Automatic Differentiation Variational Inference with Mixtures: Supplementary Material

A Theorems and proofs

Theorem A.1. *$\mathcal{L}^T_{SIWAE}$ is a lower bound on the evidence $\log p(x)$.*

Proof.

$$
\begin{aligned}
&\log p(x) = \\
&= \log \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}_{z_t \sim q_\phi(z|x)} \left[\frac{p(x|z_t) r(z_t)}{q_\phi(z_t|x)} \right] \\
&= \log \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_{k,\phi}(x) \mathbb{E}_{z_{kt} \sim q_{k,\phi}(z|x)} \left[\frac{p(x|z_{kt}) r(z_{kt})}{q_\phi(z_{kt}|x)} \right] \\
&= \log \int \cdots \int \left[\prod_{t=1}^{T} \prod_{k=1}^{K} dz_{kt} q_{k,\phi}(z_{kt}|x) \right] \times \\
&\qquad \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_{k,\phi}(x) \frac{r(z_{kt}) p(x|z_{kt})}{q_\phi(z_{kt}|x)} \\
&\geq \int \cdots \int \left[\prod_{t=1}^{T} \prod_{k=1}^{K} dz_{kt} q_{k,\phi}(z_{kt}|x) \right] \times \\
&\qquad \log \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_{k,\phi}(x) \frac{p(x|z_{kt}) r(z_{kt})}{q_\phi(z_{kt}|x)} \\
&= \mathbb{E}_{\{z_{kt} \sim q_{k,\phi}(z|x)\}_{k=1,t=1}^{K,T}} \Bigg[\\
&\log \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} \alpha_{k,\phi}(x) \frac{p(x|z_{kt}) r(z_{kt})}{q_\phi(z_{kt}|x)} \Bigg] \\
&\equiv \mathcal{L}^T_{\text{SIWAE}}(\phi)
\end{aligned}
$$

□

Theorem A.2. *When $K > 1$, $\mathcal{L}^T_{SIWAE}$ is a tighter lower bound than $\mathcal{L}^T_{IWAE}$.*

Proof.

$$
\begin{aligned}
&\mathcal{L}^T_{\text{SIWAE}}(\phi) \equiv \\
&\equiv \mathbb{E}_{\{z_{kt}\sim q_{k,\phi}(z_k|x)\}^{T,K}_{t=1,k=1}}\Bigg[\\
&\qquad \log \frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{K}\alpha_{k,\phi}(x)\frac{p(x|z_{kt})r(z_{kt})}{q_\phi(z_{kt}|x)}\Bigg] \\
&\geq \sum_{k=1}^{K}\alpha_{k,\phi}(x)\mathbb{E}_{\{z_{kt}\sim q_{k,\phi}(z_{kt}|x)\}^{T,K}_{t=1,k=1}}\Bigg[\\
&\qquad \log \frac{1}{T}\sum_{t=1}^{T}\frac{p(x|z_{kt})r(z_{kt})}{q_\phi(z_{kt}|x)}\Bigg] \\
&= \mathbb{E}_{\{z_t\sim q_\phi(z|x)\}^{T}_{t=1}}\left[\log \frac{1}{T}\sum_{t=1}^{T}\frac{p(x|z_t)r(z_t)}{q_\phi(z_t|x)}\right] \\
&\equiv \mathcal{L}^T_{\text{IWAE}}(\phi)
\end{aligned}
$$

□

B Experimental Details

For all experiments, we assume that the data was generated according to the following graphical model:

$$
\begin{aligned}
&\text{for } i = 1 \dots n: \\
&\quad Z_i \sim r(Z) && (3)\\
&\quad y_i \sim p(Y_i|Z_i) && (4)
\end{aligned}
$$

Here, for each instance of the data, we sample the local representation of this data z_i from its marginal distribution $r(Z)$ and draw the observed data y_i from the conditional distribution $p(Y_i|Z_i)$. In this section, we present specific details of the implementation of each of these experiments, including training and evaluation procedures.

B.1 Single Column MNIST Classification

Architecture. For our experiments, we use an MLP architecture with 4 layers of 128 hidden units and ELU activation functions [Clevert et al., 2015] for the encoder. The last layer predicts the parameters for a distribution over a two dimensional latent variable (in subsection C.2, we run the same experiment but with a one dimensional latent variable). For all models, we use a mixture of K multivariate normal distributions with full covariance with mixture weights as a learnable parameter which is predicted by the encoder. For models with $K = 1$, this reduces to a single multivariate normal distribution with no learnable mixture weights. For the decoder, we use an affine transformation that outputs the logits for a categorical distribution. We use such a simple architecture for decoder to encourage the encoder to capture potentially multimodal information about the class of an image. For our prior distribution $r(z)$, we use a trainable mixture of Gaussians, although we found the prior makes relatively little difference in the final results.

Training Procedure. For a single component model, we optimize both the traditional evidence lower bound (ELBO), as well as the importance weighted estimate of the evidence (IWAE). For the mixture models, we use stratified sampling to compute the ELBO (SELBO), as well as the Stratified-IWAE (SIWAE) derived in Section 2. We use $K = [1, 2, 5, 10]$ for the number of mixture components, and $T = [1, 2, 5, 10]$ for the number of samples drawn *per component*. To regulate the information content of the posterior, we use a $\beta = 0.05$ penalty on the KL divergence term (and the equivalent term in the SIWAE objective), as used in Higgins et al. [2017]. Because one-column MNIST does not have an established benchmark, we also train two deterministic models to use as baselines: (1) a "pyramid" MLP with 5 layers of 256 hidden units to approximate the peak deterministic accuracy, and (2) a "bottleneck" MLP with the same architecture as our VIB models, therefore containing a two dimensional "latent space." All models were trained for 50 epochs using the Adam optimizer Kingma and Ba [2014] with a learning rate of 0.001 which was decayed by 0.5 every 15000 train steps. When training SELBO models, T refers to the number of samples drawn to compute the Monte-Carlo estimate of the objective.

