Generalization Gap in Amortized Inference

Mingtian Zhang **Peter Hayes** **David Barber**

AI Center, University College London
{mingtian.zhang.17,peter.hayes.15,david.barber}@ucl.ac.uk

Abstract

The ability of likelihood-based probabilistic models to generalize to unseen data is central to many machine learning applications. The Variational Auto-Encoder (VAE) is a popular class of latent variable model used for many such applications including density estimation, representation learning and lossless compression. In this work, we highlight how the common use of amortized inference to scale the training of VAE models to large data sets can be a major cause of poor generalization performance. We propose a new training phase for the inference network that helps reduce over-fitting to training data. We demonstrate how the proposed scheme can improve generalization performance in the context of image modeling.

1 Generalization of Probabilistic Models

We are interested in approximating the data distribution $p_d(x)$ with a model $p_\theta(x)$. A principled method to learn parameters θ is to minimize the Kullback-Leibler (KL) divergence training objective

$$\mathrm{KL}(p_d||p_\theta) = -\langle \log p_\theta(x)\rangle_{p_d(x)} - \mathrm{H}(p_d(x)), \tag{1}$$

where the entropy term $\mathrm{H}(p_d(x))$ is a constant. We use $\langle\cdot\rangle$ to denote integration: $\langle f(x)\rangle_{p(x)} = \int f(x)p(x)dx$. Minimizing the KL divergence is equivalent to maximizing the likelihood $\langle \log p_\theta(x)\rangle_{p_d(x)}$. In practice p_d is unknown and we only have access to a finite training data set $\mathcal{X}_{train} = \{x_1, \ldots, x_N\} \sim p_d(x)$. We therefore approximate $p_d(x)$ by the empirical distribution $\hat{p}_d(x) = \frac{1}{N}\sum_{n=1}^{N} \delta(x - x_n)$. The training objective becomes:

$$\langle \log p_\theta(x)\rangle_{p_d(x)} \approx \langle \log p_\theta(x)\rangle_{\hat{p}_d(x)} = \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x_n). \tag{2}$$

For a flexible $p_\theta(x)$, this approximation can over-fit the model to the training data. This results in $\mathrm{KL}(p_d||p_\theta)$ being greater than 0. We refer to $\mathrm{KL}(p_d||p_\theta)$ as the *model generalization gap*. This gap is not tractable because we cannot access the $p_d(x)$. We instead use the test likelihood $\frac{1}{M}\sum_{m=1}^{M} \log p_\theta(x'_m)$ to approximate the generalization gap:

$$\mathrm{KL}(p_d||p_\theta) \approx \frac{1}{M}\sum_{m=1}^{M} \log p_\theta(x'_m) + const., \tag{3}$$

where $\mathcal{X}_{test} = \{x'_1, \ldots, x'_M\} \sim p_d(x)$ is the test dataset. As $M \to \infty$, higher test likelihood indicates a smaller model generalization gap.

In the application of lossless compression, the test likelihood has a practical meaning : a lossless compressor can be designed based on p_θ where the average compression length is approximately equal to $-\frac{1}{M}\sum_{m=1}^{M} \log_2 p_\theta(x'_m)$ on the test data set (see [9] for a detailed introduction). Therefore, the better model in terms of test likelihood leads to a greater saving in bits. This is an illustration of the practical importance of model generalization.

2 Generalization of Latent Variable Model with Amortized Inference

For a latent variable models of the form $p_\theta(x) = \int p_\theta(x|z)p(z)dz$, where z is the unobserved latent. When $p_\theta(x|z)$ is parameterized by a non-linear neural network, the evaluation of $\log \int p_\theta(x|z)p(z)dz$ is usually intractable. In this case, the evidence lower bound (ELBO) can instead be used

$$\langle \log p_\theta(x) \rangle_{p_d(x)} \geq \langle \log p_\theta(x,z) - \log q_\phi(z|x) \rangle_{q_\phi(z|x)p_d(x)} \equiv \big\langle \mathrm{ELBO}(x,\theta,\phi) \big\rangle_{p_d(x)}, \tag{4}$$

where $q_\phi(z|x)$ is a posterior approximation parameterized by an inference neural network with parameters ϕ. The ELBO is traditionally maximized with respect to both ϕ and θ.

This model is referred to as a Variational Auto-Encoder (VAE) [6, 11]. The use of an approximate posterior of the form $q_\phi(z|x)$ is called *amortized inference*. We denote the posterior family of $q_\phi(z|x)$ as $\mathcal{Q}$. If $\mathcal{Q}$ is flexible enough such that $p_\theta(z|x) \in \mathcal{Q}$ then at the optimum of equation 4 $q_\phi(z|x) = p_\theta(z|x)$ for every $x \in p_d(x)$ and the inequality in equation 4 becomes an equality. Many previous methods are developed to increase the flexibility of $\mathcal{Q}$ increasing the practical utility of these methods. For example, adding auxiliary variables [1, 8] or flow-based methods [3, 10].

