \ p(y = c \mid x, \hat{w}_t), \qquad \hat{w}_t \sim p(w \mid \mathcal{D}), \quad t = 1, \dots, T.$$

Define the empirical probability mean

$$\bar{p}_c \ := \ \frac{1}{T} \sum_{t=1}^{T} \hat{p}_c^{(t)}.$$

Also define $\bar{\boldsymbol{p}} = [\bar{p}_1, ..., \bar{p}_C]^\top$ as the probability mean vector. All acquisition functions below are expressed in terms of $\bar{\boldsymbol{p}}$, $\bar{p}_c$ and $\{\hat{p}_c^{(t)}\}_{t=1}^{T}$.

1. BALD (Bayesian Active Learning by Disagreement)

The BALD acquisition function is defined as the mutual information

$$a_{\mathrm{BALD}}(x) = \mathbb{I}[y, w \mid x, \mathcal{D}] = \mathbb{H}[y \mid x, \mathcal{D}] - \mathbb{E}_{p(w \mid \mathcal{D})}\left[\mathbb{H}[y \mid x, w]\right].$$

Using Monte Carlo sampling, this is approximated as

$$a_{\mathrm{BALD}}(x) \approx -\sum_c \bar{p}_c \log \bar{p}_c + \frac{1}{T} \sum_{t=1}^{T} \sum_c \hat{p}_c^{(t)} \log \hat{p}_c^{(t)}.$$

Equivalently,

$$a_{\mathrm{BALD}}(x) = \mathbb{H}[\bar{\boldsymbol{p}}] - \mathbb{E}_t \Big[\mathbb{H}[\text{model prediction}^{(t)}] \Big].$$

2. Maximum Entropy

The maximum entropy acquisition function is defined as

$$a_{\mathrm{MaxEnt}}(x) = \mathbb{H}[y \mid x, \mathcal{D}] = -\sum_c p(y = c \mid x, \mathcal{D}) \log p(y = c \mid x, \mathcal{D}).$$

Using the Monte Carlo approximation of the predictive distribution,

$$a_{\mathrm{MaxEnt}}(x) \ \approx \ -\sum_c \bar{p}_c \log \bar{p}_c = \mathbb{H}[\bar{\boldsymbol{p}}].$$

3. Variation Ratios

The variation ratios acquisition function is defined as

$$a_{\mathrm{VR}}(x) = 1 - \max_c p(y = c \mid x, \mathcal{D}).$$

Using Monte Carlo approximation,

$$a_{\mathrm{VR}}(x) \ \approx \ 1 - \max_c \bar{p}_c$$

4. Mean Standard Deviation (Mean STD)

The Mean STD acquisition function is defined as

$$a_{\mathrm{MSD}}(x) = \frac{1}{C}\sum_c \sqrt{\mathbb{E}_{p(w|\mathcal{D})}[p(y=c \mid x,w)^2] - \mathbb{E}_{p(w|\mathcal{D})}[p(y=c \mid x,w)]^2}.$$

Using Monte Carlo sampling,

$$a_{\mathrm{MSD}}(x) \;\approx\; \frac{1}{C}\sum_c \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{p}_c^{(t)})^2 - \left(\frac{1}{T}\sum_{t=1}^{T}\hat{p}_c^{(t)}\right)^2}.$$

Equivalently,

$$a_{\mathrm{MSD}}(x) = \mathbb{E}_c\Big[\mathrm{std}_t\big(\hat{p}_c^{(t)}\big)\Big].$$

C Optimising ELBO of Bayesian CNN with Non-zero Gaussian Prior Mean

In this section, we show that by incorporating non-zero SSL pretrained weight as the Gaussian prior distribution mean of a Bayesian CNN with approximate Bernoulli VI on the posterior, optimising the variational ELBO is equivalent to training the BCNN model with standard Cross Entropy loss and L2 regularisation anchored at the prior mean. This is a key result for our novel extension in Section 5.

