Self-Supervised Variational Priors for Robust Bayesian Inference

Anonymous Authors

Abstract

The standard in Bayesian Deep Learning is to use zero-mean isotropic Gaussian priors over neural network weights. When the likelihood is reliable (*e.g.*, with abundant clean labels), these priors matter little. However, when the likelihood is unreliable due to imperfect data, the weakness of standard priors becomes apparent, resulting in poor robustness and uncertainty estimates. To address this, we propose to replace the standard priors with self-supervised variational posteriors. Specifically, we propose a variational method based on instance discrimination. Our objective is derived from the evidence lower bound and made scalable through approximations. Empirically, our priors enhance data-efficiency, robustness to label noise, generalization and calibration on the CIFAR datasets.

1. Introduction

The Issue with Standard Priors. Bayesian Deep Learning (BDL) provides a principled framework for uncertainty quantification, which is essential for robust decision-making (Berger, 2013). However, BDL relies on the challenging task of specifying prior distributions over neural network parameters, where the standard practice employs a zero-mean isotropic Gaussian prior (Fortuin, 2022). With abundant clean labelled data, the posterior can rely on the likelihood while the prior matters little. However, with imperfect data, the likelihood becomes less reliable, revealing the effect of the prior. Figure 1 shows that for scarce and noisy data, the standard prior does not lead to robust inference, highlighting the need for better priors.

Leveraging Unlabelled Data via Self-Supervision. Rather than manually specifying priors over network parameters, we leverage abundant unlabelled data. Self-supervised learning (SSL), particularly contrastive methods based on instance discrimination (Dosovitskiy et al., 2014; Chen et al., 2020), has shown strong empirical success. These methods learn representations by enforcing invariance to stochastic data augmentations that preserve semantic content. However, standard SSL approaches are not Bayesian and yield point estimates of the network parameters.

Self-Supervised Variational Priors. In this work, we propose to construct highly informative priors by framing contrastive learning itself as a Bayesian inference problem. Our contributions are:

- We formulate contrastive self-supervised learning from a variational perspective by defining a classification dataset, augmentation transformations, and a variational objective that treats both augmentations and network parameters as latent variables.
- We address scalability challenges arising from the large number of classes by deriving a stochastic variational objective that avoids full softmax computation and large classification layers.
- Experiments show that the proposed learned prior consistently improves performance in low-data and label-noise regimes outperforming other data priors, and enhances calibration and OoD detection for BDL methods.

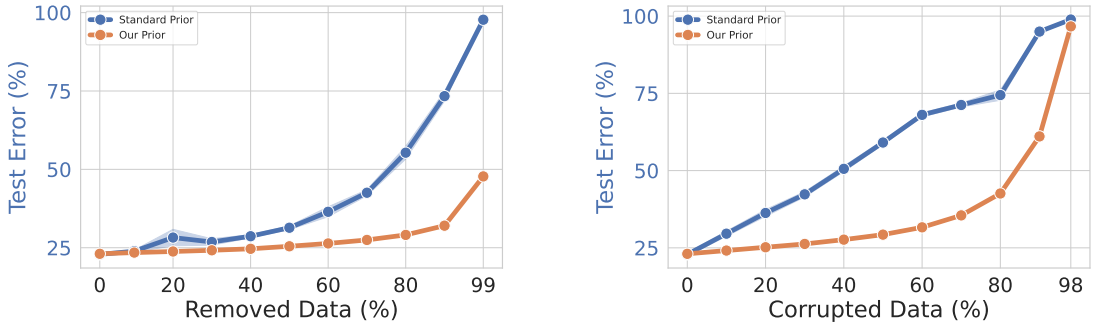

Figure 1: **When likelihoods are weak, priors must be strong.** Here we show that, for two forms of imperfect data, *i.e.*, scarce (left) and noisy (right) on the CIFAR-100 dataset, standard zero-mean isotropic Gaussian distribution priors (blue) over neural network parameters lead to poor posteriors while our self-supervised priors (orange) are more robust.

2. Robust Posterior Predictions via Self-Supervised Priors

The purpose of this section is to describe the goal, context, motivation, and challenges of our work.