Recent works [13, 14, 7] have successfully applied VAE style models to lossless compression realizing impressive performance. In this setting, the average compression length on the test data set is approximately equal to[1] $-\frac{1}{M}\sum_{m=1}^{M} \mathrm{ELBO}(x'_m, \theta, \phi)$. Hence, the better the generalization performance (as measured by the negative ELBO on the test set), the better the compression performance. This motivates finding practical ways to improve the generalization of this class of model.

2.1 Two Factors that Affect the Generalization of VAEs

To better understand the generalization performance of VAEs, we re-write the training criterion as

$$\big\langle \mathrm{ELBO}(x,\theta,\phi) \big\rangle_{p_d(x)} = \big\langle \log p_\theta(x) - \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) \big\rangle_{p_d(x)}, \tag{5}$$

where $p_\theta(z|x) \propto p_\theta(x|z)p(z)$ is the true posterior under the model. It is apparent that maximizing the average ELBO is equivalent to maximizing the likelihood $\langle \log p_\theta(x) \rangle_{p_d(x)}$ whilst simultaneously minimizing the KL divergence between the true and approximate posterior. However, similar to equation 2, in practice we only have finite training data and so we resort to an approximate training objective

$$\Big\langle \log p_\theta(x) - \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) \Big\rangle_{p_d(x)} \approx \tfrac{1}{N}\sum_{n=1}^{N} \Big(\log p_\theta(x_n) - \mathrm{KL}(q_\phi(z|x_n)||p_\theta(z|x_n)) \Big). \tag{6}$$

This illustrates that when using finite training data, in addition to the *model empirical approximation*: $\langle \log p_\theta(x) \rangle_{p_d(x)} \approx \frac{1}{N}\sum_{n=1}^{N} \log p_\theta(x_n)$, we also have an *inference empirical approximation*:

$$\Big\langle \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) \Big\rangle_{p_d(x)} \approx \frac{1}{N}\sum_{n=1}^{N} \mathrm{KL}(q_\phi(z|x_n)||p_\theta(z|x_n)). \tag{7}$$

This suggests that for a flexible inference network $q_\phi(z|x)$ this approximation can also over-fit to the training data. More specifically, if we let $\phi^*_N = \arg\min_\phi \frac{1}{N}\sum_{n=1}^{N} \mathrm{KL}(q_\phi(z|x_n)||p_\theta(z|x_n))$, we assume[2] for any training data point $x_n \in \mathcal{X}_{train}$

$$q_{\phi^*_N}(z|x_n) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(z|x_n)||p_\theta(z|x_n)) \equiv q^*(z|x_n). \tag{8}$$

If we want to use $q_{\phi^*_N}(z|x)$ to approximate the test likelihood, we need it to be a good approximation to the true posterior on test data. However, there is no such requirement built into the training procedure. For $x'_m \in \mathcal{X}_{test}$, we can have

$$q_{\phi^*_N}(z|x'_m) \neq \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q(z|x'_m)||p_\theta(z|x'_m)) \equiv q^*(z|x'_m), \tag{9}$$

[1]We use $\log_2$ in the definition of ELBO.

[2]For a flexible inference network, we assume here that there is no amortization gap [4], which means $q_{\phi^*}(z|x)$ could generate optimal $q^*(z|x_n)$ for each training data x_n, also see Section 3.

where $q^*(z|x'_m)$ is the optimal posterior for the test data and $q_{\phi^*_N}(z|x'_m)$ is the posterior learned from the training data. We refer to the difference between the ELBO that uses these two different posteriors as the *inference generalization gap*, defined as

$$\langle \log p_\theta(x,z) - \log q^*(z|x)\rangle_{q^*(z|x)p_d(x)} - \langle \log p_\theta(x,z) - \log q_{\phi^*_N}(z|x)\rangle_{q_{\phi^*_N}(z|x)p_d(x)}. \tag{10}$$

Similar to equation 3, we can approximate the gap using the test dataset $\mathcal{X}_{test}$

$$\frac{1}{M}\sum_{m=1}^{M}\Big(\langle \log p_\theta(x'_m,z) - \log q^*(z|x'_m)\rangle_{q^*(z|x'_m)} - \langle \log p_\theta(x'_m,z) - \log q_{\phi^*_N}(z|x'_m)\rangle_{q_{\phi^*_N}(z|x'_m)}\Big). \tag{11}$$

This generalization gap results from the application of amortized inference. It is important to emphasize that this gap can not be reduced by simply using a more flexible $\mathcal{Q}$. This would only make the KL divergence $\mathrm{KL}(q_\phi(z|x_n)||p_\theta(z|x_n))$ smaller for the training data $x_n \in \mathcal{X}_{train}$ but would not by design encourage better generalization performance on test data.

