Accelerating NCE Convergence with Adaptive Normalizing Constant Computation

Anish Sevekari [*1] **Rishal Aggarwal** [*1 2] **David R. Koes** [1] **Maria Chikina** [1]

Abstract

Noise Contrastive Estimation (NCE) is a widely used method for training generative models, typically used as an alternative to Maximum Likelihood Estimation (MLE) when exact computations of probability are hard. NCE trains generative models by discriminating between data and appropriately chosen noise distributions. Although NCE is statistically consistent, it suffers from slow convergence and high variance when there is small overlap between the noise and data distributions. Both these problems are related to the flatness of the NCE loss landscape. We propose an innovative approach to circumvent slow convergence rates by quick inference of the optimal normalizing constant at every gradient step. This allows the rest of the parameters to have more freedom during NCE optimization. We analyze the use of both binary search and the Bennett Acceptance Ratio (BAR) for quick computation of the normalizing constant and show improved performance for both methods on convex and non-convex settings.

1. Introduction

Noise Contrastive Estimation (NCE) is a statistical method used to learn parameterized probability distributions that are specified up to a constant of proportionality. It was first proposed by Gutmann & Hyvärinen (2010; 2012) and has seen some recent attention for training Energy Based Models (EBMs), where probabilities are modelled as $p_{\tilde{\theta}}(x) \propto \exp(E_{\tilde{\theta}}(x))$ for some parametric family $E_{\tilde{\theta}}$. The main idea behind NCE loss is to train a classifier to discriminate between samples from a desired distribution P_* and an appropriately chosen noise distribution Q (Dyer, 2014; Gutmann & Hyvärinen, 2010; 2012; Rhodes et al., 2020). If the model is considered expressive enough, the optimal discriminator will learn an estimate of the ratios of densities $p_*(x)/q(x)$ from which the densities $p_*(x)$ can be successfully extracted (Sugiyama et al., 2012; Menon & Ong, 2016). The NCE training regime is especially advantageous as it avoids the computation of a partition function (as opposed to the Maximum Likelihood Estimation (MLE) framework) that is quite often intractable (Gutmann & Hirayama, 2012).

Although NCE provides computational advantages over MLE, it suffers from low rate of convergence and asymptotically high variance. One of the primary reasons behind both problems is the phenomenon known as *density chasm* (Rhodes et al., 2020). The NCE loss optimization landscape is flat near the optimum distribution and poses problems for first order (eg. gradient descent) and second order (eg. Newton's method) optimization methods, as observed in Liu et al. (2021). The flat region is especially prevalent when the data and noise distributions are well separated, that is, the KL-divergence between the two distributions is large.

In this work, we propose a method that improves the rate of convergence of NCE loss by enhancing the ability of NCE to self-normalize. Specifically, we change the update to the log of the partition function so that it yields better estimates of an appropriate constant value at each gradient descent step. We show that the correct constant value can be easily calculated using binary search or approximated through a method developed in statistical physics known as the Bennett Acceptance Ratio (BAR) (Bennett, 1976).

Particularly, our contributions are the following:

- We show that the NCE objective function is always convex along the log partition function coordinate (keeping the other parameters $\tilde{\theta}$ constant) and the optimal value of this coordinate can be calculated up to machine precision using binary search at every gradient descent step.
- We show that we get improvements in NCE optimization if we increase the learning rate of the log partition

[*]Equal contribution [1]Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA, USA [2]Joint PhD Program in Computational Biology, Carnegie Mellon University-University of Pittsburgh, Pittsburgh, Pennsylvania. Correspondence to: Anish Sevekari <ans849@pitt.edu>.

Accepted by the Structured Probabilistic Inference & Generative Modeling workshop of ICML 2024, Vienna, Austria.

function parameter.

- We get further improvements if we use the BAR formulation to update the log partition function parameter rather than the NCE gradient. We also show that binary search of the log partition function provides the best improvement in NCE optimization, albeit, at a slightly higher cost than applying just the BAR update.

- We validate our methods on both convex and non-convex settings, seeing consistent improvements over vanilla NCE optimization.

2. Related works

NCE has become a predominant area of research in both NLP (Mnih & Teh, 2012; Mnih & Kavukcuoglu, 2013; Dyer, 2014; Kong et al., 2020; Jozefowicz et al., 2016; Oord et al., 2018) and computer vision (Hjelm et al., 2018; Henaff, 2020; Tian et al., 2020; Feeney & Hughes, 2023). A major empirically observed issue with NCE is that a fixed noise distribution Q is not sufficient to learn good generative models. The two predominant approaches to solve this issue have been to either anneal between the noise and data distributions (Rhodes et al., 2020; Chehab et al., 2024; Gelman & Meng, 1998), or iteratively updating the noise distribution Q to yield a more informative loss (Xu, 2022; Gao et al., 2020; Goodfellow et al., 2014).

For a fixed Q, a recent work provided certain solutions to overcome the density chasm problem by using normalized gradient descent and an exponential loss function that is better behaved (Liu et al., 2021). It's important to note that while these improvements are substantial, the approach is orthogonal to our proposed method and theoretically, both the methods may be combined.

3. NCE objective function and optimization strategy

3.1. Vanilla Noise Contrastive Estimation

The NCE objective function is designed to learn parameterized energy based models of the form $p_{\tilde{\theta}}(x) \propto \exp(E_{\tilde{\theta}}(x))$. The NCE method defines an additional parameter F that represent the log of the partition function so that the learnt density then becomes $p_\theta = \exp(E_{\tilde{\theta}}(x) - F)$ where $\theta = [\tilde{\theta}, F]$. It is used when we have access to samples from a desired distribution P_* that we would like to learn.

From here on we use a short hand notation to represent densities, for eg. $p_\theta(x) = p_\theta$ The NCE objective for θ is then defined as the following:

Definition 3.1. The NCE loss of θ such that $\theta = [\tilde{\theta}, F]$ w.r.t to data distribution P_* and noise distribution Q is:

$$\mathcal{L}_{\texttt{NCE}}(\theta) = -\frac{1}{2}\mathbb{E}_{p_*}\left[\log \frac{p_\theta}{p_\theta + q}\right] - \frac{1}{2}\mathbb{E}_q\left[\log \frac{q}{p_\theta + q}\right] \quad (1)$$

Note that $\mathcal{L}_{\texttt{NCE}}(\theta)$ can be computed without the condition that p_θ is normalized. The crucial property of NCE loss is that it is consistent and has a unique global minima at $\theta = \theta_*$, with the corresponding constant F satisfying $F^* = \log \int_x \exp(E_{\tilde{\theta}}(x))dx$ (Gutmann & Hyvärinen, 2012)[1], provided that support of Q contains that of $P*$.