Evaluation. To evaluate the accuracy of the model, we first need the posterior predictive distribution. We sample the posterior predictive by decoding 10^4 samples from $q_\phi(z|x)$ and averaging the class probabilities returned by each sample. This marginalizes over the uncertainty in the latent variables and if our prior beliefs are correct, nominally produces calibrated probabilities. From these probabilities, we take the highest-probability class, and consider that the prediction of the model. Accuracy is then defined as the number of correct predictions divided by the total number of examples in the test set.

We also compute the *Expected Calibration Error* [ECE; Guo et al., 2017]. For this, we decode 1000 samples from the posterior and compute the average probability of each class. We take the model prediction to then be

$$\hat{y} = \text{argmax} p(y|z) \qquad (5)$$

This prediction is labeled correct if it is equal to the true class label y, otherwise it is labeled incorrect. In addition to checking if each prediction is correct, we also get the predicted confidence for the true class $p(y_{true})$. We then rank our data and divide into 10 bins such that each bin contains 10% of the examples, ranked by

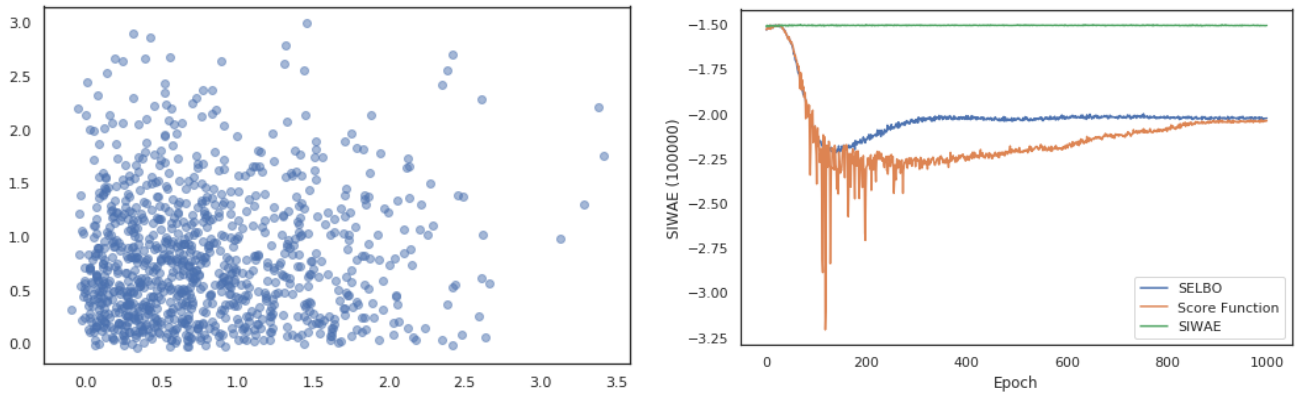


Figure B.1: On the left is a toy dataset generated by sampling $z \sim \mathcal{N}(0, I); x \sim \mathcal{N}(|z|, \sigma^2 I)$. On the right are SIWAE values at each epoch while training posteriors using SELBO, SIWAE, and score function estimators of the evidence. Due to mixture components collapse, the SELBO and score function posteriors achieve lower values of SIWAE.

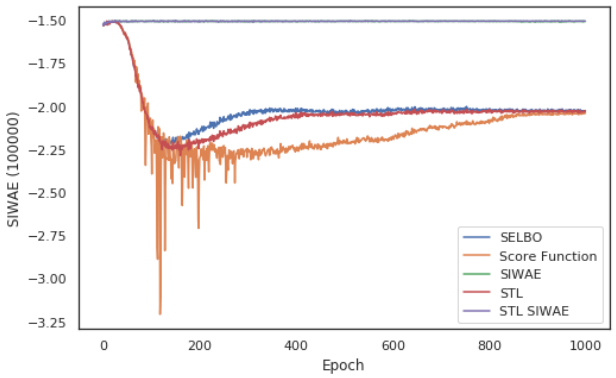
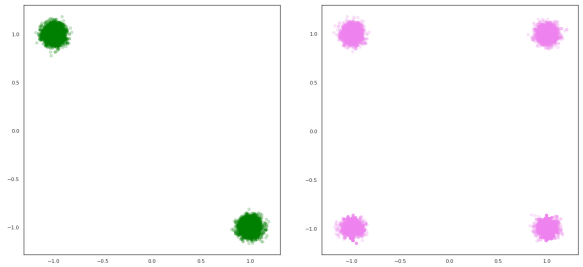


Figure B.2: Training results for the toy experiment with the addition of "sticking-the-landing" versions of SELBO and SIWAE. We observe no significant difference between the final training SIWAEs of STL-SELBO vs. SELBO (-2.026 vs. -2.024 respectively) and STL-SIWAE vs. SIWAE (-1.505 and -1.505 respectively)



(a) STL-SELBO (b) STL-SIWAE posterior

Figure B.3: Samples from of the learned (implicit) posteriors for the observed data point $(1, 1)$ for the STL-SELBO and STL-SIWAE [Roeder et al., 2017].

confidence in the true class $p(y_{true})$. The confidence of a bin is computed as $p^{bin}(y_{true} = \frac{1}{n}\sum_i^n p(y_{true}^{(n)})$. The probability of the truth for a given bin is given by the fraction of predictions in the bin which were correct. The expected calibration error is then defined as the average absolute value of difference between the confidence in a bin and the probability of correctness in that bin.