In conclusion, the generalization performance of VAE style models depends on two factors:

1. **model generalization gap**, where the model is $p_\theta(x) = \int p_\theta(x|z)p(z)dz$;
2. **inference generalization gap**, caused by the amortized posterior $q_\phi(z|x)$.

In the next section we demonstrate how these two gaps will affect the generalization performance of a VAE in practice.

2.2 Contributions of the different Generalization Gaps

For simple data distributions, a powerful VAE can easily over-fit the training data set. To test this, we fit a simple VAE model to the binarized MNIST dataset. Our VAE consists of a 2-layer feed-forward neural network with hidden dimension 500 for both the model network $p_\theta(x|z)$ and inference network $q_\phi(z|x)$. The model has latent dimension 10 and is trained using the Adam optimizer[5] with learning rate $1e-3$. We let $p_\theta(x|z)$ be a Bernoulli distribution. $p(z)$ is the standard Gaussian distribution and the amortized variational distribution $q_\phi(z|x)$ is a Gaussian distribution with diagonal variance. We train the VAE (with amortized inference) for 1000 epochs and save and evaluate the model every 100 epochs. Figure 1a shows that the training BPD[3] decreases during training whereas the test BPD first decreases and then gradually increases, indicating the model has over-fit the training data. This over-fitting is potentially caused by both the model network over-fitting and the inference network over-fitting. These relate to the model and inference generalization gaps respectively.

We notice that the inference generalization gap can be eliminated by knowing the test dataset. Specifically, since $q^*(z|x'_m) \equiv \arg\min_{q\in\mathcal{Q}} \mathrm{KL}(q(z|x'_m)||p_\theta(z|x'_m))$, we can just minimize the KL divergence for the test data $x'_m \in \mathcal{X}_{test}$ with respect to ϕ fixing θ:

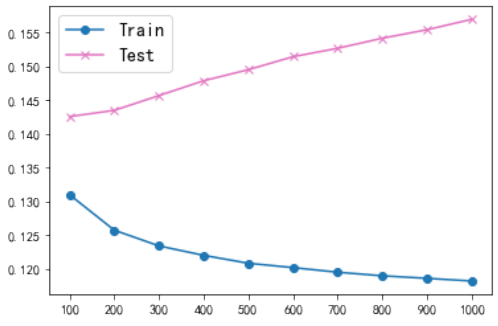
$$\min_\phi \mathrm{KL}(q_\phi(z|x'_m)||p_\theta(z|x'_m)) = -\big\langle \log p_\theta(x'_m,z) - q_\phi(z|x'_m)\big\rangle_{q_\phi(z|x'_m)} + const. \tag{12}$$

Updating ϕ using this objective would result in an optimal inference strategy. To further investigate the contributions of two the types of over-fitting discussed in section 2.1, we propose to use this optimal inference strategy to eliminate the negative effect of the inference generalization gap. This then allows us to isolate the degree to which both the model and inference generalization gap are contributing to over-fitting.

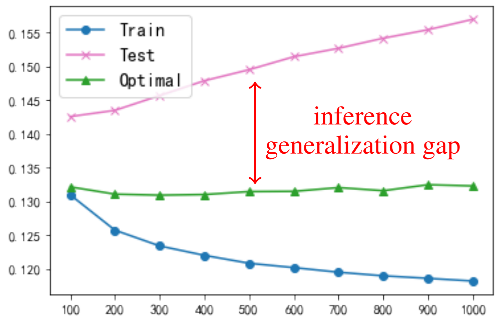
We plot the test BPD using the optimal inference setup, leaving only the effect of the model generalization gap - see figure 1b. The difference between the optimal inference test likelihood curve and the original test likelihood curve illustrates the *inference generalization gap*. We find that after eliminating the inference generalization gap the test BPD is significantly improved and is stable during training. This suggests the model over-fitting is negligible in this experiment and the over-fitting behavior is in fact *dominated by the inference network over-fitting to the training data* in this case.

This is evidence to support that in order to improve the generalization of VAEs, an important challenge is to reduce the inference generalization gap. Although the optimal inference procedure can help eliminate the inference generalization gap, training q_ϕ at test time is not practical in most applications of interest. In the next section, we propose a simple method that can help alleviate the inference generalization gap in amortized inference without having to further train q_ϕ at test time.