Proposition 4. *Consider a Bayesian model for classification whose weights are represented as $w \in \mathbb{R}^D$, having a Gaussian prior with* ***non-zero mean*** *μ_0:*

$$p(w) = \mathcal{N}(w \mid \mu_0,\ \ell^2 I_D),$$

and a categorical likelihood (softmax) for independent dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$:

$$p(y_i \mid x_i, w) = \mathrm{Categorical}\big(y_i \mid \mathrm{softmax}(f_w(x_i))\big).$$

Approximate the posterior with an independent Bernoulli (dropout) variational family $q(w)$ with mass placed on a point estimate m: let $p \in [0,1]$ denote the dropout probability. Then under q,

$$w_j = m_j z_j, \qquad z_j \sim \textit{Bernoulli}(1-p), \quad j = 1, ..., D,$$

where the variational parameters are m_j, Then maximising the variational ELBO

$$\mathcal{L}_{\textit{ELBO}}(q) = \mathbb{E}_{q(w)}\big[\log p(Y \mid X, w)\big] - \mathrm{KL}\big(q(w)\,\|\,p(w)\big).$$

is (up to additive constants and the standard Monte Carlo approximation of the expectation) equivalent to minimising the empirical cross-entropy loss with an ℓ_2 regularisation anchored at μ_0:

$$\widehat{\mathcal{L}}(m) \;\approx\; \frac{1}{N}\sum_{i=1}^{N} \mathrm{CE}\big(y_i,\ \hat{p}(y \mid x_i; m)\big) \;+\; \lambda\,\|m - \mu_0\|_2^2,$$

where $\hat{p}(y \mid x_i; m)$ denotes the MC estimate of the predictive probabilities obtained by dropout around the point m, and

$$\lambda \propto \frac{1-p}{2\ell^2}$$

(precise constant depends on the exact Bernoulli parametrisation and deterministic normalisation chosen).

Proof. We treat the two terms in ELBO definition separately:

$$\mathcal{L}_{\mathrm{ELBO}}(q) = \mathbb{E}_{q(w)}\big[\log p(Y \mid X, w)\big] - \mathrm{KL}\big(q(w)\,\|\,p(w)\big).$$

1. Expected log-likelihood (first term). Using Monte Carlo dropout one approximates the expectation by stochastic forward passes:

$$\mathbb{E}_{q(w)}\big[\log p(Y \mid X, w)\big] \approx \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N}\log p\big(y_i \mid x_i, \widehat{w}_t\big), \qquad \widehat{w}_t \sim q(w),$$

and for classification each inner log-likelihood $\log p(y_i \mid x_i, \widehat{w}_t)$ is the usual log-probability of the softmax; averaging over t yields the MC estimate of the expected log-probability. Maximising the expected log-likelihood is therefore (after negation) equivalent to minimising the Monte Carlo approximation to the cross-entropy loss:

$$-\mathbb{E}_{q(w)}\big[\log p(Y \mid X, w)\big] \;\approx\; \frac{1}{N}\sum_{i=1}^{N}\mathrm{CE}\big(y_i,\; \hat{p}(y \mid x_i; m)\big),$$

where $\hat{p}(y \mid x_i; m)$ is the empirical predictive probability obtained by the MC forward passes around centre m. This approximates the standard dropout-trained cross-entropy objective used in practice.

2. KL term under Bernoulli VI (second term).

$$\begin{aligned}\mathrm{KL}\big(q(w)\,\|\,p(w)\big) &= \mathbb{E}_{q(w)}\left[\log\frac{q(w)}{p(w)}\right]\\ &= -\mathbb{H}[q(w)] + \mathbb{E}_{q(w)}[-\log p(w)]\,.\end{aligned}$$

Since the entropy term $\mathbb{H}[q(w)]$ is constant of m for a fixed dropout probability p, we only need to minimise the m-dependent term $\mathbb{E}_{q(w)}[-\log p(w)]$.