3.2. Exponential families

For some of our experiments, we work with parameter estimation setting for distributions in the exponential family, where the density of distributions P_θ is given by

$$p_\theta(x) = \exp(\langle \tilde{T}(x), \tilde{\theta} \rangle - F). \quad (2)$$

where $\tilde{T}(x)$ is sufficient statistics of x. Since we want to pay special attention to the normalizing constant, we will assume that the last coordinate of θ and $T(x)$ correspond to the normalizing constant, that is, $T(x) = [\tilde{T}(x), -1]$ and $\theta = [\tilde{\theta}, F]$. We will further assume that the data distribution corresponds to some distribution $P_* = P_{\theta_*}$ in the family, with a normalizing constant given by $\exp(F_*)$ for some F_*.

Further, in case of exponential families, the NCE loss function is well known to be convex (Uehara et al. (2020), Liu et al. (2021)) allowing for a nice analysis of convergence rates to the right parameter values. The proof for convexity is provided in Appendix A.1 for completeness.

3.3. Optimization of the Normalizing Constant

The NCE loss has two very distinct facets, one corresponding to finding a function proportional to the density of p_* and the other corresponding to finding the correct partition function. Since NCE is self-normalizing, it optimizes for both the parameters and the normalizing constant. The optimization dynamics are very different in both of these directions, suggesting that separating these two directions might be beneficial for improving optimization. In particular, we would like to take the advantage of following observation:

Lemma 3.2. *For any energy based parameterized family of distributions where $p_\theta(x)$ is given by*

$$p_\theta(x) = \exp(E_{\tilde{\theta}}(x) - F),$$

the function $\mathcal{L}_{\texttt{NCE}}(\theta)$ is convex as a function of F, for any fixed $\tilde{\theta}$.

[1]The result holds even in an non-parametric setting.

The partial derivative of $\mathcal{L}_{\texttt{NCE}}$ with respect to F is given by

$$\frac{\partial}{\partial F}\mathcal{L}_{\texttt{NCE}}(\theta) = \frac{1}{2}\mathbb{E}_{p_*}\left[\frac{q}{p_\theta + q}\right] - \frac{1}{2}\mathbb{E}_q\left[\frac{p_\theta}{p_\theta + q}\right] = \frac{1}{2}\int_x \frac{q(p_* - p_\theta)}{p_\theta + q}dx \tag{3}$$

Similarly, the second derivative is given by

$$\frac{\partial^2}{\partial F^2}\mathcal{L}_{\texttt{NCE}}(\theta) = \frac{1}{2}\int_x \frac{q(p_\theta)(p_* + q)}{(p_\theta + q)^2}dx \geq 0, \tag{4}$$

which shows convexity. If we look more closely at Equation 3 we see that it has limits of $-\frac{1}{2}$ when $F \to -\infty$ ($p_\theta \to \infty$) and $\frac{1}{2}$ when $F \to \infty$ ($p_\theta \to 0$). In particular, the NCE gradient is a monotonically increasing function which takes values from $\left[-\frac{1}{2}, \frac{1}{2}\right]$. Therefore, for fixed value of θ, there is a unique value of F where the loss $\mathcal{L}_{\texttt{NCE}}$ is minimized, and it can be computed to machine precision by using binary search or approximated using the Bennett Acceptance Ratio (introduced in Section 3.4). Furthermore, the optimization surface for exponential families remains convex when we get this optimal value of F (proof Appendix A.2). This raises the following question - does convergence performance of NCE increase when we treat the constant separately and optimize it using one of the aforementioned methods?

3.4. Bennett Acceptance Ratio

The Bennett Acceptance Ratio is a method developed in statistical physics for the estimation of the ratio of partition functions between two energy distributions. The BAR method has been mainly used on Boltzmann distributions with densities of the form $p(x) \propto \exp(-\beta E(x))$, where β is a constant dependant on temperature of the physical system being modelled. This method, however, is easily adaptable to EBMs by substituting $E(x) = -\beta E(x)$.

Now, we provide the application of this method on EBMs. Consider two densities $p(x)$ and $q(x)$ modelled as EBMs $p(x) \propto \exp(E_p(x))$ and $q(x) \propto \exp(E_q(x))$, where the partition functions are given by:

$$Z_{p\backslash q} = \int_x \exp(E_{p\backslash q}(x))dx \tag{5}$$

There exists appropriate weighting functions W such that:

$$W(\Delta E)\exp(E_p(x)) = W(-\Delta E)\exp(E_q(x)) \tag{6}$$

where: $\Delta E = E_p(x) - E_q(x)$

An example of such a weighting function can be the canonical Metropolis function used in Monte Carlo sampling, given by $W(\Delta E) = \min\{1, \exp(\Delta E)\}$.

Taking Equation 6, integrating over all configurations and multiplying and dividing by partition functions we get:

$$\frac{Z_p}{Z_q} = \frac{\mathbb{E}_q[W(-\Delta E)]}{\mathbb{E}_p[W(\Delta E)]} \tag{7}$$

Bennett (1976) found an optimal weight function that would minimize the variance of the estimate and showed that it is of the form

$$\frac{Z_p}{Z_q} = \frac{\mathbb{E}_q[\sigma(\Delta E - c)]}{\mathbb{E}_p[\sigma(-\Delta E + c)]} \tag{8}$$

where σ is the sigmoid function and $c = \log[Z_p/Z_q] = \log Z_p - \log Z_q$. Since c also contains the log of the ratio of the partition functions, the equation can be solved iteratively through fixed point iteration to make it self consistent. Representing our estimate of $\log Z_p$ as $\hat{F}_p$, the update to the estimate would be:

$$\hat{F}_p = \hat{F}_p - (\log \mathbb{E}_p[\sigma(-\Delta E + \hat{c})] - \log \mathbb{E}_q[\sigma(\Delta E - \hat{c})]) \tag{9}$$

with $\hat{c} = \hat{F}_p - \log Z_q$ or equivalently:

$$\Delta \hat{F}_p = -\log \mathbb{E}_p\left[\frac{q}{\hat{p} + q}\right] + \log \mathbb{E}_q\left[\frac{\hat{p}}{q + \hat{p}}\right] \tag{10}$$

Equation 10 could also be viewed as the update applied to the log partition function coordinate when optimizing it with gradient descent optimizers. We show in Section 4.1 that an update of this form can be applied to the constant to yield better convergence rates than the vanilla NCE objective function.