[3]BPD represents Bits-per-dimension, which is the negative ELBO with $\log_2$ normalized by the data dimension. Lower BPD means higher likelihood.



(a) Overfitting in VAE



(b) Optimal Amortized Inference

Figure 1: In both figures, x-axis represents the training epoch and the y-axis represents the BPD. In Figure a, we plot the BPD for both training data and test data for every saved model. We find the test BPD is going up while the training BPD is going down, which indicates the VAE model has overfit to the training data. In Figure b, we plot the test results using the optimal inference strategy, which fixes the model parameter θ and only trains the $q_\phi(z|x)$ on the test data using equation 12. This evaluation method eliminates the effect of the inference generalization gap. We can see the results using optimal inference significantly improve the BPD and no longer exhibits the over-fitting behavior. Additionally, the gap between the purple line (amortized inference) and green line (optimal inference) is the *inference generalization gap*.

3 Reducing the Inference Generalization Gap

Recall the empirical approximation of the ELBO in equation 6. Even with a perfect model $p_\theta = p_d$ the second term in the objective $\frac{1}{N}\sum_{n=1}^{N}\mathrm{KL}(q_\phi(z|x_n)||p_\theta(z|x_n))$ can still cause the inference network to over-fit to the training data. A solution when the model is perfect is to replace p_d with p_θ:

$$\big\langle \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))\big\rangle_{p_d(x)} = \big\langle \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))\big\rangle_{p_\theta(x)}. \tag{13}$$

Similar to equation 12, we can fix θ and train ϕ using

$$\big\langle \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x))\big\rangle_{p_\theta(x)} = -\big\langle \log p_\theta(x,z) - \log q_\phi(z|x)\big\rangle_{q_\phi(z|x)p_\theta(x)} + const. \tag{14}$$

The integration $\langle\cdot\rangle_{p_\theta(x)}$ is approximated by Monte-Carlo samples generated from $p_\theta(x)$ instead of $\hat{p}_d(x)$. We refer to this training criterion as *self-consistent training* because a perfect $q_\phi(z|x)$ should satisfy the following consistency condition

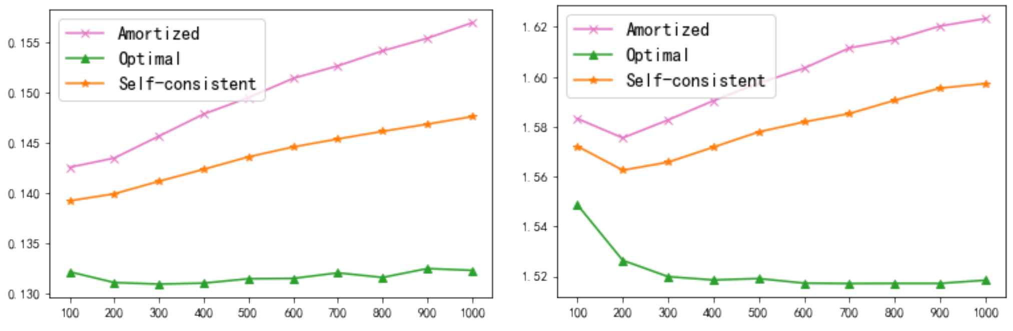
$$\mathrm{KL}\big(q_\phi(z|x)p_\theta(x)||p_\phi(z|x)p_\theta(x)\big) = 0 \Rightarrow \big\langle \mathrm{KL}\big(q_\phi(z|x)||p_\phi(z|x)\big)\big\rangle_{p_\theta(x)} = 0. \tag{15}$$

In practice our model will not be perfect, $p_\theta \neq p_d$. Empirically we find that samples from even a well trained model p_θ may not always be sufficiently like the samples from the true data distribution leading to degradation in the performance of the inference network when trained using the self-consistent criterion. For this reason, within the self-consist objective we propose to use a mixture distribution between the model and the empirical training data distribution as follows

$$\big\langle \mathrm{KL}\big(q_\phi(z|x)||p_\theta(z|x)\big)\big\rangle_{m(x)} \quad \text{with} \quad m(x) \equiv \alpha p_\theta(x) + (1-\alpha)\hat{p}_d. \tag{16}$$

When $\alpha = 0$ then this reduces to the standard approach. When $\alpha = 1$ we recover equation 15. We find that a setting of $\alpha = 0.5$ works well in practice. This balances samples from the true underlying data distribution with samples from the model. This can also be interpreted as a form of training data augmentation to help over-come the over-fitting caused by the application of amortized inference. In contrasts to the optimal inference approach discussed in section 2.1, our approach does not require further training on test data to improve generalization performance.