In the main paper, we also included no discussion of the relative computational cost between methods. Because SELBO/SIWAE require stratification over components, they by design have to use more samples than ELBO/IWAE models, which only need to sample from one mode and therefore have T samples (compared to $K \times T$). SELBO and SIWAE have comparable cost to each other, as they only run different mathematical operations when computing the loss (a sum versus a log-sum-exp).

B.2 Single Column MNIST VAE

Architecture. For the encoder, we used the same architecture as in subsection 4.2. For the decoder, we used the same architecture as used in the Tensorflow Probability GitHub example with a few small differences. In order to ensure that gradients were passed to the encoder early on in training, we used a skip connection, represented as a single affine layer to project the latent space directly to the output space. We also experimented with both the affine and tensorflow probability decoders separately, and found that the final results did not depend on the decoder architecture, though the overall performance of all models was better for the nonlinear decoder architectures. For the prior, we used a mixture distribution with 200 components.

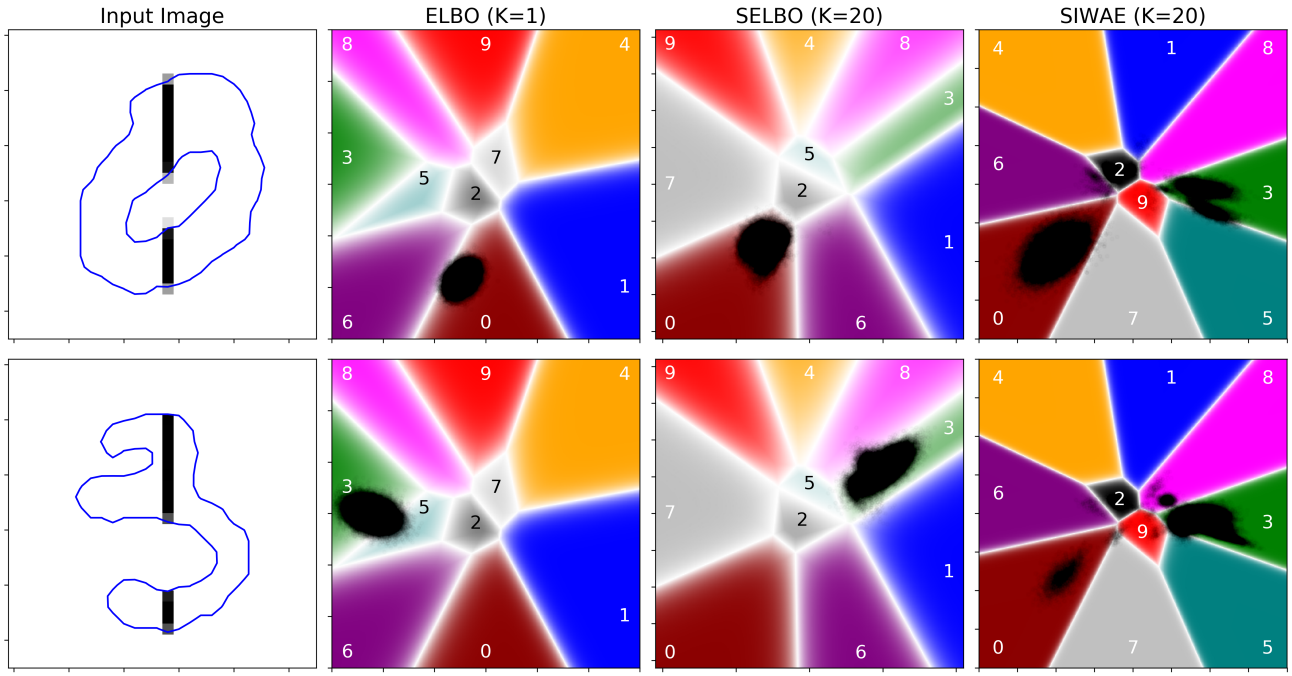


Figure B.4: Similar to Figure 4, but for two examples where the input to the model is extremely similar even though the inputs are from different classes. We find that while all models infer the correct class, models trained with SIWAE are better suited to recognize the similarity between these two images, assigning some probability to the other class.

Training. For a single component model, we optimize both ELBO and IWAE, and for mixture models we use the corresponding stratified loss. We experiment with $k = [1, 2, 5, 10]$ for the number of mixture components and $T = [1, 2, 5, 10]$ for the number of samples drawn per component. All models are trained for 100 epochs using Adam with a learning rate of 0.0001, decayed by 0.5 every 15000 training steps.

Evaluation. We evaluate models as a function of k and T. To ensure consistency in the evaluation, we used 100 samples from the posterior of each model to evaluate an IWAE estimate of the evidence.

B.3 Burn-in against the prior

Typical initialization schemes attempt to facilitate gradient backpropagation by ensuring that the first two moments of the activations remain approximately 0, and 1, respectively. We found that these initialization schemes don't typically produce a posterior distribution tuned reasonably to the prior. This violates our intuition, as the posterior in the absence of evidence should be identical to the prior. Furthermore, even if the model were initialized such that the posterior and prior were aligned, the alignment would quickly be broken by large bulk gradients being given to the posterior from the likelihood. We observed that this was a consequence of the decoder being not well tuned to the dataset at initialization. For example, the edge pixels in MNIST are essentially all zero, but the initial decoder predicts a uniform distribution over these pixels. The model would therefore systematically shift the posterior to compensate for the poor initialization of the likelihood. This bulk shift early on in training often produced a final posterior that was well tuned to the likelihood, but poorly tuned to the prior.