3.5. Training Noise Contrastive Estimation with the BAR and binary search

We provide here an easy way to apply the BAR update to the log partition function of our parameterized probability density function. We take the terms in Equation 3, and scale the magnitude of those terms with log to yield a *BAR-like* update that is of the form:

$$\nabla_F = \log \frac{1}{2}\mathbb{E}_{p_*}\left[\frac{q}{p_\theta + q}\right] - \log \frac{1}{2}\mathbb{E}_q\left[\frac{p_\theta}{p_\theta + q}\right] \tag{11}$$

Notice, this is not exactly a BAR update as the data samples are not from the density specified by p_θ but from the density we want to learn p_*. To apply such an update, we just need to take the log of the gradients provided by each summand of the NCE objective function. The gradients of the rest of the parameters $\tilde{\theta}$ are kept the same. In practice, we see best performance when we update the log of the partition

function using a naive Stochastic Gradient Descent (SGD) optimizer as it does not modify the magnitude of the update. The other parameters $\tilde{\theta}$ are optimized using the same optimizer as the vanilla NCE baselines in our experiments. The training method with a BAR update is provided in Algorithm 1. Note that, in practice, we apply the BAR update only once as we have noticed that to be enough to provide a performance boost over vanilla NCE.

Algorithm 1 Training NCE with BAR update

Input: Model parameters $\theta = [\tilde{\theta}, F]$, noise distribution Q, data samples D

Initialize $\tilde{\theta}$ optimizer

Initialize F optimizer as SGD

for *epoch* $\leftarrow 1$ **to** $epoch_{\max}$ **do**

$x_{p^*} \leftarrow D.\texttt{sample}()$

$x_q \leftarrow Q.\texttt{sample}()$

$\mathcal{L}_{Data}(x_{p^*}) \leftarrow -\frac{1}{2}\sum_{x_{p^*}} \log \frac{p_\theta(x)}{p_\theta(x)+q(x)}$

$\mathcal{L}_{Noise}(x_q) \leftarrow -\frac{1}{2}\sum_{x_q} \log \frac{q(x)}{p_\theta(x)+q(x)}$

$\mathcal{L}_{NCE}(x_q, x_{p^*}) \leftarrow \mathcal{L}_{Noise}(x_q) + \mathcal{L}_{Data}(x_{p^*})$

$\boxed{\nabla_F \leftarrow \log|\nabla_F \mathcal{L}_{Data}(x_{p^*})| - \log|\nabla_F \mathcal{L}_{Noise}(x_q)|}$

$\nabla_{\tilde{\theta}} \leftarrow \nabla_{\tilde{\theta}} \mathcal{L}_{NCE}(x_q, x_{p^*})$

$F \leftarrow \texttt{Update}(F, \nabla_F)$

$\tilde{\theta} \leftarrow \texttt{Update}(\tilde{\theta}, \nabla_{\tilde{\theta}})$

Output: Updated model parameters $\theta = [\tilde{\theta}, F]$

A binary search update for the log of the partition function (at a constant value of $\tilde{\theta}$) requires only one forward pass through the model as we only need to keep track of changing loss values with the change of the log of the partition function parameter. This results in very rapid computation of the ideal F with binary search.

4. Experimental results

We experimentally verify our method in both non-convex and convex settings. For all our experiments, we compare the performance of the BAR and binary search update of the log partition function coordinate to base NCE and NCE with increased learning rate on the log partition function coordinate. To implement binary search, we keep updating the log partition function parameter until the binary search loop returns the same value of the parameter (within a threshold) consecutively. Note that, while we compare our methods to vanilla NCE, our approach is orthogonal to other improvements to the NCE objective (Liu et al., 2021) and therefore could, in principle, be used in conjunction with these improvements.

4.1. BAR leads to quicker convergence on logZ values than NCE

First, we establish that BAR and binary search converge faster than NCE in a one dimensional setting, where the only parameter is F, the log of the partition function. Figure 1 show the trajectories of F while estimating the log partition function using BAR, binary search, and NCE respectively. The figure represent a total of 100 runs with a batch of 512 samples for each run. The data distribution is a mixture of 10 standard Gaussians in $\mathbb{R}^{20}$, where the means are randomly sampled and evenly distributed on a circle of radius 4 in the (x_1, x_2)-plane. The noise distribution is a standard Gaussian.

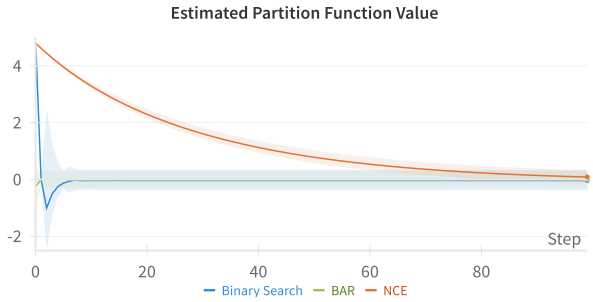

Figure 1. Trajectories of 100 runs for BAR, NCE and Binary Search. The y axis denotes predicted values of $\log Z$ and x axis denotes iterations.

Here, we can observe that for almost all runs BAR converges to a precision of 10^{-6} in less than 6 updates - a lot faster than NCE. In fact, in this case it is even quicker than binary search which takes around 20 steps to achieve same level of accuracy! Note that in the population limit with infinite precision, BAR will always converge in just one update. In fact, it has been theoretically proven that the iterative BAR procedure should converge with a finite number of samples (Meng & Wong, 1996). We provide a proof in Appendix B for completeness. Note that the convergence proof is also applicable to situations when data samples do not correspond to the density function used to compute the log of the partition function parameter.

In certain cases, we empirically observe that BAR is prone to oscillations around the true value. This is in the setting where the data and noise distributions have low overlap and their ratios are not well defined due to the constraints of numerical precision. In such cases, we show that the provided value is no further from the optimal value than the previous one during the iterative procedure.

4.2. Binary search and BAR updates perform better than NCE in convex settings

Following Liu et al. (2021), we run experiments in two convex settings. In first setting, we look at the exponential family given by $T(x) = [x, -1]$ and $q(x) = e^{-\frac{x^2}{2}}$, representing normal distributions with unit variance. The data and noise distributions are Gaussian with means separated by a distance of 16. In the second setting, the data and noise distributions are 16 dimensional Gaussians that share the same mean, but the noise is chosen to have an identity covariance matrix, while the data has a diagonal covariance matrix, with entries in $[6, 12]$. The exponential family corresponds to $T(x) = [x_1^2, \ldots, x_d^2, x_1, \ldots x_d, 1]$. The advantage of working in this setting is that we know what the best parameters θ_* are and we can evaluate performance of different methods through a comparison of distance of the learnt parameters θ from θ_*.

