# The Thermodynamic Variational Objective

**Vaden Masrani**[1], **Tuan Anh Le**[2], **Frank Wood**[1]
[1]Department of Computer Science, University of British Columbia
[2]Department of Brain and Cognitive Sciences, MIT

## Abstract

We introduce the thermodynamic variational objective (TVO) for learning in both continuous and discrete deep generative models. The TVO arises from a key connection between variational inference and thermodynamic integration that results in a tighter lower bound to the log marginal likelihood than the standard variational evidence lower bound (ELBO) while remaining as broadly applicable. We provide a computationally efficient gradient estimator for the TVO that applies to continuous, discrete, and non-reparameterizable distributions and show that the objective functions used in variational inference, variational autoencoders, wake sleep, and inference compilation are all special cases of the TVO. We use the TVO to learn both discrete and continuous deep generative models and empirically demonstrate state of the art model and inference network learning.

## 1 Introduction

Unsupervised learning in richly structured deep latent variable models [1, 2] remains challenging. Fundamental research directions include low-variance gradient estimation for discrete and continuous latent variable models [3–7], tightening variational bounds in order to obtain better model learning [8–11], and alleviation of the associated detrimental effects on learning of inference networks [12].

We present the thermodynamic variational objective (TVO), which is based on a key connection we establish between thermodynamic integration (TI) and amortized variational inference (VI), namely that by forming a geometric path between the model and inference network, the "instantaneous ELBO" [13] that appears in VI is equivalent to the first derivative of the potential function that appears in TI [14, 15]. This allows us to formulate the log evidence as a 1D integration of the instantaneous ELBO in a unit interval, which we then approximate to form the TVO.

We demonstrate that optimizing the TVO leads to improved learning of both discrete and continuous latent-variable deep generative models. The gradient estimator we derive for optimizing the TVO has empirically lower variance than the REINFORCE [16] estimator, and unlike the reparameterization trick (which is only applicable to a limited family of continuous latent variables), applies to both continuous and discrete latent variables models.

The TVO is a lower bound to the log evidence which can be made arbitrarily tight. We empirically show that optimizing the TVO results in better inference networks than optimizing the importance weighted autoencoder (IWAE) objective [8] for which tightening of the bound is known to make inference network learning worse [12]. While this problem can be ameliorated by reducing the variance of the gradient estimator in the case of reparameterizable latent variables [17], resolving it in the case of non-reparameterizable latent variables currently involves alternating optimization of model and inference networks [18–20].
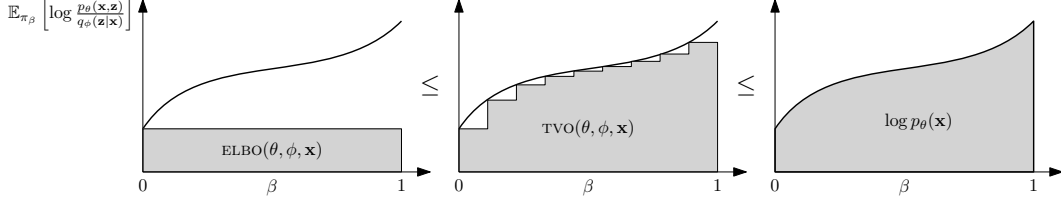
Figure 1: The thermodynamic variational objective (TVO) (center) is a finite sum numerical approximation to $\log p_\theta(\mathbf{x})$, defined by the thermodynamic variational identity (TVI) (right). The ELBO (left) is a single partition approximation of the same integral. $\pi_\beta$ is given in (7)

## 2 The Thermodynamic Variational Objective

The evidence lower bound (ELBO), used in learning variational autoencoders (VAEs), lower bounds the log evidence of a generative model $p_\theta(\mathbf{x}, \mathbf{z})$ parameterized by $\theta$ of a latent variable $\mathbf{z}$ and data $\mathbf{x}$. It can be written as the log evidence minus a Kullback-Leibler (KL) divergence

$$\text{ELBO}(\theta, \phi, \mathbf{x}) := \log p_\theta(\mathbf{x}) - \text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\right), \tag{1}$$

where $q_\phi(\mathbf{z}|\mathbf{x})$ is an inference network parameterized by $\phi$. As illustrated in Figure 1, the TVO

$$\underbrace{\frac{1}{K}\left[\text{ELBO}(\theta, \phi, \mathbf{x}) + \sum_{k=1}^{K-1} \mathbb{E}_{\pi_{\beta_k}}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}\,|\,\mathbf{x})}\right]\right]}_{\text{TVO}(\theta, \phi, \mathbf{x})} \leq \underbrace{\int_0^1 \mathbb{E}_{\pi_\beta}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}\,|\,\mathbf{x})}\right] d\beta}_{\text{THERMODYNAMIC VARIATIONAL IDENTITY}} = \log p_\theta(\mathbf{x}) \tag{2}$$

lower bounds the log evidence by using a Riemann sum approximation to the TVI, a one-dimensional integral over a scalar $\beta$ in a unit interval which evaluates to the log model evidence $\log p_\theta(\mathbf{x})$.

The integrand, which is a function of $\beta$, is an expectation of the so-called "instantaneous ELBO" [13] under $\pi_\beta(\mathbf{z})$, a geometric combination of $p_\theta(\mathbf{x}, \mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ which we formally define in §3. Remarkably, at $\beta = 0$, the integrand equals the ELBO. This therefore allows us to view the ELBO as a single-term left Riemann sum of the TVI. At $\beta = 1$, the integrand equals to the evidence upper bound (EUBO). This sheds a new unifying perspective on the VAE and wake-sleep objectives, which we explore in detail in §5 and Appendix G.

## 3 Connecting Thermodynamic Integration and Variational Inference

Suppose there are two unnormalized densities $\tilde{\pi}_i(\mathbf{z})$ ($i = 0, 1$) and corresponding normalizing constants $Z_i := \int \tilde{\pi}_i(\mathbf{z}) \, d\mathbf{z}$, which together define the normalized densities $\pi_i(\mathbf{z}) := \tilde{\pi}_i(\mathbf{z})/Z_i$. We can typically evaluate the unnormalized densities but cannot evaluate the normalizing constants.

While calculating the normalizing constants individually is usually intractable, thermodynamic integration [14, 15] allows us to compute the log of the ratio of the normalizing constants, $\log Z_1/Z_0$. To do so, we first form a family of unnormalized densities (or a "path") parameterized by $\beta \in [0, 1]$ between the two distributions of interest

$$\tilde{\pi}_\beta(\mathbf{z}) := \tilde{\pi}_1(\mathbf{z})^\beta \tilde{\pi}_0(\mathbf{z})^{1-\beta} \tag{3}$$

with the corresponding normalizing constants and normalized densities

$$Z_\beta := \int \tilde{\pi}_\beta(\mathbf{z}) \, d\mathbf{z}, \quad \text{and} \quad \pi_\beta(\mathbf{z}) := \tilde{\pi}_\beta(\mathbf{z})/Z_\beta. \tag{4}$$

Following Neal [15], we will find it useful to define a potential energy function $U_\beta(\mathbf{z}) := \log \tilde{\pi}_\beta(\mathbf{z})$ along with its first derivative $U'_\beta(\mathbf{z}) = \frac{dU_\beta(\mathbf{z})}{d\beta}$. We can then estimate the log of the ratio of the normalizing constants via the identity central to TI, derived in Appendix A,

$$\log Z_1 - \log Z_0 = \int_0^1 \mathbb{E}_{\pi_\beta}\left[U'_\beta(\mathbf{z})\right] d\beta. \tag{5}$$

2

Our key insight connecting TI and VI is the following. If we set

$$\tilde{\pi}_0(\mathbf{z}) := q_\phi(\mathbf{z} \,|\, \mathbf{x}) \quad Z_0 = \int q_\phi(\mathbf{z} \,|\, \mathbf{x}) \, \mathrm{d}\mathbf{z} = 1$$

$$\tilde{\pi}_1(\mathbf{z}) := p_\theta(\mathbf{x}, \mathbf{z}) \quad Z_1 = \int p_\theta(\mathbf{x}, \mathbf{z}) \, \mathrm{d}\mathbf{z} = p_\theta(\mathbf{x}) \tag{6}$$

this results in a geometric path between the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and the model $p_\theta(\mathbf{x}, \mathbf{z})$

$$\tilde{\pi}_\beta(\mathbf{z}) := p_\theta(\mathbf{x}, \mathbf{z})^\beta q_\phi(\mathbf{z}|\mathbf{x})^{1-\beta} \quad \text{and} \quad \pi_\beta(\mathbf{z}) := \frac{\tilde{\pi}_\beta(\mathbf{z})}{Z_\beta}, \tag{7}$$

where the first derivative of the potential is equal to the "instantaneous ELBO" [13]

$$U'_\beta(\mathbf{z}) = \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}. \tag{8}$$

Substituting (8) and $Z_0 = 1$ and $Z_1 = p_\theta(\mathbf{x})$ into (5) results in the *thermodynamic variational identity*

$$\log p_\theta(\mathbf{x}) = \int_0^1 \mathbb{E}_{\pi_\beta} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \,|\, \mathbf{x})} \right] d\beta. \tag{9}$$

This means that $\log p_\theta(\mathbf{x})$ can be expressed as a one-dimensional integral of an expectation of the instantaneous ELBO under $\pi_\beta$ from $\beta = 0$ to $\beta = 1$ (see Figure 1 (right)).

To obtain the thermodynamic variational objective (TVO) defined in (2), we lower bound the integral in (9) using a left Riemann sum. That this is in fact a lower bound follows from observation that the integrand is monotonically increasing, as shown in Appendix B. This is a result of our choice of path in (7), which allows us to show the derivative of the integrand is equal to the variance of $U'_\beta(\mathbf{z})$ under $\pi_\beta(\mathbf{z})$ and is therefore non-negative. For equal spacing of the partitions, where $\beta_k = k/K$, we arrive at the TVO in (2), illustrated in Figure 1 (middle). We present a generalized variant with non-equal spacing in Appendix C.

Maximizing the ELBO$(\theta, \phi, \mathbf{x})$ can be seen as a special case of the TVO, since for $\beta = 0$, $\pi_\beta(\mathbf{z}) = q_\phi(\mathbf{z}|\mathbf{x})$, and so the integrand in (9) becomes $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \,|\, \mathbf{x})} \right]$, which is equivalent to the definition of ELBO in (1). Because the integrand is increasing, we have

$$\text{ELBO}(\theta, \phi, \mathbf{x}) \leq \text{TVO}(\theta, \phi, \mathbf{x}) \leq \log p_\theta(\mathbf{x}), \tag{10}$$

which means that the TVO is an alternative to IWAE for tightening the variational bounds. In Appendix D we show maximizing the TVO is equivalent to minimizing a divergence between the variational distribution and the true posterior $p_\theta(\mathbf{z} \,|\, \mathbf{x})$.