Our simple solution to this problem was to burn in the decoder so that the initial decoder distribution was reflective of our prior distribution over the dataset. To do this, we fed samples from the prior to the decoder, and attempted to maximize the expected log-likelihood of the images given the prior samples $p(x|z)$. Furthermore, because the encoder could often be improperly tuned against the prior (a worsening problem in higher dimensionality), we attempted to uniformly spread the encoder across the prior. This was accomplished by minimizing the cross entropy between the prior and the posterior $H_x(r, q) = \mathbb{E}_{z\sim r(z)} q(z|x)$. This optimization was performed jointly for both the encoder and decoder variables, with the prior held fixed. Our specific procedure to do this was as follows.

1. Draw samples from the prior distribution $r(z)$.
2. For the prior samples, compute the expected

log-likelihood $\mathbb{E}_{z_i \sim r(z), x_i; i=1\ldots M}[\log p(x_i|z_i)]$ over a batch of images in the training set.

3. For the prior samples, also compute the cross entropy from the posterior $H_x(r(z_i, q(z_i|x_i))$.

4. Compute and apply the gradients of the loss $\mathcal{L} = \mathbb{E}_{z_i \sim r(z), x_i; i=1\ldots M}[-\log p(x_i|z_i) - \log q(z|x)]$ for all encoder and decoder variables.

5. Repeat until converged.

In practice, we found that convergence was typically achieved within a single epoch, so for simplicity we ran burn-in for a single epoch. This produced a decoder which, when fed samples from the prior, would produce predictions consistent with random samples of each pixel from the dataset. Note that prior samples from the burned in decoder do not resemble images from the dataset, but merely draw from a simplified estimate of the prior distribution for each pixel. At the same time, this burn in procedure matches the encoder to the prior, which makes some sense, given that we initially only know samples from the prior. We think that this burn in procedure is a worthwhile practice for initializing latent variable models.

C Additional Experiment Results

C.1 Toy Problem

In Figure B.1 we visualize the data from the toy experiment and the training curves for the ELBO estimators.

In addition to the experiments presented in the main body of the paper, we also ran a comparison to the "sticking the landing" (STL), pathwise derivative estimator, which results in reduced variance in the model gradients (with a potential increase in bias Tucker et al. [2019]). Our main interest lies in determining if the STL gradient estimator is itself sufficient for fitting multimodal posteriors, or if the use of SIWAE is truly necessary for inferring multimodality. We ran our test on the toy problem using STL to evaluate both the SELBO and the SIWAE losses. We show the evidence, as measured by a 10^5 sample SIWAE as a function of training epochs in Figure B.2. We find that for both SELBO and SIWAE, the model evidence is unchanged by using the STL gradient estimator, indicating that STL does not help in converging to a better model. Furthermore, in Figure B.3, we show samples from the learned posterior. We find that using SELBO, even with STL, results in a model which does not discover all modes in the posterior. The fact that SELBO and SIWAE give the same results as STL-SELBO and STL-SIWAE suggests that it is the SIWAE loss itself, rather than the gradient estimator, that is providing the necessary ingredients for detecting multimodality. However, we speculate that STL may offer more relative improvement in situations where the bias introduced by SELBO is low compared to the variance introduced by SIWAE.

In the main text of the paper, we showed the latent space distribution for an image wherein the ambiguity introduced by the use of a single column in the inference resulted in a multimodal latent space. Furthermore we showed that SIWAE was able to detect and capture this multimodality much better than ELBO or SELBO, which either are structurally unequipped to do so (ELBO), or which are penalized for doing so (SELBO). To show that the capacity for multimodality aids in the interpretability of our model, consider the images shown in Figure B.4. Both images, while having quite different true appearances, appear nearly identical when viewed as only their center column. Therefore, a model should classify this pair as "either a 0 or a 3", since both of these classes have this appearance. However, this is not observed when SELBO is used. The model (correctly) predicts a zero for the top image, and a three for the bottom image, with no indication that the other is a possibility. In contrast, the SIWAE model also predicts the correct class, but correctly assigns a non-negligible fraction of its samples to the other class. In this sense, uncertainty is measured in the latent space itself using the posterior distribution.

We also included an ablation study to compare the gains from using SIWAE compared to alternative means of improving gradient flow in VAE models. For this study, we trained models using 8 different losses to compare their performance. In particular, we trained models using the C-IWAE and M-IWAE from Rainforth et al. [2018], which reduce the gradient variance and purportedly result in better models than those trained with IWAE. For both of these, we use an appropriately modified SIWAE for $K > 1$. We also train models using the “sticking the landing” gradient estimator from Roeder et al. [2017] which was also suggested to improve variance in the gradient estimates. We added the stl estimator to SELBO, SIWAE, and C-IWAE to examine if the stl estimator has an effect on performance compared to models which use the naive implementations of the gradient. Finally, we compare SIWAE to the PAC^m-Bayes loss from Morningstar et al. [2020], which maximizes a bound on the log posterior predictive probability: $\mathcal{L}_{PPD} = -\log\left(1/TK \sum_{t,k=1}^{T,K} p(x|z)\right) + D_{\mathrm{KL}}(q(z|x), r(z),)$. We trained all of these losses as a function of K and T to compare the relative gains in performance with additional components or samples. For all of these, we followed the same training procedure as SIWAE and SELBO, as mentioned in Appendix B.