In both settings, we find that performing a BAR update or binary searching for optimal value of F (log of the partition function parameter) performs better than doing a joint gradient descent on the NCE loss. Although the results shown are obtained using SGD for optimization, we found that using other optimizers leads to similar trends, as long as the F parameter is handled separately. While using BAR, the log partition function coordinate F is updated using Equation (11), and the optimization algorithm is only used for $\tilde{\theta}$ coordinates. We keep a learning rate of 1 for all parameters of the model in this setting.

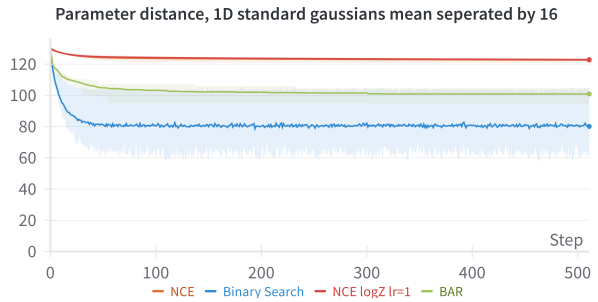

Figure 2. Results for estimating 1d Gaussian distribution at a distance of 16 from the noise Gaussian. Parameter distance $\|\theta - \theta_*\|_2$ is plotted along y axis and training steps along x. Mean results over 5 runs are shown with standard deviation being the shaded region

First, we consider the 1d case with low overlap: Gaussians with unit variance and a mean distance of 16. We expect NCE to perform poorly in this setting and so the gains made by our methods should become more visible. We also provide results for 1d cases with better overlap along with optimization trajectories in Appendix C.

Figure 2 shows the distance of learnt parameters from the true parameters. BAR and binary search do much better than vanilla NCE in this setting, indicated by the much lower parameter distance values with binary search showing the best performance. Of note, is the high variance in the performance of binary search along multiple runs. We conjecture that the main cause of these fluctuations is the low overlap between the two distributions. Due to the low overlap, the optimal value of the log partition function constant at each step is highly dependant on the points sampled and hence can vary largely from batch to batch. In contrast, for settings of good overlap, the value of the ideal log partition function constant would not vary as much and be less dependant on the batch of samples. We observe much lower variance in settings of good overlap for binary search. The results from the BAR update to the log partition function coordinate, on the other hand, has low variance in both settings. This could possibly be attributed to BAR being explicitly formulated to minimize variance of its estimates (Bennett, 1976).

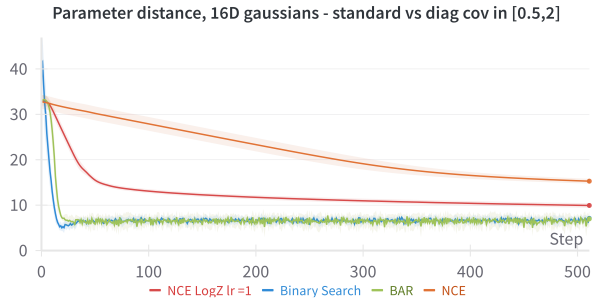

Figure 3. Results for estimating 16d Gaussian distribution. Parameter distance $\|\theta - \theta_*\|_2$ is plotted along y axis and training steps along x. Mean results over 5 runs are shown with standard deviation being the shaded region

The 16d case demonstrates better overlap, but with higher dimensions. Here, we can observe the benefit of using these methods over vanilla NCE scaling with dimensions. In this experiment we train the model parameters with a learning rate of 0.1, but also include an additional case where the learning rate of the log of the partition function parameter for base NCE is higher than the rest of the parameters.

Figure 3 shows results for this setting. BAR and binary search show much quicker convergence over vanilla NCE justifying the use of these methods even in settings with good overlap.

4.3. Binary Search and BAR show greater performance with neural networks

For non convex settings, we train neural networks on the NCE objective function. We train models on toy 2D systems where the optimization surface is relatively simple and also show performance trends on higher dimension datasets such

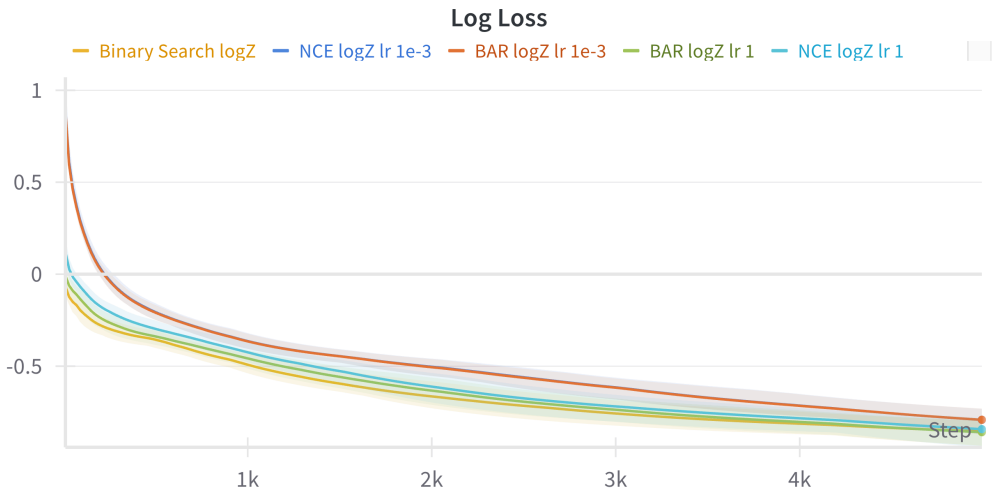

Figure 4. Log of the NCE loss while training the Neural Network on the 8-Gaussians toy system with the different methods mentioned in the legend. The other parameters of the neural network are trained with a learning rate of 10^{-3}

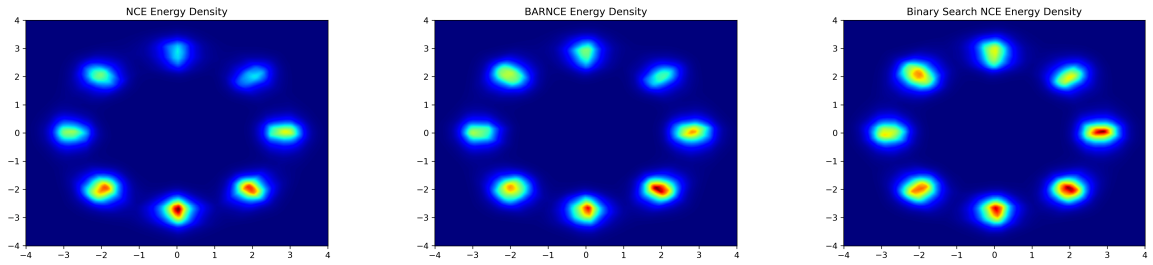

Figure 5. Energy density function on the 8-Gaussian 2D toy system learnt by the neural network through NCE (left), BAR (center), and Binary search (right). Ground truth energy has the same intensity for all 8 Gaussians.

as MNIST. For all of our experiments we also compare to using NCE with a higher learning rate on the log of the partition function coordinate (F) as we observe that even making just that minor adjustment leads to improved performance with NCE.