The integrand in (9) is typically estimated by long running Markov chain Monte Carlo chains computed at different values of $\pi_\beta(\mathbf{z})$ [21, 22]. Instead, we propose a simple importance sampling mechanism that allows us to reuse samples across an arbitrary number of discretizations and which is compatible with gradient-based learning.

## 4   Optimizing the TVO

We now provide a novel score-function based gradient estimator for the TVO which does not require the reparameterization trick.

**Gradients**   To use the TVO as a variational objective we must be able to differentiate through terms of the form $\nabla_\lambda \mathbb{E}_{\pi_{\lambda,\beta}}[f_\lambda(\mathbf{z})]$, where both $\pi_{\lambda,\beta}(\mathbf{z})$ and $f_\lambda(\mathbf{z})$ are parameterized by $\lambda$, and $\pi_{\lambda,\beta}(\mathbf{z})$ contains an intractable normalizing constant. In the TVO, $f_\lambda(\mathbf{z})$ is the instantaneous ELBO and $\lambda := \{\theta, \phi\}$, but our method is applicable for generic $f_\lambda(\mathbf{z}) : \mathbb{R}^M \mapsto \mathbb{R}$.

We can compute such terms using the *covariance gradient estimator* (derived in Appendix E)

$$\nabla_\lambda \mathbb{E}_{\pi_{\lambda,\beta}}[f_\lambda(\mathbf{z})] = \mathbb{E}_{\pi_{\lambda,\beta}}[\nabla_\lambda f_\lambda(\mathbf{z})] + \text{Cov}_{\pi_{\lambda,\beta}} [\nabla_\lambda \log \tilde{\pi}_{\lambda,\beta}(\mathbf{z}), f_\lambda(\mathbf{z})] \tag{11}$$

3

We emphasize that, like REINFORCE, our estimator relies on the log-derivative trick, but crucially *unlike* REINFORCE, doesn't require differentiating through the normalizing constant $Z_\beta = \int \tilde{\pi}_{\lambda,\beta}(\mathbf{z}) \, d\mathbf{z}$. We clarify the relationship between our estimator and REINFORCE in Appendix F.

The covariance in (11) has the same dimensionality as $\lambda \in \mathbb{R}^D$ because it is between $\nabla_\lambda \log \tilde{\pi}_{\lambda,\beta}(\mathbf{z}) \in \mathbb{R}^D$ and $f_\lambda(\mathbf{z}) \in \mathbb{R}$ and is defined as

$$\mathrm{Cov}_{\pi_{\lambda,\beta}}(\mathbf{a}, b) := \mathbb{E}_{\pi_{\lambda,\beta}} \left[ (\mathbf{a} - \mathbb{E}_{\pi_{\lambda,\beta}}[\mathbf{a}])(b - \mathbb{E}_{\pi_{\lambda,\beta}}[b]) \right]. \tag{12}$$

To estimate this, we first estimate the inner expectations which are then used in estimating the outer expectation. Thus, estimating the gradient in (11) requires estimating expectations under $\pi_\beta$.

**Expectations** By using $q_\phi(\mathbf{z}|\mathbf{x})$ as the proposal distribution in $S$-sample importance sampling, we can estimate an expectation of a general function $f(\mathbf{z})$ under any $\pi_\beta(\mathbf{z})$ by simply raising each unnormalized importance weight to the power $\beta$ and normalizing:

$$\mathbb{E}_{\pi_\beta}[f(\mathbf{z})] \approx \sum_{s=1}^{S} \overline{w_s^\beta} f(\mathbf{z}_s), \tag{13}$$

where $\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})$, $\overline{w_s^\beta} := w_s^\beta / \sum_{s'=1}^{S} w_{s'}^\beta$ and $w_s := \frac{p_\theta(\mathbf{x},\mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})}$. This follows because each unnormalized importance weight can be expressed as

$$\frac{\tilde{\pi}_\beta(\mathbf{x},\mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})} = \frac{p_\theta(\mathbf{x},\mathbf{z}_s)^\beta q_\phi(\mathbf{z}_s|\mathbf{x})^{1-\beta}}{q_\phi(\mathbf{z}_s|\mathbf{x})} = \frac{p_\theta(\mathbf{x},\mathbf{z}_s)^\beta}{q_\phi(\mathbf{z}_s|\mathbf{x})^\beta} = \left( \frac{p_\theta(\mathbf{x},\mathbf{z}_s)}{q_\phi(\mathbf{z}_s|\mathbf{x})} \right)^\beta = w_s^\beta. \tag{14}$$

Instead of sampling $SK$ times, we can reuse $S$ samples $\mathbf{z}_s \sim q_\phi(\mathbf{z}|\mathbf{x})$ across an arbitrary number of terms, since evaluating the normalized weight $\overline{w_s^{\beta_k}}$ only requires raising each weight to different powers of $\beta_k$ before normalizing. Reusing samples in this way is a use of the method known as "common random numbers" and we include experimental results showing it reduces the variance of the covariance estimator in Appendix F [23].

The covariance estimator does not require $\mathbf{z}$ to be reparameterizable, which means it can be used in the cases of both non-reparameterizable continuous latent variables and discrete latent variables (without modifying the model using continuous relaxations [24, 25]).

## 5 Generalizing Variational Objectives

As previously observed, the left single Riemann approximation of the TVI equals the ELBO, while the right endpoint ($\beta = 1$) is equal to the EUBO. The EUBO is analogous to the ELBO but under the true posterior and is defined

$$\mathrm{EUBO}(\theta, \phi, \mathbf{x}) := \mathbb{E}_{p_\theta(\mathbf{z}\,|\,\mathbf{x})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}\,|\,\mathbf{x})} \right]. \tag{15}$$

We also have the following identity

$$\mathrm{EUBO}(\mathbf{x}, \theta, \phi) = \log p_\theta(\mathbf{x}) + \mathrm{KL}\left( p_\theta(\mathbf{z}|\mathbf{x}) || q_\phi(\mathbf{z}|\mathbf{x}) \right) \tag{16}$$

which should be contrasted against (1). We define an upper-bound variant of the TVO using the right (rather than left) Riemann sum. Setting $\beta_k = k/K$

$$\mathrm{TVO}_K^U(\theta, \phi, \mathbf{x}) := \frac{1}{K} \left[ \mathrm{EUBO}(\theta, \phi, \mathbf{x}) + \sum_{k=1}^{K-1} \mathbb{E}_{\pi_{\beta_k}} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}\,|\,\mathbf{x})} \right] \right] \geq \log p(\mathbf{x}). \tag{17}$$

The wake-sleep (WS) [18] and reweighted wake-sleep (RWS) [19] algorithms have traditionally been viewed as using different objectives during the wake and sleep phase. The endpoints of the TVI, which the TVO approximates, correspond to the two objectives used in wake-sleep. We can therefore view WS as alternating between between $\mathrm{TVO}_1^L$ and $\mathrm{TVO}_1^U$, i.e. a left and right single term Riemann approximation to the TVI. We show this algebraically in Appendix G and additionally, show how the objectives used in variational inference [26], variational autoencoders [1, 2], and inference compilation [27] are all special cases of $\mathrm{TVO}_1^L$ and $\mathrm{TVO}_1^U$. We refer the reader to [20] for a further discussion of the wake-sleep algorithm.
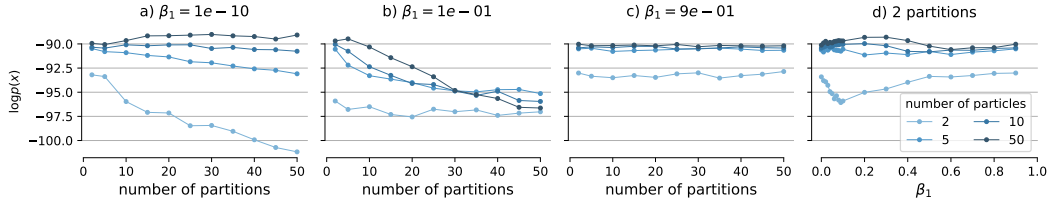
Figure 2: Investigation of how number of particles $S$, number of partitions $K$, and $\beta_1$ affect learning of the generative model. In the first three plots (a-c), we vary $S$ and $K$ for different values of $\beta_1$ and observe that while $S$ should be as high as possible, there is an optimal value for $K$, beyond which performance begins to degrade. Assuming $\beta_1$ is well-chosen, we see that as few as $K = 2$ partitions can result in good model learning, as seen in the last plot (d).

# 6    Related Work

Thermodynamic integration was originally developed in physics to calculate the difference in free energy of two molecular systems [28]. Neal [15] and Gelman and Meng [14] then introduced TI into the statistics community to calculate the ratios of normalizing constants of general probability models. TI is now commonly used in phylogenetics to calculate the Bayes factor $B = p(x|M_1)/p(x|M_0)$, where $M_0, M_1$ are two models specifying (for instance) tree topologies and branch lengths [22, 29, 30]. We took inspiration from Fan et al. [31] who replaced the "power posterior" $p(\theta|\mathbf{x}, M, \beta) = p(x|\theta, M)^{\beta} p(\theta, M)/Z_{\beta}$ of Xie et al. [29] with $p(\theta|\mathbf{x}, M, \beta) = [p(\mathbf{x}|\theta, M)p(\theta|M)]^{\beta}[p_0(\theta|M)]^{1-\beta}/Z_{\beta}$, where $p_0(\theta|M)$ is a tractable reference distribution chosen to facilitate sampling. That the integrand in (9) is strictly increasing was observed by Lartillot and Philippe [22].

We refer the reader to Titsias and Ruiz [32] for a summary of the numerous advances in variational methods over recent years. The method most similar to our own was proposed by Bornschein et al. [33], who introduced another way of improving deep generative modeling through geometrically interpolating between distributions and using importance sampling to estimate gradients. Unlike the TVO, they define a lower bound on the marginal likelihood of a modified model defined as $(p_{\theta}(\mathbf{x}, \mathbf{z})q_{\phi}(\mathbf{z}|\mathbf{x})q(\mathbf{x}))^{1/2}/Z$ where $q(\mathbf{x})$ is an auxiliary distribution.