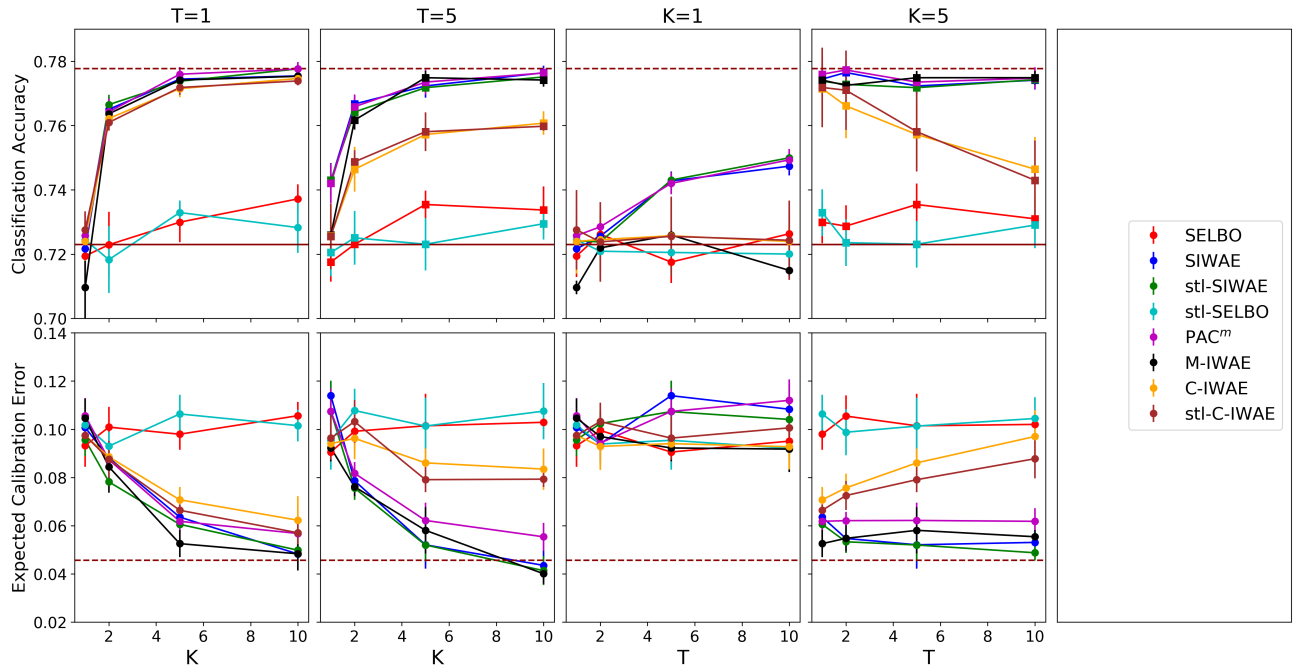


Figure C.1: Accuracy and calibration error of a model as a function of K and T for fixed values of K and T. We compared multiple different losses to study the effect of the loss on performance, focusing on different means of reducing gradient variance such as "sticking the landing" Roeder et al. [2017], and alternatives discussed in Rainforth et al. [2018].

We show the performance as a function of K and T for several choices of T and K respectively in Figure C.1. In the main paper, we already showed that SIWAE exhibits increasing classification accuracy and decreasing expected calibration error as a function of K. Here we also observe that the "sticking the landing" gradient estimator offers little benefit, since the accuracy returned from all corresponding naive implementations of each loss have comparable performance. This result agrees with our results from Figure B.3, where we show that this estimator does not help SELBO to recover multimodality. We also find that C-IWAE does not offer improvement over SIWAE, with both the naive and stl versions performing equally to alternatives when $T = 1$, and proportionally worse otherwise, though they both still appear to have improved performance as a function of K. It appears that C-IWAE also exhibits worsening performance as a function of T for $K = 5$. We also observe that M-IWAE exhibits similar performance to SIWAE as a function of K, but observes no improvement with T. This behavior is expected, as our M-IWAE implementation follows SIWAE over mixture components, but follows SELBO over samples. Finally, we also observe that the PAC^m objective exhibits comparable performance to SIWAE as a function of K and T. We also expect this behavior, since both objectives allow for similar degrees of mode exploration. From all of these results, we draw the conclusion that the SIWAE objective provides advantages to models which cannot be replicated simply by stratification over mixture components, or by reducing gradient variance either by blocking gradient flow through noisy parts of the loss or by softening the tightness of the bound.

C.2 One-dimensional latent variable

We ran the same MNIST classification experiment using a one-dimensional latent variable. In general, fitting a one-dimensional latent variable should result in an appreciable drop in accuracy because a single-dimensional bottleneck allows for a maximum of two decision boundaries for a particular class, and therefore forces a latent representation which becomes multimodal in the presence of any complex structure in the uncertainty. Therefore, a reasonable expectation is that this should force a model trained with SELBO to learn distinct modes in the encoder. However, we experimented with training for this objective using multiple components as well as with a single component, and in all cases achieve an accuracy of 51% or lower, which is substantially worse than can be reached in two dimensions. By dissecting this model, we see again that the model reduces to a single mode in the posterior, either by assigning all of the component weights to a single mode, or by merging all of the separate modes together (Figure C.2). This strongly suggests that the SELBO