Motivation. Bayesian methods make predictions on unseen data $Y_\star, X_\star$ based on a supervised dataset $\mathcal{D}_s$ with the posterior predictive distribution $p(Y_\star|X_\star, \mathcal{D}_s)$. In many cases, it is easy to get unlabelled data, but hard or costly to get reliable labels. With little or unreliable labelled data, posterior estimation becomes challenging as the prior becomes more important and it's hard to provide good priors over neural network parameters. Our goal is to improve the uncertainty estimates and robustness of the posterior predictive by incorporating self-supervised data $\mathcal{D}_{ss}$ in the posterior predictive $p(Y_\star|X_\star, \mathcal{D}_s, \mathcal{D}_{ss})$. Next, we go into the details and challenges of doing this.

Posterior Predictive. Let Z be the unknown random variables (*e.g.*, the neural network parameters), then we make predictions with the posterior predictive distribution as follows: $p(Y_\star|X_\star, \mathcal{D}_s, \mathcal{D}_{ss}) = \int p(Y_\star|X_\star, Z)p(Z|\mathcal{D}_s, \mathcal{D}_{ss})dZ$, which requires the full posterior $p(Z|\mathcal{D}_s, \mathcal{D}_{ss})$, which we study next.

Full True Posterior. By incorporating self-supervised data, our posterior is $p(Z|\mathcal{D}_s, \mathcal{D}_{ss}) \propto p(\mathcal{D}_s|Z)p(Z|\mathcal{D}_{ss})$. In the standard BDL setting, we have $p(Z|\mathcal{D}_{ss}) = p(Z)$ and thus a data-independent prior, whereas in our case we use a data-informed prior, *i.e.*, the posterior of the self-supervised task.

Approximate Posterior from Self-Supervised Data. The posterior $p(Z|\mathcal{D}_{ss})$ is intractable, and there are many methods for getting an approximate posterior $q(Z|\mathcal{D}_{ss})$ instead, *e.g.*, via Laplace approximations (MacKay, 1992), SWAG (Maddox et al., 2019), or variational inference (VI) (Graves, 2011). In particular, the variational posterior $q(Z|\mathcal{D}_{ss})$ is calculated by minimising

$$KL(q(Z|\mathcal{D}_{ss}) \,||\, p(Z|\mathcal{D}_{ss})) = -\mathbb{E}_{q(Z|\mathcal{D}_{ss})}\left[\log p(\mathcal{D}_{ss}|Z)\right] + KL(q(Z|\mathcal{D}_{ss}) \,||\, p(Z)) + C \tag{1}$$

with respect to the variational parameters and where C is a constant term independent of Z.

Robust Two-Step Inference. After this first inference step, $q(Z|\mathcal{D}_{ss})$ may approximate the true prior $p(Z|\mathcal{D}_{ss})$ in the inference of $p(Z|\mathcal{D}_s, \mathcal{D}_{ss}) \propto p(\mathcal{D}_s|Z)p(Z|\mathcal{D}_{ss})$. Thus, for unreliable labelled data $\mathcal{D}_s$, the data-informed prior can help compensate for the unreliable likelihood, see Figure 1 [1].

Related Work. Our closest work is (Sharma et al., 2024), who also takes a variational perspective, but treats augmented inputs as completely new observations, although they are highly correlated with the original inputs, making the posterior artificially overconfident. We treat the augmentation transformations as latent variables, resulting in a finite dataset. Another related work is that of Shwartz-Ziv et al. (2022), who combined the post-hoc SWAG method (Maddox et al., 2019) with the non-Bayesian SimCLR (Chen et al., 2020) method, and thus lacks a principled Bayesian foundation.

1. Indeed, a data-independent prior must be specified in Equation 1, but this is often not an issue as we have a lot of reliable self-supervised data, so the likelihood can compensate for the weak prior.

3. Variational Instance Discrimination

We present Variational Instance Discrimination, a variational adaptation of Instance Discrimination (Dosovitskiy et al., 2014; Wu et al., 2018).

Overview. Our goal is to derive a variational perspective on self-supervised learning for Bayesian deep learning. In terms of Equation 1, this requires defining a dataset, the set of unknowns Z, and the true and approximate posteriors, which then gives a variational objective. We first describe a naive way of doing this in Section 3.1. Next, in Section 3.2, we go over the many scalability issues of the naive approach and propose a new scalable variational objective. Derivations are in Appendix A.