For a Gaussian prior $p(w) \sim \mathcal{N}(\mu_0,\; \ell^2 I_D)$, we have

$$-\log p(w) = \frac{1}{2\ell^2}\|w-\mu_0\|^2 + \frac{D}{2}\log(2\pi\ell^2).$$

Therefore,

$$\mathrm{KL}\big(q(w)\,\|\,p(w)\big) = \frac{1}{2\ell^2}\mathbb{E}_{q(w)}\left[\|w-\mu_0\|^2\right] + \text{const. w.r.t. } M.$$

Since we assume independent weights, let

$$q(w) = \prod_{j=1}^{D} q_j(w_j), \qquad \text{where } q_j(w_j = m_j) = 1-p, \quad q_j(w_j = 0) = p.$$

Then

$$\begin{aligned}\mathbb{E}_{q_j(w_j)}\big[(w_j-\mu_{0,j})^2\big] &= (1-p)(m_j-\mu_{0,j})^2 + p(0-\mu_{0,j})^2\\ &= (1-p)(m_j-\mu_{0,j})^2 + p\,\mu_{0,j}^2,\end{aligned}$$

where the last term is independent of m.

Hence,

$$\begin{aligned}\mathbb{E}_{q(w)}\left[\|w-\mu_0\|^2\right] &= \mathbb{E}_{q(w)}\left[\sum_{j=1}^{D}(w_j-\mu_{0,j})^2\right]\\ &= \sum_{j=1}^{D}\mathbb{E}_{q_j(w_j)}\left[(w_j-\mu_{0,j})^2\right]\\ &= (1-p)\|m-\mu_0\|^2 + p\sum_{j=1}^{D}\mu_{0,j}^2.\end{aligned}$$

Dropping the last term that is independent of m, we obtain

$$\mathrm{KL}\big(q(w)\,\|\,p(w)\big) \;\propto\; \frac{1-p}{2\ell^2}\,\|m-\mu_0\|^2 \;+\; \text{const. w.r.t. } m.$$

3. Put together (ELBO maximisation $\Leftrightarrow$ loss minimisation). We have thus shown that the ELBO becomes (up to constants independent of m):

$$\mathcal{L}_{\text{ELBO}}(q) \approx \mathbb{E}_q\big[\log p(Y \mid X, w)\big] - \frac{1-p}{2\ell^2}\,\|m - \mu_0\|_2^2.$$

Maximising $\mathcal{L}_{\text{ELBO}}(q))$ is therefore equivalent to minimising the negative expected log-likelihood plus the ℓ_2 anchored at the prior mean:

$$\min_m\; -\mathbb{E}_q\big[\log p(Y \mid X, w)\big] \;+\; \frac{1-p}{2\ell^2}\,\|m - \mu_0\|_2^2.$$

Replacing the expected log-likelihood with its MC cross-entropy approximation yields the practical objective

$$\min_m\; \frac{1}{N}\sum_{i=1}^{N} \mathrm{CE}\big(y_i,\; \hat{p}(y \mid x_i; m)\big) \;+\; \lambda\,\|m - \mu_0\|_2^2, \quad \text{with } \lambda = \frac{1-p}{2\ell^2},$$

which is exactly the standard cross-entropy training objective with an ℓ_2 regulariser anchored at the prior mean μ_0. This completes the proof. □

D Supplementary Materials

This section provides the figures and tables relevant to our reproduction and further experiments.

D.1 Reproduction Results for *Deep Bayesian Active Learning with Image Data*

In this subsection, Figure 1 and Table 1 shows our reproduction outputs for *Section 5.1* in the original paper. Figure 2 shows our reproduction outputs for *Section 5.2* in the original paper.