For 2D toy systems, we train a neural network to learn density functions on the 8-gaussians and pinwheel toy 2D system using an isotropic Gaussian with diagonal covariance values of 2 as the base noise distribution. The neural network is a very basic 2 layer MLP with 128 and 64 hidden dimensions and leaky relu as activation function. The loss curves and learnt energy functions on the 8-Gaussians toy system are shown in Figure 11 and Figure 12. The marginal gains in the loss for binary search and BAR is not entirely surprising as this is a relatively easy task with low dimensions and good overlap. However, on visualization of the learnt energy function we can qualitatively say that BAR and binary search have made more progress than NCE with the same number of steps. We provide additional results on the pinwheel toy system in Appendix D.1 that indicate the same trend.

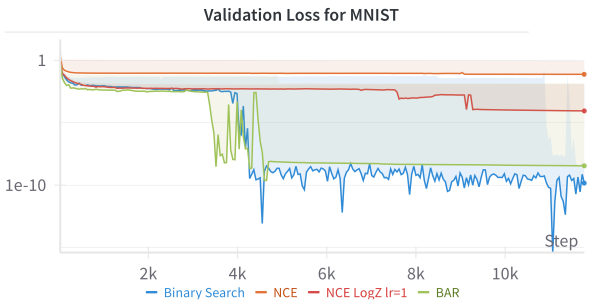

Figure 6. NCE loss of all the methods while training on the MNIST dataset

Finally, we compare our methods to vanilla NCE on the MNIST dataset to check performance in high dimensions and low overlap setting. Following Rhodes et al. (2020)

we use a resnet-18 neural network and quadratic heads. Thus, we represent the log density ratio $\log(p_\theta) - \log(q)$ as $f^\top W f + b^\top f + c$, where f denotes the resnet feature map. As is done in Rhodes et al. (2020) we restrict the matrix to $W \succ 0$. Finally, the distribution q is chosen to be a Gaussian with matching mean and covariance to a *dequantized* version of the MNIST dataset.

Figure 6 shows the loss curves of training the neural networks with the various methods discussed above. We observe that both BAR and binary search make significant progress over the usual NCE. They optimize the neural network to a loss that is 2-3 orders of magnitude better than that obtained with base NCE. To ensure that this phenomenon does not happen due to learning rate differences, we also compare to NCE trained with a higher learning rate for the log partition function coordinate parameter. Note that this is the same learning rate we use for the BAR update. This curve follows previously observed trends where it does better than base NCE but is still outperformed by the BAR/binary search update.

5. Conclusion

In this work, we aim to improve NCE training dynamics through explicit treatment of the log partition function coordinate. We notice that the NCE objective function is always convex along that coordinate when the rest of the parameters are kept fixed. We observe a measurable improvement in training dynamics when we solve for that optimal value of the log partition function parameter explicitly using binary search or approximate it using the Bennett Acceptance Ratio. Empirical results across various settings, both with low and high overlap and in both convex and non-convex scenarios, consistently demonstrate superior performance compared to the NCE baseline.

While we have strong empirical evidence that such updates work over the NCE baseline, we do not currently have a working theory on why it shows this behaviour and so we will be exploring that further to get a more principled understanding of the optimization dynamics. We are also interested in seeing what the effects of these updates would be when used along with other recent improvements in the NCE objective function such as those suggested by Liu et al. (2021). Finally, we would like to explore applying these developments to calculating partition functions of physical systems so that we can obtain important thermodynamic quantites such as the free energy.

Acknowledgements

This work is funded through R35GM140753 from the National Institute of General Medical Sciences and NSF2238125 from the National Science Foundation. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institute of General Medical Sciences or the National Institutes of Health.

References

Bennett, C. H. Efficient estimation of free energy differences from monte carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976.

Chehab, O., Hyvarinen, A., and Risteski, A. Provable benefits of annealing for estimating normalizing constants: Importance sampling, noise-contrastive estimation, and beyond. *Advances in Neural Information Processing Systems*, 36, 2024.

Dyer, C. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.

Feeney, P. and Hughes, M. C. Sincere: Supervised information noise-contrastive estimation revisited. *arXiv preprint arXiv:2309.14277*, 2023.

Gao, R., Nijkamp, E., Kingma, D. P., Xu, Z., Dai, A. M., and Wu, Y. N. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7518–7528, 2020.

Gelman, A. and Meng, X.-L. Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical science*, pp. 163–185, 1998.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

Gutmann, M. and Hirayama, J.-i. Bregman divergence as general framework to estimate unnormalized statistical models. *arXiv preprint arXiv:1202.3727*, 2012.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of machine learning research*, 13(2), 2012.

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pp. 4182–4192. PMLR, 2020.

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Kong, L., de Masson d'Autume, C., Yu, L., Ling, W., Dai, Z., and Yogatama, D. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Syx79eBKwr.

Liu, B., Rosenfeld, E., Ravikumar, P., and Risteski, A. Analyzing and improving the optimization landscape of noise-contrastive estimation. *arXiv preprint arXiv:2110.11271*, oct 2021. URL http://arxiv.org/abs/2110.11271v1.

Meng, X.-L. and Wong, W. H. Simulating ratios of normalizing constants via a simple identity: a theoretical exploration. *Statistica Sinica*, pp. 831–860, 1996.

Menon, A. and Ong, C. S. Linking losses for density ratio and class-probability estimation. In *International Conference on Machine Learning*, pp. 304–313. PMLR, 2016.

Mnih, A. and Kavukcuoglu, K. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26:2265–2273, 2013.

Mnih, A. and Teh, Y. W. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 1751–1758, 2012.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. *Advances in Neural Information Processing Systems*, 33:4905–4916, 2020.