3.1. Naive Variational Instance Discrimination

Unlabelled Data to Classification Dataset. We turn unlabelled data $X = \{x_i\}_{i=1}^N$ into a classification dataset by enumerating each example, *i.e.*, $\mathcal{D}_{ss} = \{(x_i, i)\}_{i=1}^N$. This is called Instance Discrimination (Dosovitskiy et al., 2014), and the task for the network is thus to classify each example.

Set of Unknowns Z. In our case, we consider $Z = \{a, \theta\}$, where $a = \{a_i\}_{i=1}^N$ are random variables corresponding to stochastic data augmentation transformations that are applied to the original inputs X, and $\theta = \{\theta^b, \theta^c\}$ are the neural network parameters for the feature extractor and classification layer, respectively. As we do instance discrimination, the number of classes K is equal to the number of examples N, and thus $\theta^c \in \mathbb{R}^{F \times N}$, where F is the feature dimension.

True and Naive Variational Posteriors. From the objective in Equation 1, our focus is on the true $p(Z|\mathcal{D}_{ss})$ and the approximate $q(Z|\mathcal{D}_{ss})$ posteriors. Given that $Z = \{a, \theta\}$, we have let the true distribution factorize as $p(a, \theta|\mathcal{D}) \propto p(Y|a, \theta^b, \theta^c, X)p(\theta^b)p(\theta^c)p(a)$. The naive way to define the approximate posterior would be $q(a, \theta|\mathcal{D}_{ss}) \propto q(\theta^c)q(\theta^b)q(a)$. However, the large classification layer $\theta^c \in \mathbb{R}^{F \times N}$ poses challenges for efficient learning and memory usage as it depends on N [2].

In the next section, we solve this issue by a particular choice of parameterisation of the approximate posterior and, through unbiased estimates of sums, remove dependencies on N.

3.2. Scalable Variational Instance Discrimination

Scalable Variational Posterior. To address the challenge of the classification layer scaling with dataset size, we draw inspiration from amortized variational inference and propose to use the network feature output to parametrize per-example variational parameters. Specifically, we introduce another set of augmentations and factorize the approximate posterior over network parameters as follows:

$$q(Z|\mathcal{D}_{\text{ss}}) = q(a, a', \theta|X, Y) = q(\theta^c|a', \theta^b, X)p(a, a')q(\theta^b) = q(\theta^b)p(a)\prod_{k=1}^{N} q(\theta^c_{\cdot,k}|x_k, a'_k, \theta^b)p(a'_k) \quad (2)$$

where a, a' are two augmentation transformations for each input. The classification layer parameters θ^c now depend on the augmented data $a'(X) = \{a'_1(x_1), \ldots, a'_N(x_N)\}$, while θ^b remain data-independent. Note that we have factorised the distribution over the classification layer parameters as well, where $\theta^c_{\cdot,k} \in \mathbb{R}^F$ represents the kth column of θ^c.

Combining our results so far, we get an objective similar to Equation 1:

$$\sum_{i=1}^{N}\left(\mathbb{E}\left[KL(q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \,||\, p(\theta^c_{\cdot,i}))\right] - \mathbb{E}\left[\log \frac{\exp(f_{i,y_i})}{\sum_{k=1}^{N}\exp(f_{i,k})}\right]\right) + KL(q(\theta^b) \,||\, p(\theta^b) + C \quad (3)$$

2. For example, for a standard ResNet-50 where $F = 2048$ and on a relatively small dataset with 100k examples, the parameters of the classification layer would take up 90% of all parameters.

where second expectation is over $q(a, a', \theta|X, Y)$ and first is over $q(\theta^b)p(a')$, and where we have made use of the i.i.d. assumption and where $f_{ik} = f_k(a_i(x_i), \theta) = \phi(a_i(x_i), \theta^b)\theta^c_{\cdot,k} \in \mathbb{R}$ where $\phi(a(x_i), \theta^b) \in \mathbb{R}^{1\times F}$ are called the features of the network, $\theta^c_{\cdot,k} \in \mathbb{R}^{F\times 1}$ is the kth column of the classification layer.

The VI objective for self-supervised learning in Equation 3 has fundamental scalability issues as it clearly depends on the dataset size in the two sums. Next, we propose unbiased stochastic approximations of the sums that remove the dependence on N, allowing us to scale to large datasets.