Empirically, we validate the performance of this approach for both binary and Grey-scale MNIST problems. The model architecture and the training method is the same as described in section 2.1. However, for grey-scale MNIST we use the discretized Logistic distribution as $p(x|z)$. We first train the VAE with the usual amortized inference approach with 1000 epochs and save the model every 100 epochs. We then use the saved models to 1) evaluate on the test data sets, 2) conduct optimal inference by training $q_\phi(z|x)$ on the test data and 3) run self-consistent training using samples from the learned model and the training data before calculating the test BPD. We compare the test ELBO with the standard amortized inference approaches - see figure 2 for results. We find that the self-consistent training approach consistently improves the generalization performance of our models as measured by the negative test ELBO.



(a) Binary MNIST (b) Greyscale MNIST

Figure 2: In both figures, x-axis represents the training epoch and the y-axis represents the BPD. We compare test BPD using amortized inference, optimal inference and amortized inference with our additional self-consistent training step on both Binary MNIST and Greyscale MNIST datasets. We find that although the self-consistent training still has a gap, it improves the generalization performance in both cases without training the $q_\phi(z|x)$ on the test data.

4 Related Work

Recent work [15] first studies the generalization of probabilistic image modeling in the context of the lossless compression. They focus on an auto-regressive model whose likelihood can be computed exactly. Unlike our work where we focus on the *inference related generalization gap*, their study of model generalization is limited to just the *model generalization gap*.

Previous work [4] focus on the *amortization gap* in the context of amortized inference. This gap is caused by learning a network $q^*_\phi(z|x_n)$ to generate posteriors for each input data point x_n rather than learning a posterior q^*_n for each data point x_n individually. This gap can be somewhat alleviated using a larger capacity encoder or decoder model. This amortization gap is fundamentally different from the *inference generalization gap* we discuss in this work since the latter focuses solely on test time generalization but the former problem also exists at training time.

In paper [12], they propose to regularize the encoder to prevent the inference network over-fitting to the training data and to improve the test likelihood. Such regularization techniques also affects the parameters of the model (decoder) during training wheres our proposed self-consistent training leaves the model unchanged. On the other hand, the self-consistent training makes the assumption that that trained model is approximately correct, which might not be true when the data distribution is very complex. We thus leave a further comparison between these methods to future work.

Paper [2] also proposed to use the samples from the model to train the encoder, which aims to make the learned representations more robust to adversarial attacks, whereas the motivation of our self-consistent criterion is to improve generalization performance.

5 Conclusion

In this work we first highlight the factors that contribute to poor generalization performance in latent variable models that employ amortized inference. We then empirically demonstrate the dominance of the introduced inference level generalization gap in explaining over-fitting behavior in a simple image modeling task. Finally we propose *self-consistent training* as a practically useful method to improve generalization performance and demonstrate it's utility on some standard image modeling tasks.

References

[1] F. V. Agakov and D. Barber. An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566. Springer, 2004.

[2] A. T. Cemgil, S. Ghaisas, K. Dvijotham, S. Gowal, and P. Kohli. Autoencoding variational autoencoder. *Neural Information Processing Systems*, 2020.

[3] E. Challis and D. Barber. Affine independent variational inference. *Advances in Neural Information Processing Systems*, 25:2186–2194, 2012.

[4] C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086. PMLR, 2018.

[5] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International conference on machine learning*, 2015.

[6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2013.

[7] F. Kingma, P. Abbeel, and J. Ho. Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. In *International Conference on Machine Learning*, pages 3408–3417. PMLR, 2019.

[8] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *International conference on machine learning*, pages 1445–1453. PMLR, 2016.

[9] D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[10] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[11] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and variational inference in deep latent gaussian models. In *International Conference on Machine Learning*, volume 2, page 2. Citeseer, 2014.

[12] R. Shu, H. H. Bui, S. Zhao, M. J. Kochenderfer, and S. Ermon. Amortized inference regularization. *Neural Information Processing Systems*, 2018.

[13] J. Townsend, T. Bird, and D. Barber. Practical lossless compression with latent variables using bits back coding. *International Conference on Learning Representations*, 2019.

[14] J. Townsend, T. Bird, J. Kunze, and D. Barber. Hilloc: Lossless image compression with hierarchical latent variable models. *International Conference on Learning Representations*, 2020.

[15] M. Zhang, A. Zhang, and S. McDonagh. On the out-of-distribution generalization of probabilistic image modelling. In *Neural Information Processing Systems*, 2021.