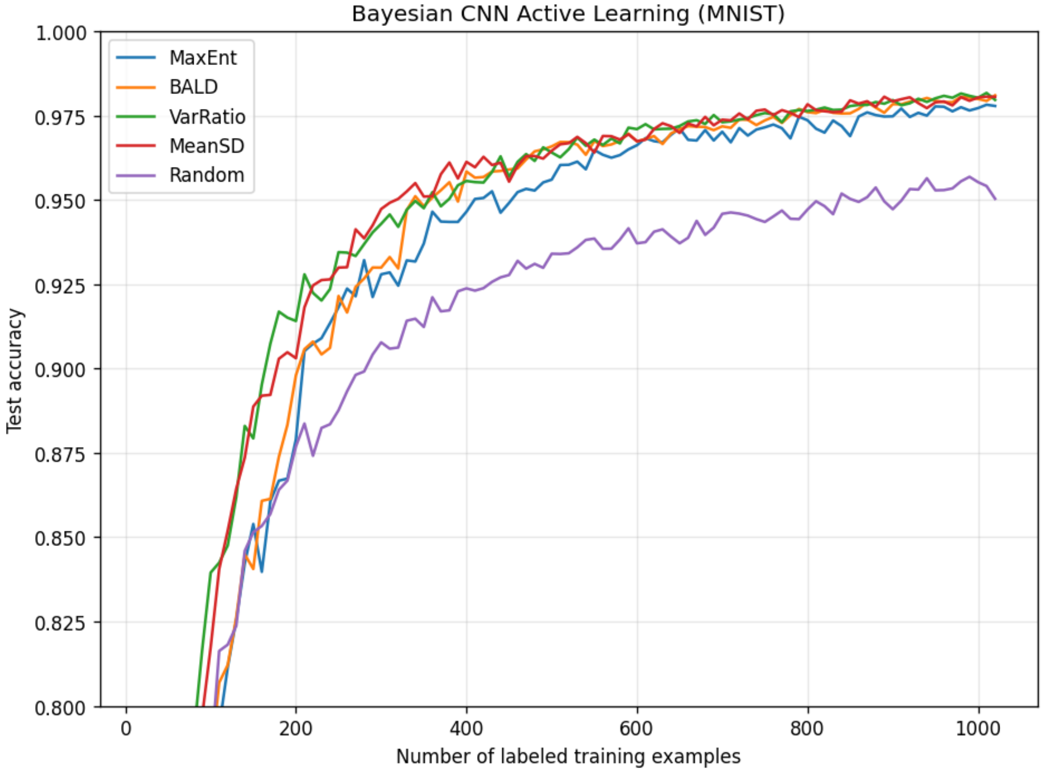


Figure 1: **Replication of Section 5.1.** Comparing test accuracy as a function of number of labelled images among different acquisition functions. Results shows that the 4 acquisition functions (*BALD, Max Entropy, Mean STD, and Variation Ratios*) all perform better than random sampling. **This reproduces the original paper's result, apart from Mean Standard Deviation** – we show more evidence in Figure 3 below.

Table 1: **Replication of Section 5.1.** Number of acquired images to reach target model error

% error	BALD	Var Ratios	Max Ent	Mean STD	Random
10%	210	170	210	180	290
5%	360	360	410	320	840

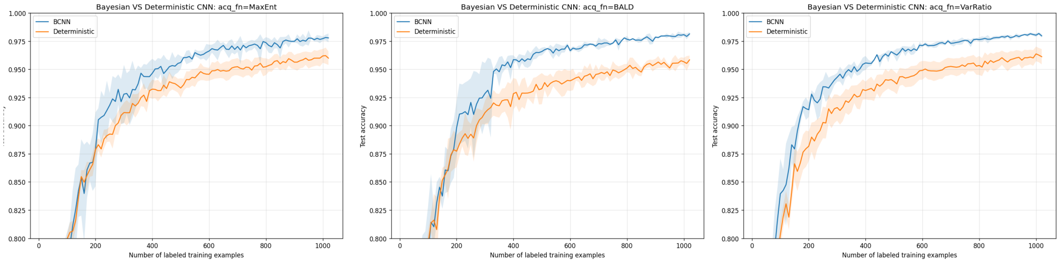


Figure 2: **Replication of Section 5.2.** Comparing test accuracy against number of labelled images for different acquisition functions between a Bayesian CNN (blue) and a deterministic CNN (orange). 1 standard deviation from mean is also shown in shaded region. The results align largely with the original paper, **showing good reproducibility**.