Removing Data Dependence on N. Similar to standard stochastic variational inference we can do an unbiased Monte Carlo estimate of the outer sum over N in Equation 3 resulting in a sum over a mini-batch $\mathcal{B}$ of size $B = |\mathcal{B}| \ll N$ instead that is scaled by N/B.

Removing Softmax Dependence on N. To remove the dependence on dataset size in the denominator of Equation 3, we again do an unbiased estimate of the sum. Concretely,

$$\log \frac{\exp\left(f_{i,y_i}\right)}{\sum_{k=1}^{N}\exp\left(f_{i,k}\right)} \approx \log \frac{\exp\left(f_{i,y_i}\right)}{\frac{N}{B}\sum_{x_k\in\mathcal{B}}\exp\left(f_{i,k}\right)} = \log \frac{\exp\left(f_{i,y_i}\right)}{\sum_{x_k\in\mathcal{B}}\exp\left(f_{i,k}\right)} + \log\frac{B}{N} \tag{4}$$

where $\log B/N$ is a constant that does not affect the optimisation of variational parameters. Note that VI to SVI batched the sum over examples, while this batches the sum over classes.

Scalable Objective. As the approximations above made the objective only depend on the current batch, we can marginalize out parameters of $q(a, a', \theta|X)$ that do not contribute. Thus, we only need to sample the classification layer parameters and augmentations related to the current batch.

We let $q(\theta^c_{i,k}|a'_k(x_k), \theta^b) = \mathcal{N}(\theta^c_{i,k}; \phi_i(a'_k(x_k), \theta^b)/\tau, s_c^2)$, where we normalize the network feature outputs such that $||\phi(a'_k(x_k), \theta^b)||_2 = 1$ and where τ and s_c^2 are variational parameters, which we keep fixed for simplicity. That is, each column $\theta^c_{\cdot,k}$ have the same magnitude $1/\tau$, but different directions $\phi(a'_k(x_k), \theta^b)$ and $f_{i,k}$ is a dot product $\phi^\top(a_i(x_i), \theta^b)\left(\phi(a'_k(x_k), \theta^b) + \epsilon\right)$ where $\epsilon \sim \mathcal{N}(0, s_c^2)$.

We combine our results and use Monte Carlo estimation of the expectations with single samples for network parameters and S samples augmentation distributions, to turn Equation 3 into

$$-\frac{1}{S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left[\frac{N}{B}\sum_{(x_i,y_i)\in\mathcal{B}}\log\frac{\exp\left(\phi(a_i^{(s)}(x_i), \theta^b)\theta^{(s')}_{\cdot,y_i}\right)}{\sum_{x_k\in\mathcal{B}}\exp\left(\phi(a_i^{(s)}(x_i), \theta^b)\theta^{(s')}_{\cdot,k}\right)}\right] + KL\left(q(\theta^b)\,||\,p(\theta^b)\right) + C \tag{5}$$

where $\theta^{(s')}_{\cdot,k} \sim q(\theta^c_{\cdot,k}|a'^{(s')}_k(x_k), \theta^b)$, the KL term for the classification layer is constant w.r.t. learnable variational parameters and is therefore in the constant C term. Through approximations, we have arrived at a variational objective that only depends on a batch and thus scales to large datasets.

4. Experiments

We evaluate the proposed data prior across several BDL and non-Bayesian inference settings. We compare predictive performance with and without our Variational Instance Discrimination (VID) prior under challenging conditions, including limited labelled data and label noise. We further assess calibration and out-of-distribution (OoD) detection.

Baselines. We consider the following inference methods: *MAP*, which performs maximum a posteriori estimation; *SGLD* (Welling and Teh, 2011), a stochastic gradient-based posterior sampler; and *VI*, variational inference with a parametric posterior approximation. For representation learning baselines, we also include Sharma et al. (2024), *SimCLR*, and *SimCLR+SWAG* (Shwartz-Ziv et al., 2022), which provide different data priors without explicit Bayesian modelling.

Table 1: **Predictive Performance.** We report results for (**a**) different data priors and (**b**) BDL methods with and without the proposed prior on CIFAR-10 and CIFAR-100. The anomaly detection is evaluated on the SVHN test set. The proposed data prior VID consistently improves performance.