Sugiyama, M., Suzuki, T., and Kanamori, T. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, USA, 1st edition, 2012. ISBN 0521190177.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding, 2020.

Uehara, M., Kanamori, T., Takenouchi, T., and Matsuda, T. A unified statistically efficient estimation framework for unnormalized models. In *International Conference on Artificial Intelligence and Statistics*, pp. 809–819. PMLR, 2020.

Xu, N. Self-adapting noise-contrastive estimation for energy-based models. *arXiv preprint arXiv:2211.02650*, 2022.

A. Convexity proofs

A.1. NCE objective is convex for exponential families

Proof of NCE loss being convex on the exponential family:

The NCE loss is given by:

$$\mathcal{L}_{\texttt{NCE}}(p_\theta) = -\frac{1}{2}\mathbb{E}_{p_*}\left[\log \frac{p_\theta}{p_\theta + q}\right] - \frac{1}{2}\mathbb{E}_q\left[\log \frac{q}{p_\theta + q}\right] \tag{12}$$

$p_\theta = \exp(T(x)^\intercal\theta)$ where $T(x) = [\tilde{T}(x), -1]$ and $\theta = [\tilde{\theta}, F]$. The gradient of the objective function is:

$$\begin{aligned}
\nabla_\theta p_\theta(x) =& p_\theta(x).T(x) \\
\nabla_\theta L(\theta) =& -\frac{1}{2}\nabla_\theta\left[\mathbb{E}_{p_*}\log\frac{p_\theta}{p_\theta+q} + \mathbb{E}_q\log\frac{q}{p_\theta+q}\right] \\
=& \frac{1}{2}\nabla_\theta\left[\mathbb{E}_{p_*}\log\frac{p_\theta+q}{p_\theta} + \mathbb{E}_q\log\frac{p_\theta+q}{q}\right] \\
=& \frac{1}{2}\left[\mathbb{E}_{p_*}\frac{p_\theta}{(p_\theta+q)}\frac{-p_\theta-q+p_\theta}{p^2}\nabla_\theta p_\theta + \mathbb{E}_q\frac{q}{p_\theta+q}\frac{1}{q}\nabla_\theta p_\theta\right] = \frac{1}{2}\int_x \frac{q}{p_\theta+q}(p_\theta - p_*)T(x)dx
\end{aligned} \tag{13}$$

The corresponding Hessian for the objective is:

$$\begin{aligned}
\nabla^2_\theta L(\theta) =& \frac{1}{2}\int_x\left(-\frac{q(p_\theta-p_*)}{(p_\theta+q)^2}\nabla_\theta p_\theta + \frac{q}{p_\theta+q}\nabla_\theta p_\theta\right)T(x)dx \\
=& \frac{1}{2}\int_x \frac{q}{p_\theta+q}\cdot\frac{p_*+q}{p_\theta+q}\cdot p_\theta\cdot T(x)T(x)^\top dx = \frac{1}{2}\int_x\frac{(p_*+q)p_\theta q}{(p_\theta+q)^2}T(x)T(x)^\top dx
\end{aligned} \tag{14}$$

Since the Hessian is Positive Semi-Definite, the objective function is convex for exponential family of distributions.

A.2. Binary search update of logZ is convex for exponential families

We work with $p_\theta = \exp(T(x)^\intercal\theta - F_\theta)$. Note that here F_θ is a function of θ and samples in our batch. We know that we can find a value of F_θ upto machine precision that makes the NCE derivative (Equation 3) go to zero. Here, we show this update still maintains a convex surface for exponential families.

The gradient of the objective function in this representation is

$$\begin{aligned}
\nabla_\theta p_\theta(x) =& p_\theta(x).(T(x) - \nabla_\theta F_\theta) \\
\nabla_\theta L(\theta) =& -\frac{1}{2}\nabla_\theta\left[\mathbb{E}_{p_*}\log\frac{p_\theta}{p_\theta+q} + \mathbb{E}_q\log\frac{q}{p_\theta+q}\right] \\
=& \frac{1}{2}\nabla_\theta\left[\mathbb{E}_{p_*}\log\frac{p_\theta+q}{p_\theta} + \mathbb{E}_q\log\frac{p_\theta+q}{q}\right] \\
=& \frac{1}{2}\left[\mathbb{E}_{p_*}\frac{p_\theta}{(p_\theta+q)}\frac{-p_\theta-q+p_\theta}{p^2}\nabla_\theta p_\theta + \mathbb{E}_q\frac{q}{p_\theta+q}\frac{1}{q}\nabla_\theta p_\theta\right] = \frac{1}{2}\int_x\frac{q(p_\theta-p_*)}{p_\theta+q}(T(x)-\nabla_\theta F_\theta)dx
\end{aligned} \tag{15}$$

The Hessian then is:

$$\begin{aligned}
\nabla^2_\theta L(\theta) =& \frac{1}{2}\int_x\left(-\frac{q(p_\theta-p_*)}{(p_\theta+q)^2}\nabla_\theta p_\theta + \frac{q}{p_\theta+q}\nabla_\theta p_\theta\right)(T(x)-\nabla_\theta F_\theta)dx - \frac{1}{2}\int_x\frac{q(p_\theta-p_*)}{p_\theta+q}\nabla^2_\theta F_\theta dx \\
=& \frac{1}{2}\int_x\frac{q}{p_\theta+q}\cdot\frac{p_*+q}{p_\theta+q}\cdot p_\theta\cdot(T(x)-\nabla_\theta F_\theta)(T(x)-\nabla_\theta F_\theta)^\top dx - \frac{1}{2}\int_x\frac{q(p_\theta-p_*)}{p_\theta+q}\nabla^2_\theta F_\theta dx \\
=& \frac{1}{2}\int_x\frac{(p_*+q)p_\theta q}{(p_\theta+q)^2}(T(x)-\nabla_\theta F_\theta)(T(x)-\nabla_\theta F_\theta)^\top dx - \frac{1}{2}\int_x\frac{q(p_\theta-p_*)}{p_\theta+q}\nabla^2_\theta F_\theta dx
\end{aligned} \tag{16}$$

The right integral becomes zero on the binary search update to F_θ and the left integral is positive semi definite, therefore the optimization surface remains convex after the binary search update.