Grosse et al. [34] studied annealed importance sampling (AIS), a related technique that estimates partition functions using a sequence of intermediate distributions to form a product of ratios of importance weights. They observe the geometric path taken in AIS is equivalent to minimizing a weighted sum of KL divergences, and use this insight to motivate an alternative path. To the best of our knowledge, our work is the first to explicitly connect TI and VI.

# 7    Experiments

## 7.1    Discrete Deep Generative Models

We use the TVO to learn the parameters of a deep generative model with discrete latent variables.[1] We use the binarized MNIST dataset with the standard train/validation/test split of 50k/10k/10k [35]. We train a sigmoid belief network, described in detail in Appendix I, using the TVO with the Adam optimizer. In the first set of experiments we investigate the effect of the discretization $\beta_{0:K}$, number of partitions $K$ and number of particles $S$. We then compare against variational inference for Monte Carlo objectives (VIMCO) and RWS (with the wake-$\phi$ objective) state-of-the-art IWAE-based methods for learning discrete latent variable models [20]. All figures have been smoothed for clarity.

**The effect of $S$, $K$, and $\beta$ locations** We expect that increasing the number of partitions $K$ makes the Riemann sum approximate the integral over $\beta$ more tightly. However, with each addition term, we add noise due to the use of importance sampling to estimate the expectation $\mathbb{E}_{\pi_{\beta}}[\log p/q]$. Importance sampling estimates of points on the curve further to the right are likely to be more biased because $\pi_{\beta}$ gets further from $q$ as we increase $\beta$. We found the combination of these two effects means that there

---

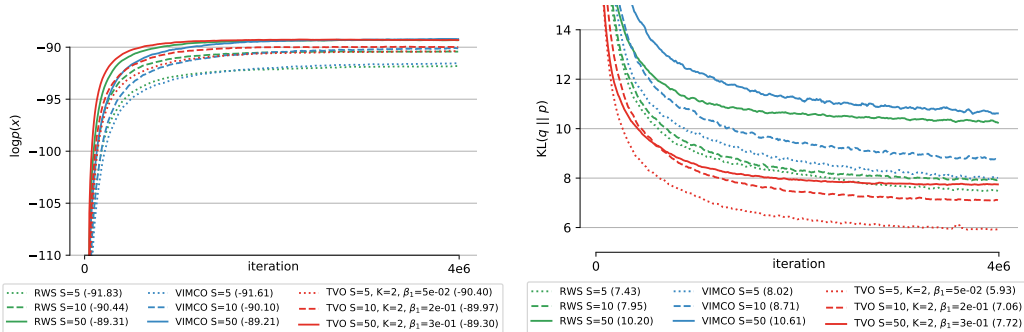[1]Code to reproduce all experiments is available at: `https://github.com/vmasrani/tvo`.

5

Figure 3: Comparisons with baselines on a held out test set. *(Left)* Learning curves for different methods. For TVO outperforms other methods both in terms of speed of convergence and the learned model for $S < 50$. At $S = 50$ VIMCO achieves a higher test log evidence but takes longer to converge than the TVO. *(Right)* KL divergence between current $q$ and $p$ (which measures how well $q$ "tracks" $p$) is lowest for TVO.

is a "sweet spot," or an optimal number of partitions beyond which adding more partitions becomes detrimental to performance.

We have empirically observed that the curve in Figure 1 is often rising sharply from $\beta = 0$ until a point of maximum curvature $\beta^*$, after which it is almost flat until $\beta = 1$, as seen in Figure 4. We hypothesized that if $\beta_1$ is located far before $\beta^*$ (the point of maximum curvature), a large number of additional partitions would be needed to capture additional area, while if $\beta_1$ is located after $\beta^*$, additional partitions would simply incur a high cost of bias without significantly tightening the bound. To investigate this, we choose small ($10^{-10}$), medium (0.1) and large (0.9) values of $\beta_1$, and logarithmically space the remaining $\beta_{2:K}$ between $\beta_1$ and 1. For each value of $\beta_1$ we train the discrete generative model for $K \in \{2, 5, 10, \ldots, 50\}$ and $S \in \{2, 5, 10, 50\}$, and show the test log evidence at the last iteration of each trial, approximated by evaluating the IWAE loss with 5000 samples.

Our hypothesis is corroborated in Figure 2, where we observe in Figure 2a that for $\beta_1 = 10^{-10}$ a large number of partitions are needed to approximate the integral. In Figure 2b we increase $\beta_1$ to $10^{-1}$ and observe only a few partitions are needed to improve performance, after which adding additional partitions becomes detrimental to model learning.



From Figure 2c we can see that if $\beta_1$ is chosen to be well beyond $\beta^*$, the Riemann sum cannot recover the "lost" area even if the number of partitions is increased. That the performance does not degrade in this case is due to the fact that for sufficiently high $\beta_k$, the curve in Figure 1 is flat and therefore $\pi_{\beta_k} \approx \pi_{\beta_k+1} \approx p_\theta(\mathbf{z} \,|\, \mathbf{x})$. We also observe that increasing number of samples $S$—which decreases importance sampling bias per partition—improves performance in all cases.

Figure 4: The location of $\beta^*$, the point of maximum curvature.

In our second experiment, shown in the Figure 2d, we fix $K = 2$ and investigate the quality of the learned generative model for different $\beta_1$, This plot clearly shows $\beta^*$ is somewhere near 0.3, as model learning improves as $\beta_1$ approaches this point then begins to degrade.

Given these results, we recommend using as many particles $S$ as possible and performing a hyper-parameter search over $\beta_1$ (with $K = 2$) when using the TVO objective. We leave finding the optimal placement of discretization points to future work.

**Performance** In Figure 3 (left), we compare the TVO against VIMCO and RWS with the wake-$\phi$ objective, the state-of-the-art IWAE-based methods for learning discrete latent variable models [20]. For $S < 50$, the TVO outperforms both methods in terms of speed of convergence and the final test log evidence $\log p_\theta(\mathbf{x})$, estimated using 5000 IWAE particles as before. At $S = 50$ VIMCO achieves a higher test log evidence but converges more slowly.
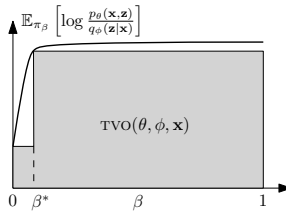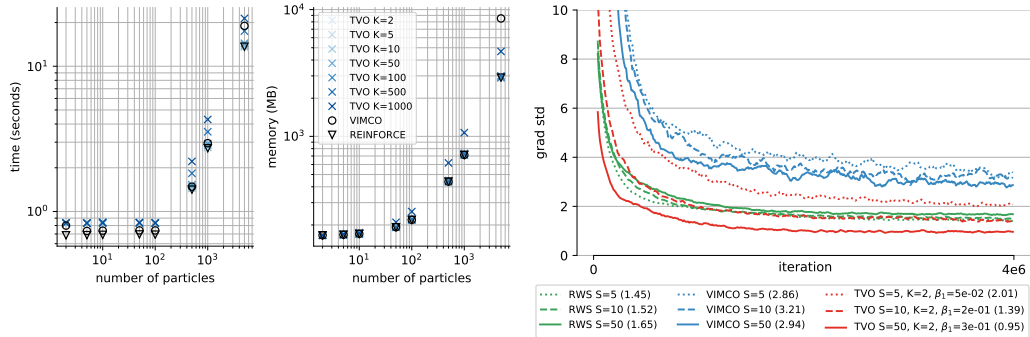
6

Figure 5: Computational and gradient estimator efficiency. *(Left)* Time and memory efficiency of the TVO with increasing number of partitions vs baselines, measured for $100$ iterations of optimization. Increasing the number of partitions is much cheaper than increasing the number of particles. *(Right)* Standard deviation of the gradient estimator for each objective. TVO is lowest variance, VIMCO is highest variance, RWS is in the middle.

We also investigate the quality of the learned inference network by plotting the KL divergence (averaged over the test set) between the current $q$ and current $p$ as training progresses (Figure 3 (right)). This indicates how well $q$ "tracks" $p$. This is estimated as log evidence minus ELBO where the former is estimated as before and the latter is estimated using 5000 Monte Carlo samples. The KL is lowest for TVO.

Somewhat surprisingly, for all methods, increasing number of particles makes the KL worse. We speculate that this is due to the "tighter bounds" effect of Rainforth et al. [12], who showed that increasing the number of samples can positively affect model learning but adversely affect inference network learning, thereby increasing the KL between the two.

**Efficiency** Since we use $K = 2$ partitions for the same number of particles $S$, the time and memory complexity of TVO is double that of other methods. While this is true, in both time and memory cases, the constant factor for increasing $S$ is much higher than for increasing $K$. As shown in Figure 5 (left), it is virtually free to increase number of partitions. This is because for each new particle, we must additionally sample from the inference network and score the sample under both $p$ and $q$ to obtain a weight. On the other hand, we can reuse the $S$ samples and corresponding weights in estimating values for the $K + 1$ terms in the Riemann sum. Thus, the region of the the computation graph that is dependent on $K$ is *after* the expensive sampling and scoring, and only involves performing basic operations on additional matrices of size $S \times K$.

**Variance** In Figure 5 (right), we plot the standard deviation of the gradient estimator for each method, where we compute the standard deviation for the $d^{\text{th}}$ element of the gradient estimated over 10 samples and take the average across all $D$.

The gradient estimator of the TVO has lower variance than both VIMCO, which uses REINFORCE with a control variate as a gradient estimator and RWS which can calculate the gradient without reparameterizing or using the log-derivative trick. At $S = 5$, RWS has lower gradient variance but its performance is worse in terms of both model and inference learning.

## 7.2 Continuous Deep Generative Models

Using the binarized MNIST dataset and experimental design described above, we also evaluated our method on a deep generative model with continuous latent variables. The model is described in detail in Appendix I. For each $S \in \{5, 10, 50\}$ we sweep over $K \in \{2, ..., 6\}$ and 20 $\beta_1$ values linearly spaced between $10^{-2}$ and $0.9$. We optimize the objectives using the Adam optimizer with default parameters.