Prior	Dataset Subset				Label Noise			
	CIFAR-10		CIFAR-100		CIFAR-10		CIFAR-100	
	1%	10%	1%	10%	20%	40%	20%	40%
Sharma	87.32±.147	90.50±.204	47.48±.197	59.80±.330	90.82±.193	90.75±.230	61.19±.400	59.79±.511
SimCLR	87.54±.190	90.50±.113	44.93±.459	59.39±.494	90.82±.142	90.74±.163	61.49±.371	60.19±.229
SimCLR+SWAG	87.86±.261	92.70±.133	45.39±.554	63.62±.536	93.99±.149	93.30±.179	72.20±.159	69.79±.254
VID	88.20±.098	92.57±.160	49.99±.386	65.41±.135	93.57±.155	93.01±.215	71.73±.326	69.46±.223
VID 8 augs.	**88.60**±.260	**93.47**±.060	**52.10**±.430	**68.37**±.210	**94.62**±.080	**94.06**±.130	**74.40**±.220	**71.39**±.460

	Method	Data Prior	Acc. (↑)	ID Calibration			OoD Detection	
				NLL (↓)	Brier (↓)	ECE (↓)	FPR@95 (↓)	AUC (↑)
CIFAR-100	MAP	✗	76.24±0.88	0.889±0.031	33.227±0.991	0.031±0.002	24.6±1.1	82.4±1.0
		✓	**79.34**±0.21	**0.692**±0.001	**28.556**±0.110	**0.011**±0.001	**23.4**±0.7	**83.3**±0.5
	SGLD	✗	77.12±0.55	0.857±0.027	32.176±0.819	0.034±0.001	24.2±0.6	82.6±0.6
		✓	**79.33**±0.25	**0.690**±0.004	**28.597**±0.203	**0.010**±0.001	**23.3**±0.7	**83.4**±0.5
	VI	✗	76.01±0.40	0.873±0.007	33.060±0.332	**0.016**±0.003	**17.3**±1.2	**88.7**±0.9
		✓	**78.46**±0.17	**0.724**±0.003	**29.829**±0.107	0.039±0.002	21.5±0.9	85.4±0.5

Benchmarks. We conduct experiments on CIFAR-10 and CIFAR-100 using a ResNet-50 backbone. We evaluate predictive accuracy, calibration, and OoD detection using SVHN as the OoD dataset.

Low-Data Regime. We simulate limited supervision by training on 1% and 10% of labelled data. As shown in Table 1a, the proposed VID prior consistently improves performance across datasets. Compared to representation-based baselines (*e.g.*, SimCLR), our approach achieves superior performance, particularly in the most data-scarce settings. This highlights the ability of the prior to regularize learning and mitigate overfitting.

Label Noise. We evaluate robustness to label corruption by injecting 20% and 40% uniform noise. The proposed prior consistently improves performance across all settings.

Uncertainty and OoD Detection. Table 1b shows that incorporating the proposed prior improves both calibration and OoD detection for MAP and SGLD compared to using the standard prior. For VI, the prior improves predictive accuracy but introduces a trade-off in calibration and OoD metrics, suggesting that further refinement of the variational family may be beneficial.

Number of Augmented Pairs. We vary the number of augmented pairs per image, *i.e.* S. Increasing the number of augmentations consistently improves performance, with the best results achieved using 8 augmentations (Table 1a). This suggests that greater augmentation diversity helps learn more invariant representations. However, the gains come at increased computational cost.

5. Conclusion

We present a variational formulation of contrastive self-supervised learning by defining a classification dataset, a likelihood, and a variational objective that explicitly incorporates data augmentation. We further show that a tractable approximation of this objective recovers a form closely related to SimCLR. Our approach bridges Bayesian inference and self-supervised learning, providing a principled framework for learning data-dependent priors and improving uncertainty estimation. We hope this perspective enables more reliable and practically useful deep learning methods.

References

James O Berger. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

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.

Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. *Advances in neural information processing systems*, 27, 2014.

Vincent Fortuin. Priors in bayesian deep learning: A review. *International Statistical Review*, 90(3): 563–591, 2022.

Alex Graves. Practical variational inference for neural networks. *Advances in neural information processing systems*, 24, 2011.