B. Comments on convergence of BAR

Lemma B.1. *Let $x_i, y_i \in \mathbb{R}$ for $i = 1, \dots, n$. Let f, q be any functions that are positive on the set $\{x_1, \dots, x_n, y_1, \dots, y_n\}$, satisfying Bennette's ratio condition, that is*

$$\frac{\frac{1}{n}\sum_i \frac{q(x_i)}{f(x_i)+q(x_i)}}{\frac{1}{n}\sum_i \frac{f(y_i)}{f(y_i)+q(y_i)}} = 1. \tag{17}$$

Then for any function g such that $g(x) = Z \cdot f(x)$, the value $\bar{Z}$ obtained by using the bar update rule

$$\log \bar{Z} = \log Z + \log\left(\frac{1}{n}\sum_i \frac{q(x_i)}{g(x_i)+q(x_i)}\right) - \log\left(\frac{1}{n}\sum_i \frac{g(y_i)}{g(y_i)+q(x_i)}\right)$$

satisfies $\left|\log \bar{Z}\right| \le |\log Z|$.

Proof. Let $F = \log Z$. Then $g(x) = e^F f(x)$. Since $f(x)$ and $q(x)$ are positive, it follows that

$$\min(1, e^F)(f(x)+q(x)) \le e^F f(x) + q(x) \le \max(1, e^F)(f(x)+q(x)).$$

Therefore,

$$\frac{1}{\max(1,e^F)}\left(\frac{1}{n}\sum_i \frac{q(x_i)}{f(x_i)+q(x_i)}\right) \le \left(\frac{1}{n}\sum_i \frac{q(x_i)}{g(x_i)+q(x_i)}\right) \le \frac{1}{\min(1,e^F)}\left(\frac{1}{n}\sum_i \frac{q(x_i)}{f(x_i)+q(x_i)}\right),$$

and similarly,

$$\frac{e^F}{\max(1,e^F)}\left(\frac{1}{n}\sum_i \frac{f(x_i)}{f(x_i)+q(x_i)}\right) \le \left(\frac{1}{n}\sum_i \frac{g(x_i)}{g(x_i)+q(x_i)}\right) \le \frac{e^F}{\min(1,e^F)}\left(\frac{1}{n}\sum_i \frac{f(x_i)}{f(x_i)+q(x_i)}\right).$$

Combining both the inequalities,

$$\frac{1}{e^F}\frac{\min(1,e^F)}{\max(1,e^F)}\left(\frac{\frac{1}{n}\sum_i \frac{q(x_i)}{f(x_i)+q(x_i)}}{\frac{1}{n}\sum_i \frac{f(x_i)}{f(x_i)+q(x_i)}}\right) \le \frac{\frac{1}{n}\sum_i \frac{q(x_i)}{g(x_i)+q(x_i)}}{\frac{1}{n}\sum_i \frac{g(x_i)}{g(x_i)+q(x_i)}} \le \frac{1}{e^F}\frac{\max(1,e^F)}{\min(1,e^F)}\left(\frac{\frac{1}{n}\sum_i \frac{q(x_i)}{f(x_i)+q(x_i)}}{\frac{1}{n}\sum_i \frac{f(x_i)}{f(x_i)+q(x_i)}}\right)$$

Note that $\max(1, e^F) = e^{|F|}\min(1, e^F)$. Further, since f satisfies Equation (17), we get

$$-F - |F| \le \log\left(\frac{1}{n}\sum_i \frac{q(x_i)}{g(x_i)+q(x_i)}\right) - \log\left(\frac{1}{n}\sum_i \frac{g(y_i)}{g(y_i)+q(x_i)}\right) \le -F + |F|.$$

Adding $F = \log Z$ to the expression, we get

$$-|F| \le \log Z + \log\left(\frac{1}{n}\sum_i \frac{q(x_i)}{g(x_i)+q(x_i)}\right) - \log\left(\frac{1}{n}\sum_i \frac{g(y_i)}{g(y_i)+q(x_i)}\right) - \log Z_* \le |F|.$$

Note that the middle term is precisely $\log \bar{Z}$, giving us

$$\left|\log \bar{Z}\right| \le |F|,$$

which completes the proof since $F = \log Z$. □

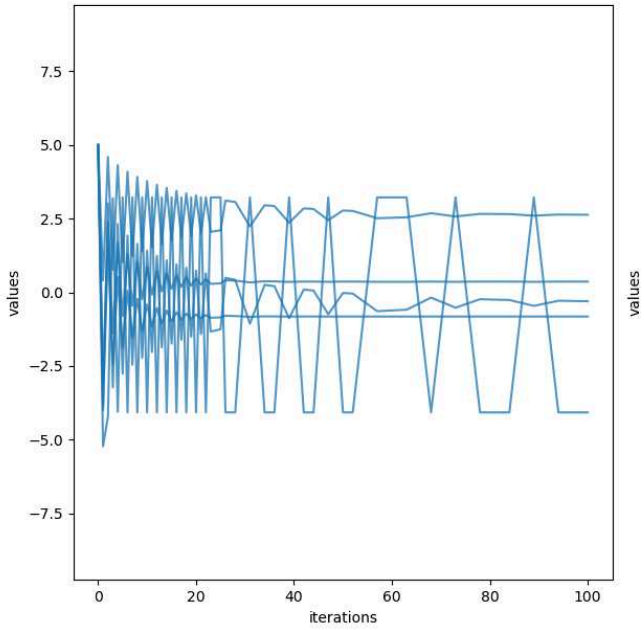

Figure 7. BAR shows oscillating behavior in case of low overlap between data and noise distributions.

Specifically, when x_i are samples from some distribution P_* and y_i are samples from some distribution Q, and if p_θ where $\theta = [\tilde{\theta}, F_*]$ satisfies the Bennette's ratio condition, then for any other F, the BAR update specified in Equation (10), given by

$$\bar{F} = F + \log \mathbb{E}_{p_*}\left[\frac{q}{p_{\tilde{\theta},F} + q}\right] - \log \mathbb{E}_q\left[\frac{p_{\tilde{\theta},F}}{p_{\tilde{\theta},F} + q}\right]$$

satisfies $\left|\bar{F} - F_*\right| \leq |F - F_*|$, in population as well as when the expectations are estimated using a finite number of samples.

This proof shows that the BAR update is almost a contraction mapping, and the statement of lemma is tight unless further assumptions are made on x_i and y_i. Experimentally, in low overlap cases, we encounter situations where BAR update is not a contraction mapping as shown in the Figure 7.