**Performance** In Figure 6 (left), we train the model using the TVO and compare against the same model trained using the single sample VAE objective and multisample IWAE objective. The TVO outperforms the VAE and performs competitively with IWAE at 50 samples, despite not using the reparameterization trick. IWAE is the top performing objective in all cases. As in the discrete case,
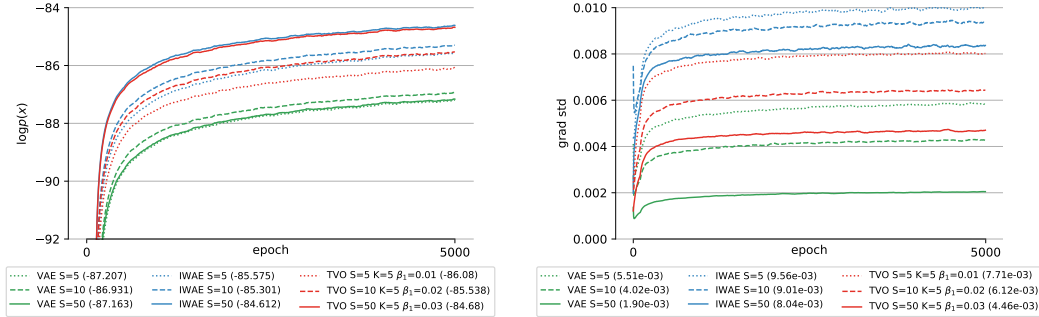
Figure 6: Learning curves for learning continuous deep generative models using different objectives. *(Left)* Despite not using the reparameterization trick, TVO outperforms VAEs and is competitive with IWAE at 50 samples. For all S, IWAE > TVO > VAE. *(Right)* Standard deviation of the gradient estimator for each objective. The TVO has lower variance than IWAE but higher than VAE.

increasing the number of particles $S$ improves model learning for all methods, but the improvement is most significant for the TVO. Interestingly VAE performance actually *decreases* when the number of samples increases from 10 to 50. A similar effect was noticed by Burda et al. [8] on the omniglot dataset.

**Variance** In Figure 6 (right), we plot the standard deviation of each method's gradient estimator. The standard deviation of the TVO estimator falls squarely between VAE (best) and IWAE (worst). The variance of each method improves as the number of samples increases, and as in the discrete model, the improvement is most significant in the case of TVO. Unlike in the discrete case, the variance does not decrease as the optimization proceeds, but plateaus early and then gradually increases. In Appendix F we include additional experiments to evaluate the properties of the covariance gradient estimator when used on the ELBO.

For both IWAE and the TVO, increasing the number of samples leads to decreased gradient variance and improved model learning. However, IWAE has the best performance but the highest variance across the three models. These results lend support to the conclusions of Rainforth et al. [12] who observe that the variance of a gradient estimator "is not always a good barometer for the effectiveness of a gradient estimation scheme."

# 8    Conclusions

The thermodynamic variational objective represents a new way to tighten evidence bounds and is based on a tight connection between variational inference and thermodynamic integration. We demonstrated that optimizing the TVO can have a positive impact on the learning of discrete deep generative models and can perform as well as using the reparameterization trick to learn continuous deep generative models.

The weakness of our method lies in choosing the discretization points. This does, however, point out opportunities for future work wherein we adaptively select optimal positions of the $\beta_{1:K}$ points, perhaps using techniques from the Bayesian numerical quadrature literature [36–38].

The approximate path integration perspective provided by our development of the TVO also sheds light on the connection between otherwise disparate deep generative model learning techniques. In particular, the TVO integration perspective points to ways to improve wake-sleep via tightening the EUBO using similar integral upper-bounding techniques. Further experimentation is warranted to explore how TVO insights can be applied to all special cases of the TVO including non-amortized variational inference and to the use of the TVO as a compliment to annealing importance sampling for final model evidence evaluation.

## References

[1] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

[2] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

[3] Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799, 2014.

[4] Andriy Mnih and Danilo Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196, 2016.

[5] George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pages 2624–2633, 2017.

[6] Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics*, pages 489–498, 2017.

[7] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. In *Advances in Neural Information Processing Systems*, pages 441–452, 2018.

[8] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

[9] Chris J Maddison, John Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Teh. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6576–6586, 2017.

[10] Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential Monte Carlo. In *International Conference on Learning Representations*, 2018.

[11] Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2018.

[12] Tom Rainforth, Adam R Kosiorek, Tuan Anh Le, Chris J Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *ICML*, 2018.

[13] David Blei. Variational inference: Foundations and innovations. URL http://www.cs.columbia.edu/~blei/talks/Blei_VI_tutorial.pdf.

[14] Andrew Gelman and Xiao-Li Meng. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pages 163–185, 1998.

[15] Radford M Neal. Probabilistic inference using Markov chain Monte Carlo methods. 1993.

[16] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[17] George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J Maddison. Doubly reparameterized gradient estimators for Monte Carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

[18] Geoffrey E Hinton, Peter Dayan, Brendan J Frey, and Radford M Neal. The "wake-sleep" algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161, 1995.

[19] Jörg Bornschein and Yoshua Bengio. Reweighted wake-sleep. In *International Conference on Learning Representations*, 2015.

[20] Tuan Anh Le, Adam R Kosiorek, N Siddharth, Yee Whye Teh, and Frank Wood. Revisiting reweighted wake-sleep. *arXiv preprint arXiv:1805.10469*, 2018.

[21] Nial Friel and Anthony N Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.

[22] Nicolas Lartillot and Hervé Philippe. Computing Bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.

[23] Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[24] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[25] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[26] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[27] Tuan Anh Le, Atilim Gunes Baydin, and Frank Wood. Inference compilation and universal probabilistic programming. *arXiv preprint arXiv:1610.09900*, 2016.

[28] DJ Evans. Molecular dynamics simulation of statistical mechanical systems. *International School of Physics, "Emico Fermi"(July 22-August 2, 1985), to be published*, 1986.

[29] Wangang Xie, Paul O Lewis, Yu Fan, Lynn Kuo, and Ming-Hui Chen. Improving marginal likelihood estimation for Bayesian phylogenetic model selection. *Systematic biology*, 60(2):150–160, 2010.

[30] Nicolas Rodrigue and Stéphane Aris-Brosou. Fast Bayesian choice of phylogenetic models: Prospecting data augmentation–based thermodynamic integration. *Systematic biology*, 60(6):881–887, 2011.

[31] Yu Fan, Rui Wu, Ming-Hui Chen, Lynn Kuo, and Paul O Lewis. Choosing among partition models in Bayesian phylogenetics. *Molecular biology and evolution*, 28(1):523–532, 2010.

[32] Michalis K. Titsias and Francisco J. R. Ruiz. Unbiased implicit variational inference, 2018.

[33] Jörg Bornschein, Samira Shabanian, Asja Fischer, and Yoshua Bengio. Bidirectional Helmholtz machines. In *International Conference on Machine Learning*, pages 2511–2519, 2016.

[34] Roger B Grosse, Chris J Maddison, and Ruslan R Salakhutdinov. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems*, pages 2769–2777, 2013.

[35] Ruslan Salakhutdinov and Iain Murray. On the quantitative analysis of deep belief networks. In *Proceedings of the 25th international conference on Machine learning*, pages 872–879. ACM, 2008.

[36] Anthony O'Hagan. Bayes–Hermite quadrature. *Journal of statistical planning and inference*, 29(3): 245–260, 1991.

[37] Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian Monte Carlo. *Advances in neural information processing systems*, pages 505–512, 2003.

[38] Michael Osborne, Roman Garnett, Zoubin Ghahramani, David K Duvenaud, Stephen J Roberts, and Carl E Rasmussen. Active learning of model evidence using Bayesian quadrature. In *Advances in neural information processing systems*, pages 46–54, 2012.

[39] Shinto Eguchi et al. A differential geometric approach to statistical inference on the basis of contrast functionals. *Hiroshima mathematical journal*, 15(2):341–391, 1985.

[40] P L'Ecuyer. On the interchange of derivative and expectation. *Management Science, to appear*, 1993.

[41] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

[42] Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 538–545. Morgan Kaufmann Publishers Inc., 2001.

[43] Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

## A  Thermodynamic Integration

Thermodynamic integration is a technique used in physics and phylogenetics to approximate intractable normalized constants of high dimensional distributions [14, 15]. It is based on the observation that it is easier to calculate the ratio of two unknown normalizing constants than it is to calculate the constants themselves. More formally, consider two densities over space $\mathcal{Z}$

$$\pi_i(\mathbf{z}) = \frac{\tilde{\pi}_i(\mathbf{z})}{Z_i}, \qquad Z_i = \int_{\mathcal{Z}} \tilde{\pi}(\mathbf{z}) \, d\mathbf{z}, \qquad i \in \{0, 1\}. \tag{18}$$

To apply TI, we form a continuous family (or "path") between $\pi_0(\mathbf{z})$ and $\pi_1(\mathbf{z})$ via a scalar parameter $\beta \in [0, 1]$

$$\pi_\beta(\mathbf{z}) = \frac{\tilde{\pi}_\beta(\mathbf{z})}{Z_\beta} = \frac{\tilde{\pi}_1(\mathbf{z})^\beta \tilde{\pi}_0(\mathbf{z})^{1-\beta}}{Z_\beta}, \qquad Z_\beta = \int_{\mathcal{Z}} \tilde{\pi}_\beta(\mathbf{z}) \, d\mathbf{z}, \qquad \beta \in [0, 1]. \tag{19}$$

The central identity that allows us to compute the ratio $\log(Z_1/Z_0)$ is derived as follows. Assuming we can exchange integration with differentiation,

$$\frac{\partial \log Z_\beta}{\partial \beta} = \frac{1}{Z_\beta} \frac{\partial}{\partial \beta} Z_\beta$$

$$= \frac{1}{Z_\beta} \frac{\partial}{\partial \beta} \int \tilde{\pi}_\beta(\mathbf{z}) \, d\mathbf{z}$$

$$= \int \frac{1}{Z_\beta} \frac{\partial}{\partial \beta} \tilde{\pi}_\beta(\mathbf{z}) \, d\mathbf{z}$$

$$= \int \frac{\tilde{\pi}_\beta(\mathbf{z})}{Z_\beta} \frac{\partial}{\partial \beta} \log \tilde{\pi}_\beta(\mathbf{z}) \, d\mathbf{z},$$

which directly implies

$$\frac{\partial \log Z_\beta}{\partial \beta} = \mathbb{E}_{\pi_\beta} \left[ U'_\beta(\mathbf{z}) \right], \tag{20}$$

where the quantity $U_\beta(\mathbf{z}) = \log \tilde{\pi}_\beta(\mathbf{z})$ is referred to as the "potential" in statistical physics and $U'_\beta(\mathbf{z}) := \frac{\partial}{\partial \beta} U_\beta(\mathbf{z})$. The variable $\beta$ can be interpreted as the inverse temperature parameter. Because one can typically compute $\log \tilde{\pi}_\beta(\mathbf{z})$, (20) allows us to exchange the first derivative of something we cannot compute with an expectation over something we can compute. Then, to calculate the ratio $\log(Z_1/Z_0)$ we integrate out $\beta$ on both sides of (20)

$$\int_0^1 \frac{\partial \log Z_\beta}{\partial \beta} d\beta = \int_0^1 \mathbb{E}_{\pi_\beta} \left[ U'_\beta(\mathbf{z}) \right] d\beta \tag{21}$$

which via the fundamental theorem of calculus results in

$$\log(Z_1) - \log(Z_0) = \int_0^1 \mathbb{E}_{\pi_\beta} \left[ U'_\beta(\mathbf{z}) \right] d\beta. \tag{22}$$