David JC MacKay. A practical bayesian framework for backpropagation networks. *Neural computation*, 4(3):448–472, 1992.

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

Mrinank Sharma, Tom Rainforth, Yee Whye Teh, and Vincent Fortuin. Incorporating unlabelled data into bayesian neural networks. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=q2AbLOwmHm. Expert Certification.

Ravid Shwartz-Ziv, Micah Goldblum, Hossein Souri, Sanyam Kapoor, Chen Zhu, Yann LeCun, and Andrew G Wilson. Pre-train your loss: Easy bayesian transfer learning with informative priors. In *Advances in Neural Information Processing Systems*, volume 35, pages 27706–27715. Curran Associates, Inc., 2022.

Max Welling and Yee Whye Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688, 2011.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.

Supplementary Material for Self-Supervised Variational Priors for Robust Bayesian Inference

A. Derivations

A.1. Deriving the Scalable Variational Instance Discrimination Objective

Overview. We will start by deriving Equation 1 for our unknowns Z. Then, we will derive Equation 3, and then the final scalable variational instance discrimination objective in Equation 5.

Deriving Equation 1. Let $q(a, a', \theta|\mathcal{D}_{ss})$ be the approximate posterior and $p(a, a', \theta|\mathcal{D}_{ss})$ be the true posterior, then the KL divergence between them is

$$KL(q(a, a', \theta|\mathcal{D}_{ss}) \,||\, p(a, a', \theta|\mathcal{D}_{ss})) = \int q(a, a', \theta|\mathcal{D}_{ss}) \log \frac{q(a, a', \theta|\mathcal{D}_{ss})}{p(a, a', \theta|\mathcal{D}_{ss})} dada'd\theta \tag{6}$$

$$= \int q(a, a', \theta|\mathcal{D}_{ss}) \log \frac{q(a, a', \theta|\mathcal{D}_{ss})p(\mathcal{D}_{ss})}{p(\mathcal{D}_{ss}|a, \theta)p(a, a', \theta)} dada'd\theta \tag{7}$$

$$= \int q(a, a', \theta|\mathcal{D}_{ss}) \log \frac{q(a, a', \theta|\mathcal{D}_{ss})}{p(\mathcal{D}_{ss}|a, \theta)p(a, a', \theta)} dada'd\theta + \log p(\mathcal{D}_{ss}) \tag{8}$$

$$= \int -q(a, a', \theta|\mathcal{D}_{ss}) \log p(\mathcal{D}_{ss}|a, \theta) dada'd\theta \tag{9}$$

$$+ \int q(a, a', \theta|\mathcal{D}_{ss}) \log \frac{q(a, a', \theta|\mathcal{D}_{ss})}{p(a, a', \theta)} dada'd\theta + \log p(\mathcal{D}_{ss}) \tag{10}$$

$$= -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\log p(\mathcal{D}_{ss}|a, \theta)\right] + KL(q(a, a', \theta|\mathcal{D}_{ss}) \,||\, p(a, a', \theta)) + \log p(\mathcal{D}_{ss}) \tag{11}$$

Deriving Equation 3. First, we consider the expected negative log likelihood term in Equation 11. By making the i.i.d. assumption and the definition of the categorical likelihood and the use of the softmax activation, we get

$$-\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\log p(\mathcal{D}_{ss}|a, \theta)\right] = -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\sum_{i=1}^{N} \log p(y_i|x_i, a_i, \theta)\right] \tag{12}$$

$$= -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\sum_{i=1}^{N} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{j=1}^{N} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] \tag{13}$$

Next, we consider the KL term in Equation 11. We reiterate the factorization of the approximate posterior

$$q(a, a', \theta|\mathcal{D}_{ss}) = q(a, a', \theta^b, \theta^c|\mathcal{D}_{ss}) \tag{14}$$

$$= q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(a, a', \theta^b|\mathcal{D}_{ss}) \tag{15}$$

$$= q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b|a, a', \mathcal{D}_{ss})q(a, a'|\mathcal{D}_{ss}) \tag{16}$$

$$= q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)q(a, a'|\mathcal{D}_{ss}) \tag{17}$$

$$= q(\theta^b) \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a'_i, \theta^b, x_i)q(a_i, a'_i) \tag{18}$$