D.2 Evidence for Mean Standard Deviation Acquisition Reproduction Failure

Our results in Figure 3 show that for all 4 acquisition functions other than Random (including *mean standard deviation*), the score distributions are uneven and has a lighter tail on higher scores, indicating that they are able to identify unlabelled samples that would give more information compared to uniform sampling. **This casts doubts on the original paper's implementation of the mean standard deviation acquisition**, as we expect it to perform better than random acquisition.

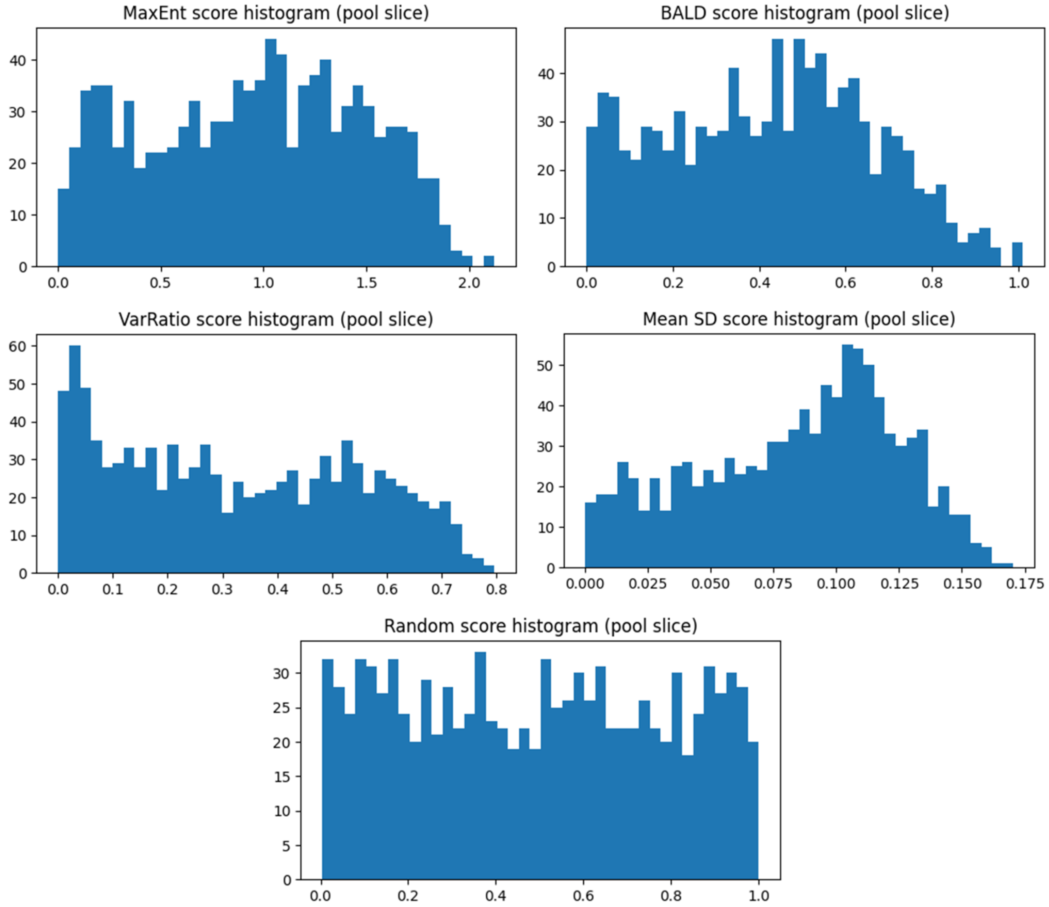


Figure 3: **Acquisition Scores Distribution Histogram.** The histograms shows the distribution of acquisition scores of 1,000 unlabelled samples, evaluated by different acquisition functions. A BCNN model trained on 100 datapoints is used to perform the scoring.