## B  The Increasing Integrand

### B.1  Notation

$$\log p_\theta(\mathbf{x}) = \int_0^1 g(\beta) d\beta \tag{23}$$

$$g(\beta) = \mathbb{E}_{\pi_\beta(\mathbf{z})} \left[ U'(\mathbf{z}) \right] \tag{24}$$

$$U'(\mathbf{z}) = \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \,|\, \mathbf{x})} \tag{25}$$

Given our choice of geometric path $\pi_\beta(\mathbf{z}) = \tilde{\pi}_\beta(\mathbf{z})/Z_\beta$, $\tilde{\pi}_\beta(\mathbf{z}) = p(\mathbf{x}, \mathbf{z})^\beta q(\mathbf{z} \,|\, \mathbf{x})^{1-\beta}$, the potential $U'(\mathbf{z}) = \frac{\partial}{\partial \beta} \log \tilde{\pi}_\beta(\mathbf{z})$ loses its dependency on $\beta$ after differentiating. This allows us to show

$$\frac{\partial}{\partial \beta} g(\beta) = \operatorname{Var}_{\pi_\beta(\mathbf{z})}[U'(\mathbf{z})] \tag{26}$$

11

which means $\frac{\partial}{\partial\beta}g(\beta) \geq 0, \forall\beta \in [0,1]$ and therefore that $g(\beta)$ is monotonically non-decreasing. Changes between lines are tracked in blue.

*Proof of Equation* (26).

$$\frac{\partial}{\partial\beta}g(\beta) = \frac{\partial}{\partial\beta}\mathbb{E}_{\pi_\beta(\mathbf{z})}\left[U'(\mathbf{z})\right]$$

$$= \frac{\partial}{\partial\beta}\left[\int \pi_\beta(\mathbf{z})U'(\mathbf{z})\,\mathrm{d}\mathbf{z}\right]$$

$$= \int U'(\mathbf{z})\frac{\partial}{\partial\beta}\pi_\beta(\mathbf{z})\,\mathrm{d}\mathbf{z}$$

$$= \int U'(\mathbf{z})\frac{\partial}{\partial\beta}\left[Z_\beta^{-1}\tilde{\pi}_\beta(\mathbf{z})\right]\mathrm{d}\mathbf{z}$$

$$= \int U'(\mathbf{z})\left[\tilde{\pi}_\beta(\mathbf{z})\frac{\partial}{\partial\beta}Z_\beta^{-1} + Z_\beta^{-1}\frac{\partial}{\partial\beta}\tilde{\pi}_\beta(\mathbf{z})\right]\mathrm{d}\mathbf{z}\,.$$

Now we use the "inverse log-derivative" trick $\frac{\partial}{\partial x}(f(x)^{-1}) = -\frac{1}{f(x)}\frac{\partial}{\partial x}\log f(x)$ on the first term, and the log-derivative trick on the second

$$= \int U'(\mathbf{z})\left[\tilde{\pi}_\beta(\mathbf{z})\frac{-1}{Z_\beta}\frac{\partial}{\partial\beta}\log Z_\beta + \frac{1}{Z_\beta}\tilde{\pi}_\beta(\mathbf{z})\frac{\partial}{\partial\beta}\log\tilde{\pi}_\beta(\mathbf{z})\right]\mathrm{d}\mathbf{z} \tag{27}$$

$$= \int U'(\mathbf{z})\left[-\pi_\beta(\mathbf{z})\frac{\partial}{\partial\beta}\log Z_\beta + \pi_\beta(\mathbf{z})\frac{\partial}{\partial\beta}\log\tilde{\pi}_\beta(\mathbf{z})\right]\mathrm{d}\mathbf{z}, \tag{28}$$

Then we use (20) on the first term, and the definition of $U'(\mathbf{z})$ in the second

$$= \int U'(\mathbf{z})\left[-\pi_\beta(\mathbf{z})\mathbb{E}_{\pi_\beta(\mathbf{z})}\left[U'(\mathbf{z})\right] + \pi_\beta(\mathbf{z})U'(\mathbf{z})\right]\mathrm{d}\mathbf{z} \tag{29}$$

$$= -\int \pi_\beta(\mathbf{z})U'(\mathbf{z})\,\mathbb{E}_{\pi_\beta(\mathbf{z})}\left[U'(\mathbf{z})\right]\mathrm{d}\mathbf{z} + \int U'(\mathbf{z})U'(\mathbf{z})\pi_\beta(\mathbf{z})\mathrm{d}\mathbf{z} \tag{30}$$

Finally we rearrange, noting that the expectation is a scalar and can therefore come out of the integrand

$$= -\left[\mathbb{E}_{\pi_\beta(\mathbf{z})}\left[U'(\mathbf{z})\right]\right]\left[\int \pi_\beta(\mathbf{z})U'(\mathbf{z})\,\mathrm{d}\mathbf{z}\right] + \int U'(\mathbf{z})U'(\mathbf{z})\pi_\beta(\mathbf{z})\,\mathrm{d}\mathbf{z} \tag{31}$$

$$= -\left[\mathbb{E}_{\pi_\beta(\mathbf{z})}[U'(\mathbf{z})]\right]^2 + \mathbb{E}_{\pi_\beta(\mathbf{z})}\left[U'(\mathbf{z})^2\right] \tag{32}$$

$$= \mathrm{Var}_{\pi_\beta(\mathbf{z})}[U'(\mathbf{z})]. \tag{33}$$

Therefore,

$$\frac{\partial}{\partial\beta}g(\beta) = \mathrm{Var}_{\pi_\beta(\mathbf{z})}[U'(\mathbf{z})]. \tag{34}$$

$\square$

## C   The generalized TVO

The TVO presented in §2 is a lower bound to $\log p_\theta(\mathbf{x})$ using a left Riemann sum approximation to the thermodynamic variational identity. Using the right Riemann sum results in an upper bound which can be minimized (rather than maximized) during optimization (cf. §5). This loss is used in the inference compilation and during the sleep-phase $\phi$ update in the Wake-Sleep algorithm. Below we present both the upper-bound and lower-bound variants of the TVO, with non-equally spaced partitions $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$, $\Delta_{\beta_k} = \beta_k - \beta_{k-1}$, $K > 1$

$$\text{TVO}_K^L(\theta, \phi, \mathbf{x}) := \Delta_{\beta_1} \text{ELBO}(\theta, \phi, \mathbf{x}) + \sum_{k=2}^{K} \Delta_{\beta_k} \mathbb{E}_{\pi_{\beta_{k-1}}} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \leq \log p(\mathbf{x}) \quad (35)$$

$$\text{TVO}_K^U(\theta, \phi, \mathbf{x}) := \Delta_{\beta_K} \text{EUBO}(\theta, \phi, \mathbf{x}) + \sum_{k=1}^{K-1} \Delta_{\beta_k} \mathbb{E}_{\pi_{\beta_k}} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \geq \log p(\mathbf{x}), \quad (36)$$

where

$$\text{ELBO}(\theta, \phi, \mathbf{x}) := \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})} \left[ \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right], \qquad \text{EUBO}(\theta, \phi, \mathbf{x}) := \mathbb{E}_{p_\theta(\mathbf{z} \mid \mathbf{x})} \left[ \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right],$$

$$\pi_{\beta_k}(\mathbf{z}) := p_\theta(\mathbf{x}, \mathbf{z})^\beta q_\phi(\mathbf{z} \mid \mathbf{x})^{1-\beta} / Z_\beta, \qquad Z_\beta := \int p_\theta(\mathbf{x}, \mathbf{z})^\beta q_\phi(\mathbf{z} \mid \mathbf{x})^{1-\beta} \, d\mathbf{z} \,.$$

## D  Maximizing the TVO minimizes a divergence between the variational distribution and true posterior

We now show:

$$\text{TVO}(\theta, \phi, \mathbf{x}) = \log p_\theta(\mathbf{x}) - \mathcal{D}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p_\theta(\mathbf{z} \mid \mathbf{x})) \quad (37)$$

Where $\mathcal{D}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p_\theta(\mathbf{z} \mid \mathbf{x}))$ is a divergence between the variational distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ and true posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$. We refer to the notation defined in Appendix B.1 and the definition of divergence defined by Eguchi et al. [39].

*Proof.* The TVO is a left Riemann sum approximation of $\log p_\theta(\mathbf{x}) = \int_0^1 g(\beta) d\beta$, where $g(\beta) = \mathbb{E}_{\pi_\beta(\mathbf{z})} [U'(\mathbf{z})]$ and $g(\beta)$ is a differentiable monotonically non-decreasing function in $\beta$ (cf. Equation (26)). The TVO is therefore a lower bound of $\log p_\theta(\mathbf{x})$ and can be written

$$\text{TVO}(\theta, \phi, \mathbf{x}) \leq \log p_\theta(\mathbf{x})$$
$$\text{TVO}(\theta, \phi, \mathbf{x}) = \log p_\theta(\mathbf{x}) - c(\theta, \phi, \mathbf{x}), \quad c(\theta, \phi, \mathbf{x}) \geq 0 \quad (38)$$

We will show $c(\theta, \phi, \mathbf{x}) = \mathcal{D}(q_\phi(\mathbf{z} \mid \mathbf{x}) || p_\theta(\mathbf{z} \mid \mathbf{x}))$, which is equivalent to showing

① $c \geq 0, \ \forall p_\theta(\mathbf{z} \mid \mathbf{x}), q_\phi(\mathbf{z} \mid \mathbf{x}) \in \mathcal{Z}$

② $c = 0 \iff p_\theta(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x})$

① is implied in the definition of $c$ in 38. We now show ②.