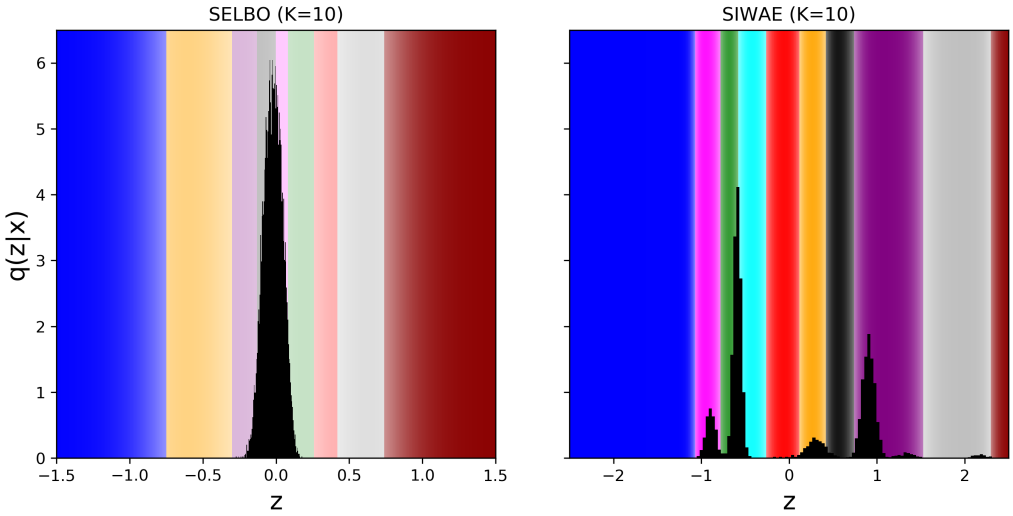


Figure C.2: Similar to Figure 4, but using a one-dimensional latent variable. The input example is the same as in Figure 4. The left panel shows the latent representation found by optimizing the SELBO objective. Only a single mode is identified in the latent space. In contrast, optimizing the SIWAE objective produces a latent representation with multiple distinct modes.

objective actively opposes the formation of multiple modes in the posterior.

Using the SIWAE objective instead of the SELBO, we see our accuracy climb to 76%, nearly equivalent to the peak accuracy in two dimensions. We also find that using the SIWAE objective with only a single component (.e. IWAE) outperforms the traditional ELBO substantially as well, achieving 63% accuracy. However, there is still a substantial gap between the IWAE model and the SIWAE model. All of the probabilistic models outperform a deterministic model in this case, which achieves a peak accuracy of merely 25%.

C.3 Single Column MNIST VAE

One qualitative metric that helps to asses the performance of VAEs, independent of quantitative metrics such as the log-evidence, which may not always be entirely informative, is the appearance of images drawn from the model. This can be examined in multiple ways. One can draw samples from the generative model to see if they appear qualitatively similar to images from the dataset. Alternatively, one can generate reconstructions using samples from the inference model to see if the inference appears reasonable. To assess the performance of SIWAE compared to SELBO, we found it more reasonable to examine the inference model.

To generate templates of each mode from the posterior, we passed the mode of the encoder component distributions to the decoder, and took the mode of each decoded image. These are shown in Figure C.3.

We find that all modes from SELBO make roughly the same prediction, showing that the modes have collapsed together. We also find that these modes often do not capture the correct appearance of the input data over any component of the encoder. In some cases, this may be an example of the decoder trying to "hedge its bets" to make up for the inability of SELBO to recover multimodality, and therefore predicting nearly 0.5 for pixels which have competing explanations. However, it should be pointed out that this cripples the generative model, as the samples produced are of lower quality than they could otherwise be if uncertainty were represented correctly.

In contrast, SIWAE does not encounter issues with collapsing modes, and produces multiple different explanations for each instance fed to the encoder. Notably, we find that for more unambiguous inputs (e.g. zeros), the encoder produces multiple template images from the correct class, but having different stylistic appearance. For images which may be explained by multiple different classes, we find that the modes produce a census of the potential classes. We find that at least one mode will typically provide a good explanation for the output data, with some exceptions occurring for rarer images (such as a crossed seven, which only occurs in roughly 10% of sevens). For the same reason, we suspect that samples drawn from the generative model will exhibit qualitative appearance more indicative of examples from the dataset. To this end, the ability to represent multimodality has prevented the generative model from being hindered by the inference model, as is observed in models using SELBO.

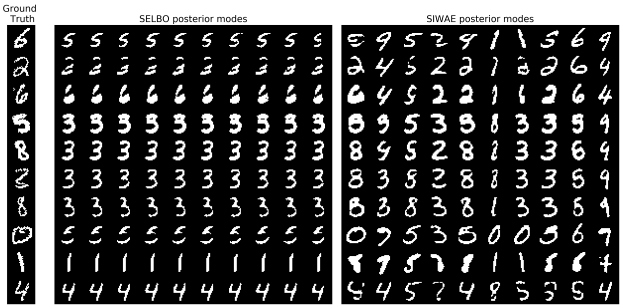


Figure C.3: A visualization of modes from the posterior distribution of models trained with SELBO, compared to models trained with SIWAE. The leftmost column shows the true image, from which the centermost column was fed to the encoder. The central block of images shows the reconstructed modes of the posterior for a model trained with SELBO. The rightmost block of images shows reconstructed modes from the posterior of a model trained with SIWAE. Each column corresponds to a different mode in the posterior. The SELBO modes all appear identical, suggesting that the model is not leveraging multimodality. In contrast, the SIWAE models learn a diverse assortment of modes, which offer competing explanations of the output data. Note that at least one of the SIWAE modes provides an accurate description of the data, while the same cannot be said for SELBO modes.

C.4 Single Column FashionMNIST VAE

We also replicated our single column VAE experiments using the FashionMNIST dataset Xiao et al. [2017]. For this experiment, we replicated the experimental setup for the Single Column MNIST VAE, but with several differences. First, instead of a Bernoulli distribution for the likelihood, we use a Logit-Normal distribution. This fits better with the observed FashionMNIST images, which take on approximately continuous values, as compared to MNIST which is binarized to be either 0 or 1. For this, we transform the pixel values of the output image into the range 0.001 to 0.999, which avoids any instability brought about by pixels at the edge of the support of the distribution. To implement our likelihood, we use a transformed Normal distribution, and use the Sigmoid bijector to transform this distribution. We further use a single shared variance for all images, which we constrain to the positive real numbers and allow to be learnable. We find that this resulted in better output reconstructions, due to the reduced variance in the output images. We also use a 16 dimensional latent space, since we expect that the information content of each image is a lot richer for FashionMNIST than for MNIST. Similar to our experiments with MNIST, we evaluate performance using the log-evidence, as measured using a 100 sample IWAE estimate.