D.3 Comparing Bayesian NN Analytical Inference versus Approximate VI

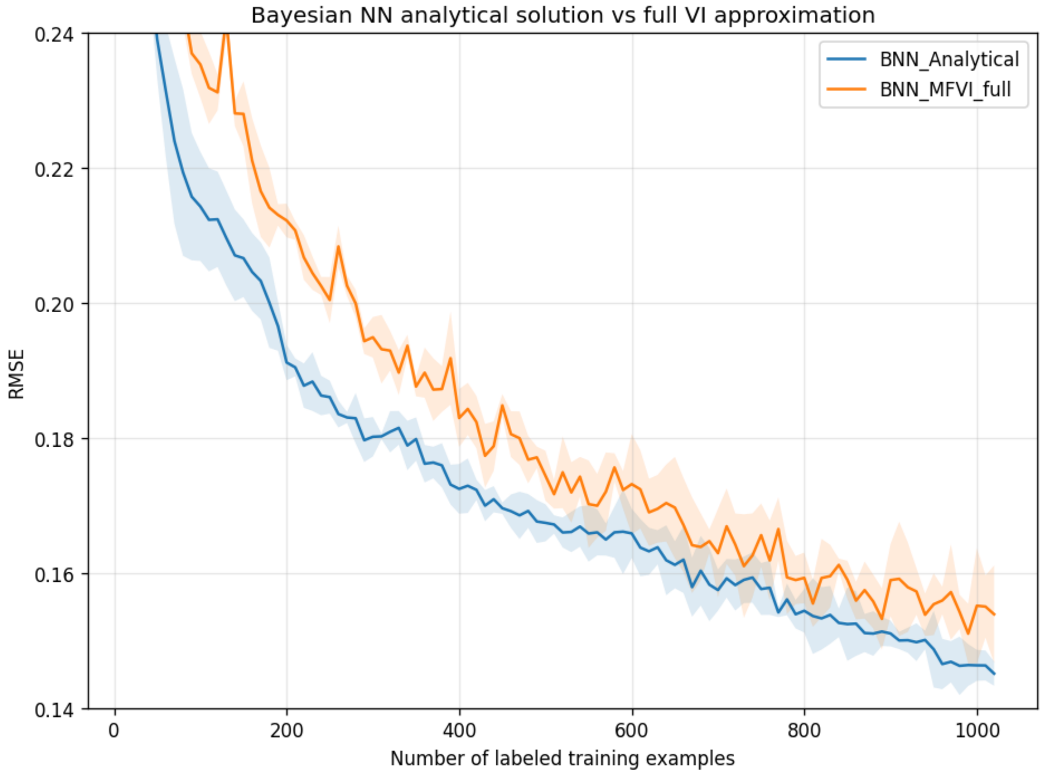
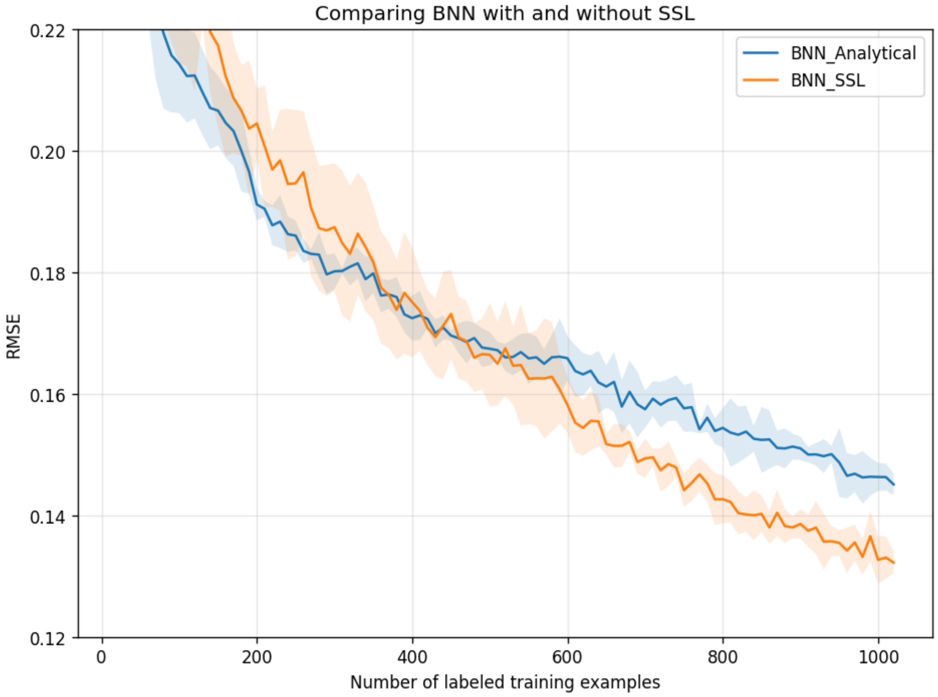