**Forward direction** $\ \ (c = 0) \Rightarrow (p_\theta(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x}))$

If $c = 0$, the left Riemann sum must be an exact approximation to $\int_0^1 g(\beta) d\beta$. Because is $g(\beta)$ is differentiable (and assuming it is finite), the Riemann approximation can only be exact when $g(\beta)$ is flat (i.e. $\frac{\partial g(\beta)}{\partial \beta} = 0$) in the region $\beta \in [0, 1]$. We first recall that by definition, $\pi_0(\mathbf{z}) = q_\phi(\mathbf{z} \mid \mathbf{x})$ and $\pi_1(\mathbf{z}) = p_\theta(\mathbf{z} \mid \mathbf{x})$. Therefore

$$\int_0^1 \frac{\partial g(\beta)}{\partial \beta} d\beta = \int_0^1 0 \, d\beta \quad (39)$$
$$g(1) - g(0) = 0 \quad (40)$$
$$g(1) = g(0) \quad (41)$$
$$\mathbb{E}_{\pi_1(\mathbf{z})} [U'(\mathbf{z})] = \mathbb{E}_{\pi_0(\mathbf{z})} [U'(\mathbf{z})] \quad (42)$$
$$\mathbb{E}_{p_\theta(\mathbf{z} \mid \mathbf{x})} [U'(\mathbf{z})] = \mathbb{E}_{q_\phi(\mathbf{z} \mid \mathbf{x})} [U'(\mathbf{z})] \quad (43)$$

Which is only possible when $p_\theta(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x})$.

**Reverse direction**   $\left(p_\theta(\mathbf{z}\,|\,\mathbf{x}) = q_\phi(\mathbf{z}\,|\,\mathbf{x})\right) \Rightarrow \left(c = 0\right)$

If $p_\theta(\mathbf{z}\,|\,\mathbf{x}) = q_\phi(\mathbf{z}\,|\,\mathbf{x})$, the TVO can be written as

$$\text{TVO}(\theta, \phi, \mathbf{x}) = \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{\pi_{\beta_k}(\mathbf{z})}\left[\log\frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z}\,|\,\mathbf{x})}\right] \tag{44}$$

$$= \frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}_{\pi_{\beta_k}(\mathbf{z})}\left[\log p_\theta(\mathbf{x})\right] \tag{45}$$

$$= \log p_\theta(\mathbf{x}) \tag{46}$$

Therefore $c = 0$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\square$

# E   Derivation of the Covariance Gradient Estimator

We want to show that

$$\nabla_\lambda\,\mathbb{E}_{\pi_{\lambda,\beta}}[f(\mathbf{z}, \lambda)] = \mathbb{E}_{\pi_{\lambda,\beta}}[\nabla_\lambda f(\mathbf{z}, \lambda)] + \text{Cov}_{\pi_{\lambda,\beta}}\left[\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z}), f(\mathbf{z}, \lambda)\right]. \tag{47}$$

Our estimator holds under the common regularity conditions assumed for the score function estimator L'Ecuyer [40]. We begin with a simple lemma.

**Lemma 1.**

$$\nabla_\lambda\log Z_{\lambda,\beta}(\mathbf{x}) = \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})] \tag{48}$$

*Proof of lemma 1.*

$$\nabla_\lambda\log Z_{\lambda,\beta}(\mathbf{x}) = \frac{1}{Z_{\lambda,\beta}(\mathbf{x})}\nabla_\lambda Z_{\lambda,\beta}(\mathbf{x}) \tag{49}$$

$$= \frac{1}{Z_{\lambda,\beta}(\mathbf{x})}\nabla_\lambda\int\tilde{\pi}_{\lambda,\beta}(\mathbf{z})\,\mathrm{d}\mathbf{z} \tag{50}$$

$$= \frac{1}{Z_{\lambda,\beta}(\mathbf{x})}\int\nabla_\lambda\tilde{\pi}_{\lambda,\beta}(\mathbf{z})\,\mathrm{d}\mathbf{z} \tag{51}$$

$$= \int\frac{\tilde{\pi}_{\lambda,\beta}(\mathbf{z})}{Z_{\lambda,\beta}(\mathbf{x})}\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})\,\mathrm{d}\mathbf{z} \tag{52}$$

$$= \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})] \tag{53}$$

$$\square$$

To prove (47), we use the product rule and rearrange

$$\nabla_\lambda\,\mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z}, \lambda)] = \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda f(\mathbf{z}, \lambda) + f(\mathbf{z}, \lambda)\nabla_\lambda\log\pi_{\lambda,\beta}(\mathbf{z}\,|\,\mathbf{x})] \tag{54}$$

$$= \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda f(\mathbf{z}, \lambda) + f(\mathbf{z}, \lambda)\left(\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z}) - \nabla_\lambda\log Z_{\lambda,\beta}(\mathbf{x})\right)] \tag{55}$$

$$= \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda f(\mathbf{z}, \lambda)] + \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z}, \lambda)\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})]$$
$$- \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z}, \lambda)\nabla_\lambda\log Z_{\lambda,\beta}(\mathbf{x})]. \tag{56}$$

Now using lemma 1 on the third term

$$= \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda f(\mathbf{z}, \lambda)] + \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z}, \lambda)\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})]$$
$$- \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z}, \lambda)]\mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z})] \tag{57}$$

$$= \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda f(\mathbf{z}, \lambda)] + \text{Cov}_{\pi_{\lambda,\beta}(\mathbf{z}\,|\,\mathbf{x})}\left[\nabla_\lambda\log\tilde{\pi}_{\lambda,\beta}(\mathbf{z}), f(\mathbf{z}, \lambda)\right]. \tag{58}$$

Table 1: The effect of Common Random Numbers (CRN) on TVO variance. We use the discrete model of §7.1

| Iterations | 10 | 1e6 | 2e6 | 3e6 | 4e6 |
|---|---|---|---|---|---|
| Gradient std w/o CRN | 52.33 | 2.88 | 2.57 | 2.39 | 2.47 |
| Gradient std w/ CRN | 8.19 | 1.38 | 1.17 | 1.05 | 1.03 |

## F  Variance of the Covariance Gradient Estimator and its Relationship to REINFORCE

In this section we clarify the difference between the covariance estimator (11) and the REINFORCE estimator and empirically investigate its variance.

While both estimators use the log-derivative trick, the main difference between the two is the REINFORCE estimator requires differentiating through $\log \pi_\beta(\mathbf{z})$ which contains the intractable normalizing constant, while the covariance estimator only requires differentiating through the unnormalized distribution $\log \tilde{\pi}_\beta(\mathbf{z})$.

We can state the difference as follows. Assuming $\pi_\beta(\mathbf{z}) = \tilde{\pi}_\beta(\mathbf{z})/Z_\beta$ depends on parameters $\lambda$, to compute $\nabla_\lambda \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z})]$, one can use the following gradient estimators:

$$\text{REINFORCE:} \quad \mathbb{E}_{\pi_\beta(\mathbf{z})}\left[f(\mathbf{z})\nabla_\lambda \log \pi_\beta(\mathbf{z})\right]$$

$$\text{REINFORCE + BASELINE:} \quad \mathbb{E}_{\pi_\beta(\mathbf{z})}\left[\left(f(\mathbf{z}) - \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z})]\right)\nabla_\lambda \log \pi_\beta(\mathbf{z})\right]$$

$$\text{COV. ESTIMATOR (ours):} \quad \mathbb{E}_{\pi_\beta(\mathbf{z})}\left[\left(f(\mathbf{z}) - \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z})]\right)\left(\nabla_\lambda \log \tilde{\pi}_\beta(\mathbf{z}) - \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda \log \tilde{\pi}_\beta(\mathbf{z})]\right)\right]$$

Unlike REINFORCE, where a baseline is typically added ad-hoc to reduce variance, the baseline $b = \mathbb{E}_{\pi_\beta(\mathbf{z})}[f(\mathbf{z})]$ naturally appears as a result of differentiating through $\pi_\beta(\mathbf{z})$ using the identity $\nabla_\lambda \log Z_{\lambda,\beta}(\mathbf{x}) = \mathbb{E}_{\pi_\beta(\mathbf{z})}[\nabla_\lambda \log \tilde{\pi}_{\lambda,\beta}(\mathbf{z})]$ derived in appendix E. The baseline also partially explains the low variance of our estimator, as it is equivalent to the "average baseline" often used reinforcement learning [41, 42].

A second source of variance-reduction comes from reusing samples, a method known as "common random numbers" [23]. The terms in the TVO are highly correlated, thus we expect reusing a single batch of samples for each additional term will act to reduce variance according to equation 8.21 in Owen [23]. However, because the covariance term breaks into both positive and negative terms, common random numbers could potentially increase variance. In Table 1 we show the average gradient std at different iterations during the training procedure, using the $S = 50$ discrete model described in §7.1 and in Figure 3 and Figure 5. It is evident reusing samples significantly reduces the variance of the covariance gradient estimator, often by more than a factor of two.

In Figure 7 we compare the variance of our estimator to the reparameterization trick and REINFORCE on the continuous model described in §7.2. To control for any possible effect on variance the additional terms in the TVO could have, we use the ELBO (i.e the TVO with $K = 1$), and plot the gradient standard deviation for the COV estimator (ours), the reparameterization trick and REINFORCE. The COV estimator has higher variance than the reparameterization trick estimator, and outperforms the REINFORCE estimator which is numerically unstable. Both the standard deviation of both the COV estimator and the reparameterization trick improves as samples increase but the effect is more prominent for the COV estimator.

## G  Special Cases of the TVO

In Table 2, we summarize the different ways the TVO generalizes existing variational objectives, and in the following subsection we list the mathematical form of each objective. In the main text, we mentioned that the lower bound variant of the TVO with $K = 1$ partition can be seen as the ELBO. This connects the TVO to all methods that maximize the ELBO. The upper bound variant of the TVO with $K = 1$ partition can be seen as EUBO. This therefore connects the TVO to all methods
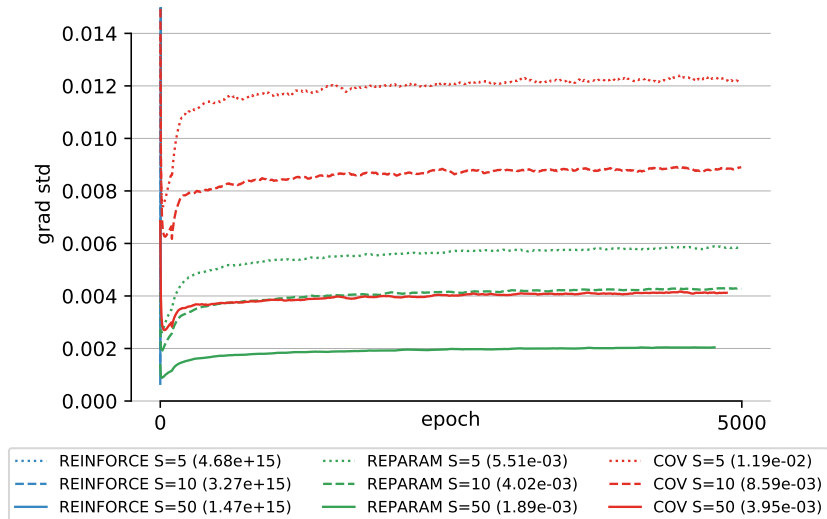
Figure 7: Comparing the standard deviation of gradient estimators on continuous VAEs trained on the ELBO. The covariance estimator has higher variance than the reparameterization trick for all $S$ but much lower than REINFORCE, which is numerically unstable.

that minimize the reverse KL divergence KL $(p_\theta(\mathbf{z}|\mathbf{x})||q_\phi(\mathbf{z}|\mathbf{x}))$, including WS, RWS and inference compilation.