$$= q(\theta^b) \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a'_i, \theta^b, x_i)p(a_i, a'_i) \tag{19}$$

where in the last step we assume $q(a_i, a_i') = p(a_i, a_i')$, *i.e.*, we have access to the true data augmentation distribution. Plugging in this factorization in the KL term, we get

$$KL(q(a, a', \theta|\mathcal{D}_{ss}) \,||\, p(a, a', \theta)) = KL(q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)p(a)p(a') \,||\, p(\theta^c)p(\theta^b)p(a)p(a')) \tag{20}$$

$$= \int q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)p(a)p(a') \log \frac{q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)p(a)p(a')}{p(\theta^c)p(\theta^b)p(a)p(a')} d\theta^c d\theta^b da da' \tag{21}$$

$$= \int q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)p(a)p(a') \left(\log \frac{q(\theta^c|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c)} + \log \frac{q(\theta^b)}{p(\theta^b)} \right) d\theta^c d\theta^b da da' \tag{22}$$

$$= \int q(\theta^c|a', \theta^b, \mathcal{D}_{ss})q(\theta^b)p(a') \log \frac{q(\theta^c|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c)} d\theta^c d\theta^b da' + \int q(\theta^b) \log \frac{q(\theta^b)}{p(\theta^b)} d\theta^b \tag{23}$$

$$= \int q(\theta^b)p(a') \left[\int q(\theta^c|a', \theta^b, \mathcal{D}_{ss}) \log \frac{q(\theta^c|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c)} d\theta^c \right] d\theta^b da' + \int q(\theta^b) \log \frac{q(\theta^b)}{p(\theta^b)} d\theta^b \tag{24}$$

$$= \int q(\theta^b)p(a') KL(q(\theta^c|a', \theta^b, \mathcal{D}_{ss}) \,||\, p(\theta^c)) d\theta^b da' + KL(q(\theta^b) \,||\, p(\theta^b)) \tag{25}$$

$$= \mathbb{E}_{q(\theta^b)p(a')} \left[KL(q(\theta^c|a', \theta^b, \mathcal{D}_{ss}) \,||\, p(\theta^c)) \right] + KL(q(\theta^b) \,||\, p(\theta^b)) \tag{26}$$

Now, let's look closer at the first KL term and make use of the factorization over data for the variational distribution over θ^c

$$KL(q(\theta^c|a', \theta^b, \mathcal{D}_{ss}) \,||\, p(\theta^c)) = \int q(\theta^c|a', \theta^b, \mathcal{D}_{ss}) \log \frac{q(\theta^c|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c)} d\theta^c \tag{27}$$

$$= \int \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \log \frac{\prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss})}{\prod_{i=1}^{N} p(\theta^c_{\cdot,i})} d\theta^c \tag{28}$$

$$= \int \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \left[\log \prod_{i=1}^{N} \frac{q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c_{\cdot,i})} \right] d\theta^c \tag{29}$$

$$= \int \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \left[\sum_{i=1}^{N} \log \frac{q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c_{\cdot,i})} \right] d\theta^c \tag{30}$$

$$= \int \sum_{i=1}^{N} q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \log \frac{q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c_{\cdot,i})} d\theta^c \tag{31}$$

$$= \sum_{i=1}^{N} \int q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \log \frac{q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss})}{p(\theta^c_{\cdot,i})} d\theta^c \tag{32}$$

$$= \sum_{i=1}^{N} KL(q(\theta^c_{\cdot,i}|a', \theta^b, \mathcal{D}_{ss}) \,||\, p(\theta^c_{\cdot,i})) \tag{33}$$

Now, plugging Equation 33 in Equation 26 and combining it with Equation 13 and making use of the linearity of expectation gives Equation 3.

Deriving Equation 5. As mentioned in Section 3.2, we let $q(\theta^c_{i,k}|a'_k(x_k), \theta^{bp}) = \mathcal{N}(\theta^c_{i,k}; \phi_i(a'_k(x_k), \theta^{bp})/\tau, s_c^2)$, where the variational parameters τ and s_c^2 are treated as fixed, and the KL term in Equation 33 only depends on these variational parameters, it is just a constant.