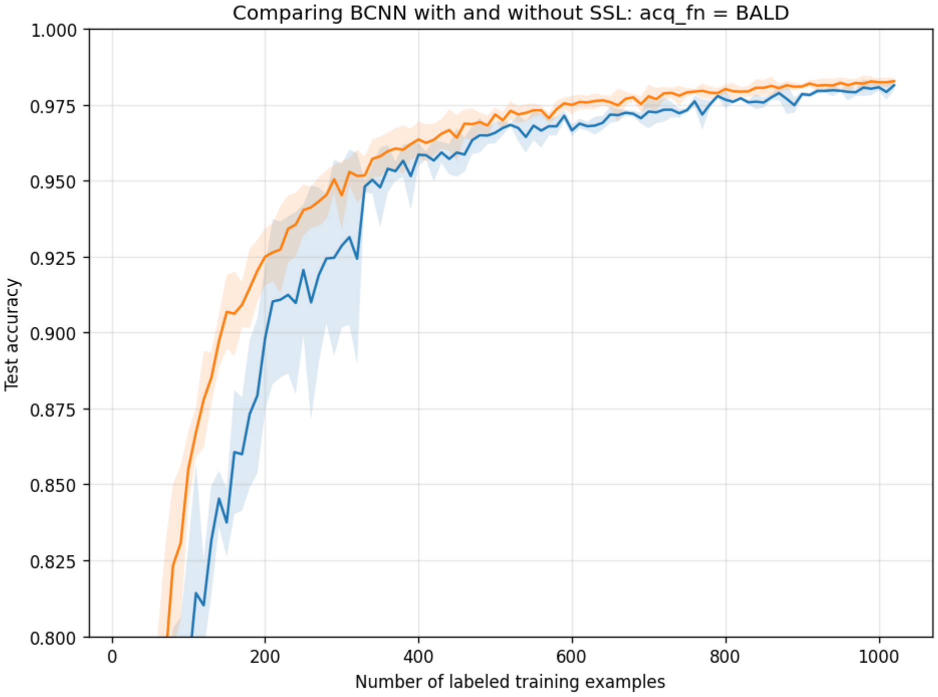
Figure 4: **Comparing analytical AL curve (blue) with VI approximate (orange).** Line shows average of 5 experiments, and 1 standard deviation from mean is also shown in shaded region. The two AL curves theoretically converge to the same RMSE value, but because VI introduces errors in optimising ELBO, the convergence is slower than analytical solution.

D.4 Comparing the Effect of SSL Pretraining on BNN and BCNN Baselines

In this subsection, we demonstrate that by pretraining Bayesian architectures with rotation prediction, and setting pretrained weights to the mean prior distribution over weights, can effectively improve posterior and predictive inference, therefore improving overall AL efficiency. 5 experiments are performed and averaged for all model setups. In the figure below, 1 standard deviation from mean is also shown in shaded region.



(a) **Comparing BNN with SSL (orange) versus without SSL (blue).** SSL pretraining yield better RMSE at later stage, converging to lower error.



(b) **Comparing BCNN with SSL (orange) versus without SSL (blue).** SSL improves AL convergene by showing higher test accuracy at early stage, and slight improvement of overall accuracy.

Figure 5: **Effective SSL improves baseline models.** We demonstrate that SSL pretraining improves AL efficiency for both BNN and BCNN baselines. Average from 5 experiments is displayed, with 1 standard deviation from mean shaded.