For $K > 1$, we have a novel objective which we can optimize with respect to $\theta$, $\phi$ or both and therefore apply to all the variational methods summarized in Table 2.

Table 2: The thermodynamic variational identity generalizes existing variational objectives.

| Approximation | | Left Riemann sum (lower bound—maximize) | | Right Riemann sum (upper bound—minimize) | |
|---|---|---|---|---|---|
| Number of partitions | | 1 | $> 1$ | 1 | $> 1$ |
| Optimize | $\theta$ | wake in WS | $\mathrm{TVO}_K^L(\theta, \mathbf{x})$ | N/A | N/A |
| | $\phi$ | VI | $\mathrm{TVO}_K^L(\phi, \mathbf{x})$ | wake-$\phi$ in RWS, sleep in WS, inference compilation | $\mathrm{TVO}_K^U(\theta, \phi, \mathbf{x})$ |
| | $\theta, \phi$ | VAE | $\mathrm{TVO}_K^L(\theta, \phi, \mathbf{x})$ | N/A | N/A |

## G.1 Variational Objective Zoo

In the following we show how a number of commonly used variational objectives can be recovered from the TVO using a single partition $K = 1$. Each method can be extended by setting $K > 1$.

We have three degrees of freedom: 1) Whether we optimize $\theta$, $\phi$, or both 2) whether we maximize $\mathrm{TVO}_1^L(\theta, \phi, \mathbf{x})$ or minimize $\mathrm{TVO}_1^U(\theta, \phi, \mathbf{x})$, and 3) whether we use data sampled from the true data distribution $\{\mathbf{x}_i\}_{i=1}^N \overset{\text{i.i.d}}{\sim} p(\mathbf{x})$ or from our generative model $\{\mathbf{x}_i\}_{i=1}^N \overset{\text{i.i.d}}{\sim} p_\theta(\mathbf{x})$, as in the case of inference compilation and the sleep phase of the wake-sleep algorithm.

**Variational Inference**   Variational inference [26] can be recovered by learning $\phi$, maximizing $\text{TVO}_1^L(\phi, \mathbf{x})$, and using real data $\{\mathbf{x}_i\}_{i=1}^N \sim p(\mathbf{x})$:

$$\phi^* = \arg\max_\phi \mathbb{E}_{x \sim p(\mathbf{x})} \left[ \text{TVO}_1^L(\phi, \mathbf{x}) \right] \tag{59}$$

$$= \arg\max_\phi \mathbb{E}_{x \sim p(\mathbf{x})} \left[ \text{ELBO}(\phi, \mathbf{x}) \right] \tag{60}$$

**Inference Compilation**   If we instead sample data from our generative model $\{\mathbf{x}_i\}_{i=1}^N \sim p_\theta(x)$ and minimize $\text{TVO}_1^U(\phi, \mathbf{x})$ we recover the inference compilation objective [27]:

$$\phi^* = \arg\min_\phi \mathbb{E}_{x \sim p_\theta(\mathbf{x})} \left[ \text{TVO}_1^U(\phi, \mathbf{x}) \right] \tag{61}$$

$$= \arg\min_\phi \int p_\theta(\mathbf{x}) \left[ \mathbb{E}_{p_\theta(\mathbf{z} \mid \mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \right] \mathrm{d}\mathbf{x} \tag{62}$$

$$= \arg\min_\phi \int p_\theta(\mathbf{x}) \left[ \int \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x})} \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \right] \mathrm{d}\mathbf{z} \, \mathrm{d}\mathbf{x} \tag{63}$$

$$= \arg\min_\phi \int \int p_\theta(\mathbf{x}, \mathbf{z}) \left[ \log \frac{p(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \mathrm{d}\mathbf{z} \, \mathrm{d}\mathbf{x} \tag{64}$$

$$= \arg\min_\phi \mathbb{E}_{p(\mathbf{x}, \mathbf{z})} \left[ -\log q_\phi(\mathbf{z} \mid \mathbf{x}) \right] \tag{65}$$

**Variational Autoencoders**   The loss for VAEs[1, 2] follows the same setting as in the variational inference objective, above except now we learn both $\phi$ and $\theta$.

$$\phi^* = \arg\max_{\phi, \theta} \mathbb{E}_{x \sim p(\mathbf{x})} \left[ \text{TVO}_1^L(\theta, \phi, \mathbf{x}) \right] \tag{66}$$

$$= \arg\max_{\phi, \theta} \mathbb{E}_{x \sim p(\mathbf{x})} \left[ \text{ELBO}(\theta, \phi, \mathbf{x}) \right] \tag{67}$$

**Wake-sleep and reweighted wake sleep**   In the original wake-sleep algorithm [18], the authors proposed the *wake-phase $\theta$* update and *sleep-phase $\phi$* updates to train the generative model and inference network respectively. In Reweighted Wake-Sleep [19], two more objectives were proposed, the *reweighted wake-phase $\theta$* update[2] and the *wake-phase $\phi$* update. All except the *reweighted wake-phase $\theta$*[3] are special cases of the TVO and are listed below.

- **Wake-phase $\theta$ update** In the wake phase $\theta$ update, we consider $\phi$ fixed and maximize $\text{TVO}_1^L(\theta, \mathbf{x})$, using data $\{\mathbf{x}_i\}_{i=1}^N \sim p(\mathbf{x})$ sampled from the true distribution. This is similar to the variational inference update except we're learning $\theta$ instead of $\phi$:

$$\theta^* = \arg\max_\theta \mathbb{E}_{x \sim p(\mathbf{x})} [\text{TVO}_1^L(\theta, \mathbf{x})] \tag{68}$$

$$= \arg\max_\theta \mathbb{E}_{x \sim p(\mathbf{x})} \left[ \text{ELBO}(\theta, \mathbf{x}) \right] \tag{69}$$

- **Sleep-phase $\phi$ update** In the sleep phase $\phi$ update, we consider $\theta$ fixed and minimize $\text{TVO}_1^U(\phi, \mathbf{x})$ using simulated data $\{\mathbf{x}_i\}_{i=1}^N \sim p_\theta(\mathbf{x})$ and a single partition. This objective is the same as the inference compilation objective.

- **Wake-phase $\phi$ update** In the wake phase $\phi$ update, we instead use real data $\{\mathbf{x}_i\}_{i=1}^N \sim p(\mathbf{x})$ and again minimize $\text{TVO}_1^U$:

$$\phi^* = \arg\min_\phi \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \text{TVO}_1^U(\phi, \mathbf{x}) \right] \tag{70}$$

$$= \arg\min_\phi \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \mathbb{E}_{p_\theta(\mathbf{z} \mid \mathbf{x})} \left[ \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})} \right] \right] \tag{71}$$

$$= \arg\min_\phi \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \left[ \mathbb{E}_{p_\theta(\mathbf{z} \mid \mathbf{x})} \left[ -\log q_\phi(\mathbf{z} \mid \mathbf{x}) \right] \right] \tag{72}$$

---

[2]This was not the authors' original terminology and is used here to differentiate this objective from the original wake-phase $\theta$ update.

[3]This objective is not a special case of the TVO and is therefore not included in Table 2

This is the objective given in the wake-phase $\phi$ update in equation 6 of Le et al. [20]. The gradient estimator for performing this update given in Le et al. [20] is equivalent to the gradient estimator obtained via equations (11) and (13).

## H Additional Illustrations of the Thermodynamic Variational Identity

In Figure 8, we provide illustrations of how the $\mathbb{E}_{\pi_\beta}[U'(\mathbf{z})]$ curve relates to $\log p_\theta(\mathbf{x})$, KL $(q||p)$, KL $(p||q)$, ELBO and EUBO for the cases of ELBO $< 0 <$ EUBO and ELBO $<$ EUBO $< 0$. In the following, we provide derivations to justify the illustrations.
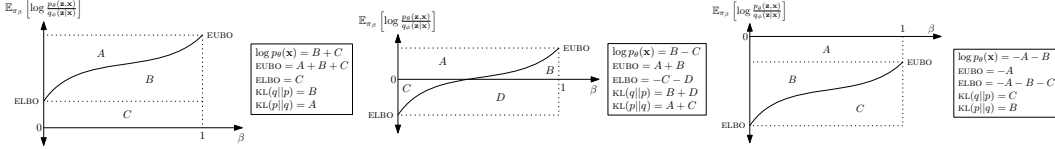


Figure 8: Different scenarios of the $\mathbb{E}_{\pi_\beta}[U'(\mathbf{z})]$ curve where ELBO $< 0$. On the left, $0 <$ ELBO $<$ EUBO. In the middle, ELBO $< 0 <$ EUBO. On the right ELBO $<$ EUBO $< 0$.

**Case ELBO $< 0 <$ EUBO** The top-most point of the curve is the EUBO by definition which means that the area $A + B$ is equal to the EUBO because of the unit length of the rectangle. In a similar manner, the ELBO is the negative of the area of $C + D$. Now, due to the thermodynamic identity, $\log p_\theta(\mathbf{x}) = \int_{\beta=0}^{1} \mathbb{E}_{\pi_\beta}[U'(\mathbf{z})]\,\mathrm{d}\beta$, it is equal to $B - C$ which is the area denoted by the definite integral.