What is left to show is the expected log-likelihood term. Starting from Equation 13 we have

$$- \mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\log p(\mathcal{D}_{ss}|a,\theta)\right] = -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\sum_{i=1}^{N} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{j=1}^{N} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] \tag{34}$$

$$= -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[N\mathbb{E}_{i\sim\mathcal{U}([1,N])} \left[\log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{N\mathbb{E}_{j\sim\mathcal{U}([1,N])}\left[\exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})\right]}\right]\right] \tag{35}$$

$$\approx -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\frac{N}{B} \sum_{(x_i,y_i)\in\mathcal{B}} \left[\log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\frac{N}{B}\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right]\right] \tag{36}$$

$$= -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\frac{N}{B} \sum_{(x_i,y_i)\in\mathcal{B}} \left[\log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right]\right] + C \tag{37}$$

where we have used unbiased Monte Carlo estimation of each sum by first rewriting the sum as N times the expectation over a uniform distribution. Note that although each sum is an unbiased estimate, the estimate of the full likelihood is still biased.

To get to Equation 5, we need to simplify the distribution we are taking the expectation over

$$- \mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\log p(\mathcal{D}_{ss}|a,\theta)\right] \approx -\mathbb{E}_{q(a,a',\theta|\mathcal{D}_{ss})} \left[\frac{N}{B} \sum_{(x_i,y_i)\in\mathcal{B}} \left[\log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right]\right] \tag{38}$$

$$= -\frac{N}{B}\mathbb{E}_{q(\theta^b)} \left[\mathbb{E}_{q(\theta^c|a',\theta^b,X)p(a,a')} \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right]\right] \tag{39}$$

$$\approx -\frac{N}{B}\mathbb{E}_{q(\theta^c|a',\theta^b,X)p(a,a')} \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] \tag{40}$$

$$= -\frac{N}{B} \int \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] \prod_{i=1}^{N} q(\theta^c_{\cdot,i}|a'_i, \theta^b, x_i)p(a_i)p(a'_i)da_i da'_i d\theta^c_{1:N} \tag{41}$$

$$= -\frac{N}{B} \int \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] \prod_{i=1}^{B} q(\theta^c_{\cdot,i}|a'_i, \theta^b, x_i)p(a_i)p(a'_i)da_i da'_i d\theta^c_{1:B} \tag{42}$$

$$= -\frac{N}{B} \int \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j, \theta^b)\theta^c_{\cdot,j})}\right] p(a_{1:B})p(a'_{1:B})q(\theta^c_{\cdot,1:B}|a'_{1:B}, \theta^b, x_{1:B})da_{1:B}da'_{1:B}d\theta^c_{1:B} \tag{43}$$

$$\approx -\frac{N}{B}\frac{1}{S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S} \int \left[\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i^{(s)}, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j^{(s)}, \theta^b)\theta^c_{\cdot,j})}\right] q(\theta^c_{\cdot,1:B}|(a')^{(s')}_{1:B}, \theta^b, x_{1:B})d\theta^c_{1:B} \tag{44}$$

$$\approx -\frac{N}{B}\frac{1}{S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\sum_{(x_i,y_i)\in\mathcal{B}} \log \frac{\exp(\phi^\top(x_i, a_i^{(s)}, \theta^b)\theta^c_{\cdot,y_i})}{\sum_{x_j\in\mathcal{B}} \exp(\phi^\top(x_j, a_j^{(s)}, \theta^b)\theta^c_{\cdot,j})} \tag{45}$$

where we have used the definition of the expectation and the factorization of the approximate posterior, and marginalized out all unused parameters of θ^c, rewritten it as three integrals over

$p(a_{1:B})$, $p(a'_i)$ and $q(\theta^c_{\cdot,1:B}|a'_{1:B}, \theta^b, x_{1:B})$. Next, we use S samples to separately estimate the integrals over $p(a_{1:B})$ and $p(a'_i)$, and lastly a single sample estimate for $q(\theta^c_{\cdot,1:B}|a'_{1:B}, \theta^b, x_{1:B})$.

Thus, to evaluate this, we would first sample the parameters θ^b, then sample the augmentations for the features $a_{1:B}$, then sample the augmentations for the last layer $a'_{1:B}$, and lastly sample the last layer weights $\theta^c_{\cdot,1:B}$, and then we have all information to evaluate the inner sum, and we do this S^2 times and take the average (note that we use the same weight sample in all S^2 cases).