We show the log-evidence as a function of components and samples in Figure C.4. We observe that SIWAE causes rapid improvement in the log-evidence as a function of bpth K and T, while SELBO does not appear to observe any improvement with either. This follows our results as observed in Section 4.3, where we also see SIWAE offer improvements where SELBO does not. This offers further confirmation of the advantages of SIWAE as compared to SELBO.

C.5 Single Column CIFAR-10 VAE

We further replicated our single column VAE experiments using the CIFAR-10 dataset Krizhevsky et al. [2009]. Here, we replicated our problem setup for FashionMNIST, where we used a Logit-Normal distribution with learnable shared variance for our likelihood, a multivariate normal distribution for our posterior, and a mixture of multivariate normal distributions for the prior. We scaled the output image pixel values to the range 0.001 to 0.999, and the input pixel values from -1 to 1.

We show the log-evidence as a function of the number of components and samples in Figure C.5. As we observed with MNIST and FashionMNIST, we also observe that the log-evidence improves with K and T on CIFAR-10. This further indicates that SIWAE allows for an increased model capacity to model data which is particularly uncertain.

C.6 Full Image VAE

Our previous experiments all indicate that SIWAE offers advantages over SELBO when the input data does not contain sufficient information to unambiguously determine the output quantity. However, when the input data is not ambiguous, these advantages are no longer present and SIWAE may therefore offer fewer relative improvements compared to SELBO. At the same time, experiments comparing IWAE to ELBO indicate that losses like SIWAE may also exhibit higher variance in the gradients, which may affect the training dynamics of the model, resulting in worse outcomes Rainforth et al. [2018]. While our previous experiments have shown that higher variance in the gradients is overcome by the ability to successfully leverage multimodality when it exists, it is logical that higher variance gradients could become a detriment to the relative performance when multimodality offers no advantages. We therefore expect that SELBO should perform equal to or better than SIWAE in clean and simple problems where there is no need for multimodality in the latent representation of the data.

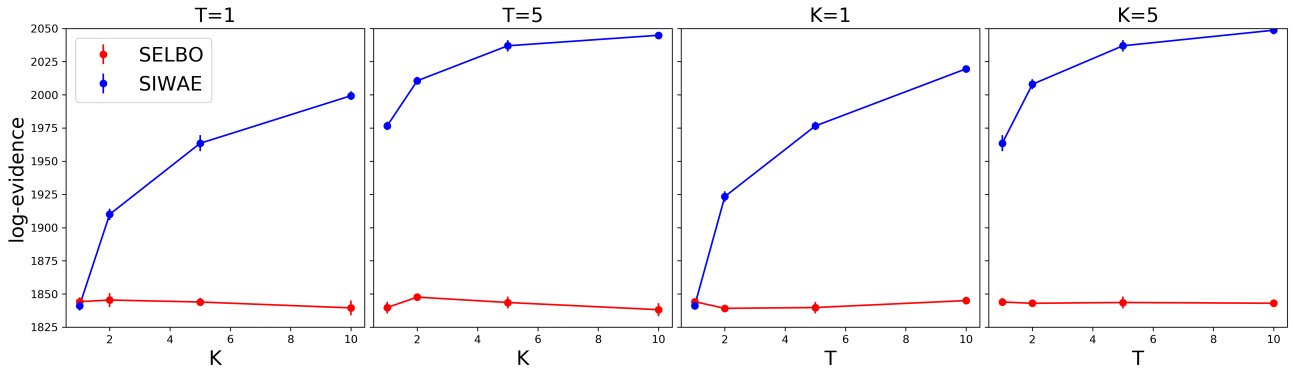


Figure C.4: Log-evidence as a function of K and T for fixed values of T and K, evaluated on the FashionMNIST dataset. Models trained with ELBO and SELBO are shown in red, while models trained with IWAE and SIWAE are shown in Blue.

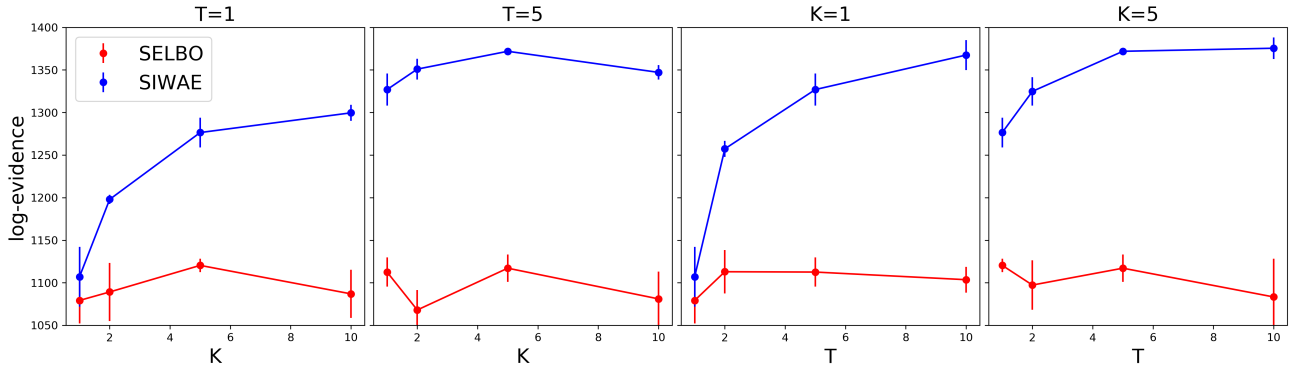


Figure C.5: Log-evidence as a function of the number of mixture components K or samples T, at several fixed values of T and K, evaluated on the CIFAR-10 dataset. Blue points show SIWAE, which outperforms SELBO, shown in red.