To obtain the expressions for the KL, we use the identities

$$\log p_\theta(\mathbf{x}) = \text{ELBO}(\mathbf{x}, \theta, \phi) + \text{KL}\left(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\right) \tag{73}$$

$$= \text{EUBO}(\mathbf{x}, \theta, \phi) - \text{KL}\left(p_\theta(\mathbf{z}|\mathbf{x})||q_\phi(\mathbf{z}|\mathbf{x})\right) \tag{74}$$

**Case ELBO $<$ EUBO $< 0$** The top-most point of the curve is the EUBO by definition which means that $-A$ is equal to the EUBO because of the unit length of the rectangle. In a similar manner, the ELBO is $-A - B - C$. Due to the thermodynamic identity, $\log p_\theta(\mathbf{x}) = \int_{\beta=0}^{1} \mathbb{E}_{\pi_\beta}[U'(\mathbf{z})]\,\mathrm{d}\beta$, it is equal to $-A - B$ which is the area denoted by the definite integral. We obtain expressions for the KL similarly as before.

Similar line of reasoning gives rise to the relationships in Figure 8 (left).

## I Details for Deep Generative Models

**Discrete latent variables.** Sigmoid belief networks are used to evaluate objectives, continuous relaxations and control variate methods for learning discrete latent variable models [3–5, 19, 24, 25, 43]. The generative model is of the form $p(\mathbf{z}_{1:L}, \mathbf{x}) = p(\mathbf{z}_L) \prod_{\ell=1}^{L-1} p(\mathbf{z}_\ell|\mathbf{z}_{\ell+1})p(\mathbf{x}|\mathbf{z}_1)$ where each conditional on $\mathbf{z}_\ell$ is an independent Bernoulli whose parameters are a linear function of $\mathbf{z}_{\ell+1}$. The likelihood $p(\mathbf{x}|\mathbf{z}_1)$ is also an independent Bernoulli whose parameters are a linear function of $\mathbf{z}_1$ and we parameterize the prior $p(\mathbf{z}_L)$.

$$p_\theta(\mathbf{z}_L) = \text{Bernoulli}(\mathbf{z}_L|\mathbf{b}_L),$$
$$p_\theta(\mathbf{z}_\ell|\mathbf{z}_{\ell+1}) = \text{Bernoulli}(\mathbf{z}_\ell|\text{decoder}_\ell(2\,\mathbf{z}_{\ell+1} - 1)) \qquad \ell = L - 1, \ldots, 1,$$
$$p_\theta(\mathbf{x}|\mathbf{z}_1) = \text{Bernoulli}(\mathbf{x}|\text{decoder}_x(2\,\mathbf{z}_1 - 1) + \tilde{\mathbf{x}})$$

The inference network is factorized in the opposite way to the generative model, where $q(\mathbf{z}|\mathbf{x}) = q(\mathbf{z}_1|\mathbf{x}) \prod_{\ell=2}^{L} q(\mathbf{z}_\ell|\mathbf{z}_{\ell-1})$. Here, each conditional is an independent Bernoulli whose parameters are linear functions of the condition.

$$q_\phi(\mathbf{z}_1|\mathbf{x}) = \text{Bernoulli}\left(\mathbf{z}_1\,\middle|\,\text{encoder}_1\left(\frac{\mathbf{x} - \bar{\mathbf{x}} + 1}{2}\right)\right),$$
$$q_\phi(\mathbf{z}_\ell|\mathbf{z}_{\ell-1}) = \text{Bernoulli}(\mathbf{z}_\ell|\text{encoder}_\ell(2\,\mathbf{z}_{\ell-1} - 1)) \qquad \ell = 2, \ldots, L,$$

where $\mathbf{x} \in \{0, 1\}^{D_x}$ and $\mathbf{z}_\ell \in \{0, 1\}^{D_z}$. We set $L = 2$, $D_x = 784$ and $D_z = 200$. We used Pytorch's default parameter initialization. The Bernoulli distributions are independent Bernoulli distributions whose parameters are logits, i.e. they get passed through a sigmoid function to obtain the probability. $\bar{\mathbf{x}}$ is the mean over training data set and $\tilde{\mathbf{x}} = \log(\bar{\mathbf{x}} - 1)$. In the linear case, the encoders and decoders are linear functions of their inputs. In the non-linear case, they are a three-layer multilayer perceptrons with $\tanh$ nonlinearities of the form $\texttt{input\_dim} \xrightarrow{\text{Lin+tanh}} D_z \xrightarrow{\text{Lin+tanh}} D_z \xrightarrow{\text{Lin}} \texttt{output\_dim}$.

We used the Adam optimizer with the learning rate in $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$ and the other hyperparameters being set to the defaults. We picked the learning rate which performed best on the validation set which was $3 \times 10^{-4}$ for all algorithms. We ran the optimization for 4 million iterations with batch size 24.

**Continuous latent variables.** The model is of the form $p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z}) = \text{Normal}(\mathbf{z}|0, I)\text{Bernoulli}(\mathbf{x}|\text{decoder}_\theta(\mathbf{z}))$, where $\mathbf{z}$ is 200-dimensional and $\text{decoder}_\theta$ is a three-layer multilayer perceptron with $\tanh$ activations and sigmoid output which parameterizes the probabilities of the independent Bernoulli distribution.

$$p(\mathbf{z}) = \text{Normal}(\mathbf{z}|0, I),$$
$$p_\theta(\mathbf{x}|\mathbf{z}) = \text{Bernoulli}(\mathbf{x}|\text{decoder}_\theta(\mathbf{z}))$$

The inference network is of the form $q_\phi(\mathbf{z}|\mathbf{x}) = \text{Normal}(\mathbf{z}|\text{encoder}_\phi(\mathbf{x}))$, where the encoder is a two-layer multilayer perceptron with $\tanh$ activations. The output is passed through two separate linear layers which output the mean and the log standard deviations of the independent normal distribution.

$$q_\phi(\mathbf{z}|\mathbf{x}) = \text{Normal}(\mathbf{z}|\text{encoder}_\phi(\mathbf{x})),$$

where $\mathbf{x} \in \{0, 1\}^{D_x}$ and $\mathbf{z} \in \mathbb{R}^{D_z}$ for $D_x = 784$ and $D_z = 200$. The decoder is of the form $D_z \xrightarrow{\text{Lin+tanh}} D_z \xrightarrow{\text{Lin+tanh}} D_z \xrightarrow{\text{Lin}} D_x$ and its output is passed through a sigmoid to obtain probabilities for the Bernoulli distribution. The encoder is of the form $D_x \xrightarrow{\text{Lin+tanh}} D_z \xrightarrow{\text{Lin+tanh}} D_z$. Its output is passed through two *separate* neural networks of the form $D_z \xrightarrow{\text{Lin}} D_z$ which output the means and log standard deviations of the independent Normal distribution.

# J   Notation

<div align="center">

Table 3: Table of Notation

</div>

| | | |
|---:|:---:|:---|
| $\{\mathbf{x}_i\}_{i=1}^N$ | := | Data set consisting of N i.i.d samples $\mathbf{x}_i \in \mathbb{R}^D$ |
| $\{\mathbf{z}_i\}_{i=1}^N$ | := | Unobserved latent random variables $\mathbf{z}_i \in \mathbb{R}^M$ |
| $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z}) p_\theta(\mathbf{z})$ | := | The joint model parameterized by $\theta$, which factorizes into a likelihood $p_\theta(\mathbf{x} \mid \mathbf{z})$ and prior $p_\theta(\mathbf{z})$ |
| $p_\theta(\mathbf{z} \mid \mathbf{x}) = p_\theta(\mathbf{x}, \mathbf{z})/p_\theta(\mathbf{x})$ | := | The true (often intractable) posterior |
| $q_\phi(\mathbf{z} \mid \mathbf{x})$ | := | The variational distribution parameterized by $\phi$. By assumption $q_\phi(\mathbf{z} \mid \mathbf{x})$ is correctly normalized. |
| $\tilde{\pi}_{\lambda,\beta}(\mathbf{z}) = p_\theta(\mathbf{x}, \mathbf{z})^\beta q_\phi(\mathbf{z} \mid \mathbf{x})^{1-\beta}$ | := | The unnormalized path distributions. By construction, $\tilde{\pi}_{\lambda,\beta=1}(\mathbf{z}) = p_\theta(\mathbf{x}, \mathbf{z})$ and $\tilde{\pi}_{\lambda,\beta=0}(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x})$ |
| $\pi_{\lambda,\beta}(\mathbf{z} \mid \mathbf{x}) = \tilde{\pi}_{\lambda,\beta}(\mathbf{z})/Z_{\lambda,\beta}(\mathbf{x})$ | := | The path distributions parameterized by $\lambda = \{\theta, \phi\}$ and scalar parameter $\beta \in [0, 1]$. By construction, $\pi_{\lambda,\beta=1}(\mathbf{z} \mid \mathbf{x}) = p_\theta(\mathbf{z} \mid \mathbf{x})$ and $\pi_{\lambda,\beta=0}(\mathbf{z} \mid \mathbf{x}) = q_\phi(\mathbf{z} \mid \mathbf{x})$ |
| $Z_{\lambda,\beta}(\mathbf{x}) = \int \tilde{\pi}_{\lambda,\beta}(\mathbf{z}) \, \mathrm{d}\mathbf{z}_{1:N}$ | := | The normalizing constant for the path distributions. By construction $Z_{\lambda,\beta=1}(\mathbf{x}) = p_\theta(\mathbf{x})$ and $Z_{\lambda,\beta=0}(\mathbf{x}) = 1$ (because $q_\phi(\mathbf{z} \mid \mathbf{x})$ is assumed to be correctly normalized). |
| $U_{\lambda,\beta}(\mathbf{z}) = \log \tilde{\pi}_{\lambda,\beta}(\mathbf{z})$ | := | The potential energy. |
| $U'_{\lambda,\beta}(\mathbf{z}) = \frac{\partial}{\partial \beta} U_{\lambda,\beta}(\mathbf{z})$ | := | The first derivative of the potential w.r.t $\beta$, the inverse temperature. |

# K   Acronyms

**AIS** Annealed Importance Sampling

**ELBO** Evidence Lower Bound

**EUBO** Evidence Upper Bound

**IS** Importance Sampling

**IWAE** Importance Weighted Autoencoder

**KL** Kullback Leibler

**RWS** Reweighted Wake Sleep

**SGD** Stochastic Gradient Descent

**TI** Thermodynamic Integration

**TVI** Thermodynamic Variational Identity

**TVO** Thermodynamic Variational Objective

**VAE** Variational Autoencoder

**VI** Variational Inference

**VIMCO** Variational Inference For Monte Carlo Objectives

**WS** Wake Sleep