To that end, we trained a VAE for 100 epochs on the standard binarized MNIST benchmark dataset, with no corruption of the inputs. Similar to our previous experiment, we evaluate performance as a function of K and T, and the dimensionality of the latent space using a 100 sample SIWAE estimate of the evidence. The training procedure follows that outlined in Appendix B.

We find that when the latent space is low dimensionality, the results are similar to our results from previous experiments. Model performance is improved by using SIWAE instead of SELBO, with the performance improving by increasing either K or T (though the improvement with T is ambiguous). This makes sense, as the encoder is able to overcome the limitations imposed by low dimensionality by using multimodality to represent complex nonlinear structure. In higher dimensionality however, we find that SELBO performs better than SIWAE. This also makes sense intuitively: as the number of dimensions increases, so too does the number of ways in which two unimodal entities can differ. Therefore, the advantages that multimodality provides in low dimensions no longer exist as the dimensionality gets sufficiently large. We therefore expect that using SELBO in larger encoding spaces gives qualitatively better results, though this comes at the cost of explainability. In this regard we present SIWAE and SELBO as two different tools enabling exploration of two different regimes.

C.7 MNIST Style Modality

Thus far, we have shown that when input information to the encoder is limited, SELBO is unable to offer competing explanations for data. We have shown in subsection 4.2 that this is a detriment to model calibration and we have also shown in subsection 4.3 that this limits generative model performance. In these two experiments, the model effectively had to represent class-specific explanations as each mode in the posterior. However, equally interesting is if multimodality can be used to represent style in images.

To test this, we trained a VAE on MNIST, where the

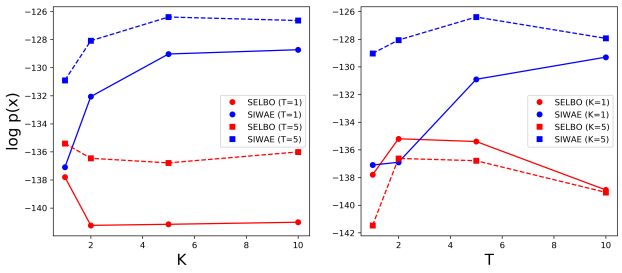


Figure C.6: Same as Figure 6, but for a model given the full MNIST image as an input, and with a 2 dimensional latent space. We find that SIWAE models improve with increasing either K or T, while SELBO models appear to have ambiguous improvement with either K or T.

output target was a randomly chosen image from the same class as the input. This effectively gives the model unambiguous information as to the class, but is completely ambiguous with respect to style. Our hypothesis is that because SIWAE can provide multiple explanations for an input, that it will produce multiple images with different styles. SELBO meanwhile would be penalized for producing multiple explanations, and would therefore produce a single fuzzy image for the output.

In Figure C.7, we show the decoder means of the encoder style modes learned by SELBO and SIWAE models. As expected, we find that SELBO learns only a single style mode, with all 5 possible encoder components producing roughly the same image, indicating that they have collapsed together. Furthermore, the single mode learned by SELBO appears fuzzy, indicating that uncertainty in the output pixels is being explained by the decoder. This makes intuitive sense: SELBO penalizes any posterior mode for providing an explanation that is incorrect, even if that explanation is reasonable. The model therefore compensates by learning only 1 explanation, but making that explanation as reasonable as possible. However, the decoder can only represent uncertainty on each pixel individually, so in making a "reasonable" explanation, it can only make an explanation that is a blurred combination of all digits.

In contrast, when trained with SIWAE, each of the posterior modes produces a different explanation for the data. These different explanations correspond to different styles of each digit. This corresponds to the different styles of ones in the MNIST training set. By allowing the posterior to provide multiple explanations, the decoder produces outputs which are less uncertain. This not only results in improved visual appearance of the outputs, but also shows that SIWAE is able to represent more complex forms of uncertainty in the

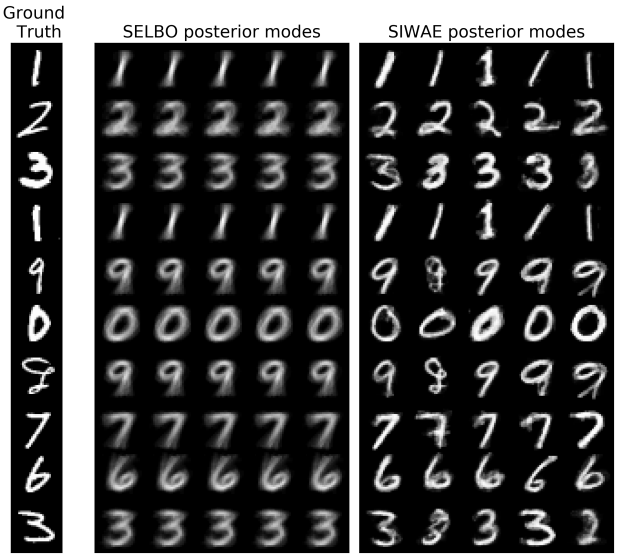


Figure C.7: Posterior modes learned by SELBO and SIWAE when trained to predict a random image from the same class. SELBO learns only to infer the class of the image, producing a fuzzy reconstruction that is able to explain all different styles from that class simultaneously. SIWAE instead learns to encode multiple different styles for each image. This results in perceptually sharper reconstructions, and also in a better capture of uncertainty in the data.

posterior predictive distribution.

In the main text of the paper, we presented several results with regard to VAE models which were given the full image as an input. Here we will show the full details informing these results. The first result was that in low dimensionality, SIWAE models outperformed SELBO models, and exhibited improving performance as a function of K and T. This is shown in Figure C.6. For SELBO, we do not find a corresponding improvment, as the $K = 1$ model outperformed $K > 1$. The origins of this are unclear. Furthermore, we find that SELBO does not appear to exhibit strong dependence on T.