C. Supplementary Results on Exponential Family

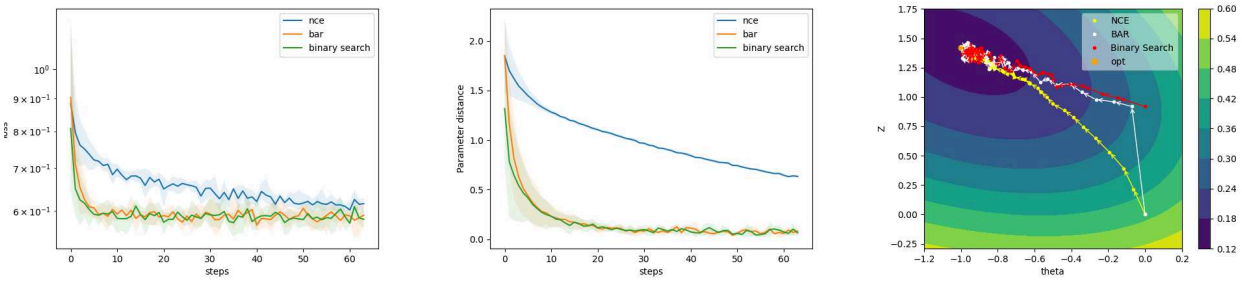

Figure 8. NCE loss (left), parameter distance (center), and an example parameter trajectory for 1d setting where the means of the two distributions are at a distance of 1 from each other. The orange dot in the trajectory figures represents the optimal value of the parameters in that setting.

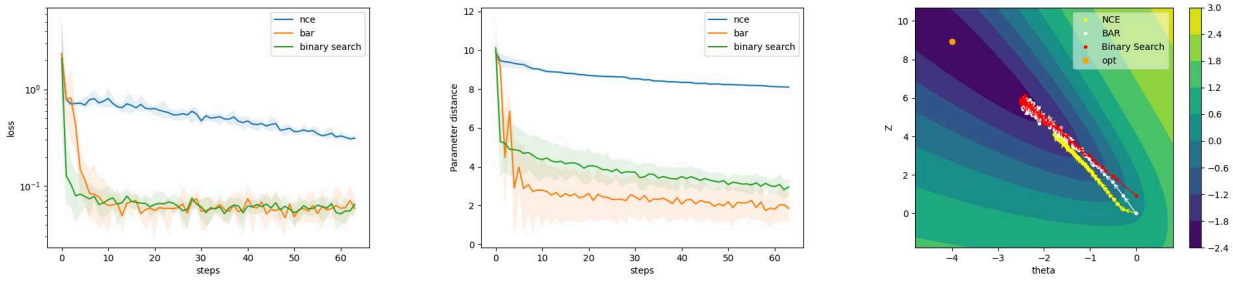

Figure 9. NCE loss (left), parameter distance (center), and an example parameter trajectory for 1d setting where the means of the two distributions are at a distance of 4 from each other. The orange dot in the trajectory figures represents the optimal value of the parameters in that setting.

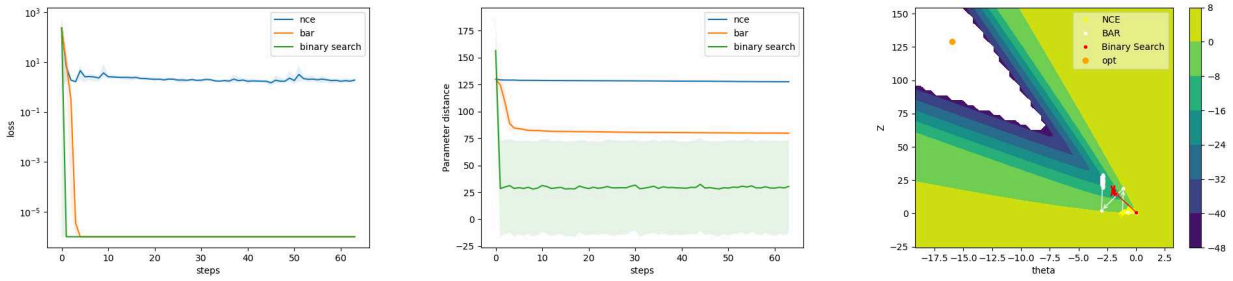

Figure 10. NCE loss (left), parameter distance (center), and an example parameter trajectory for 1d setting where the means of the two distributions are at a distance of 16 from each other. The orange dot in the figure on the right represents the optimal value of the learn parameters.

Results for NCE, BAR and binary search on varying levels of overlap in 1d exponential family settings. All experiments show a consistent improvement of BAR and binary search over base NCE. While, there isnt much to take from the trajectory plots, its worthwhile to note that both BAR and binary search make a big jump towards the right value of Z in the initial steps itself for a high overlap problem (Figure 8). We also notice in Figure 9, the parameter distance of BAR is lower than that of binary search. This is an indication that there is still some task specific variance between the performance of BAR and binary search that needs to be explored more.

D. Supplementary Results on Neural Networks

D.1. Neural Network training on toy 2D pinwheel system

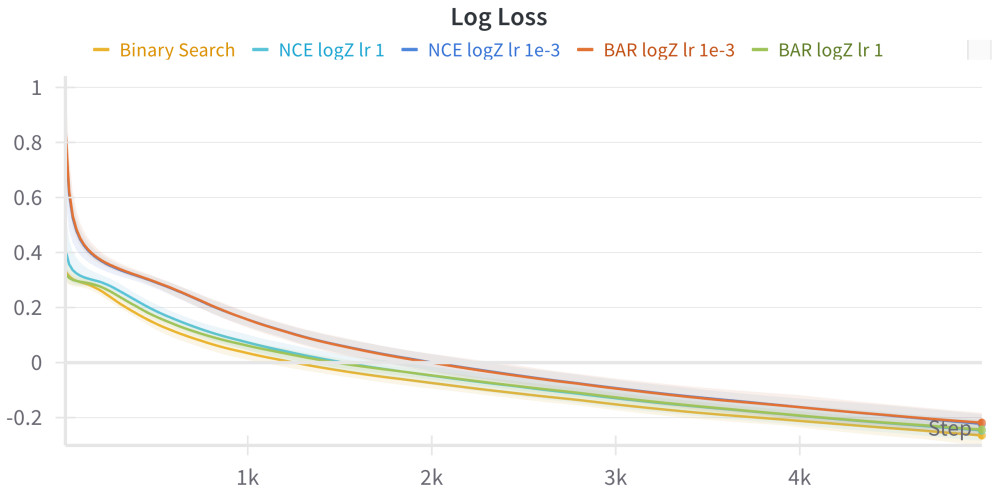

Figure 11. Log of the NCE loss while training the Neural Network on the 8-Gaussians toy system with the different methods mentioned in the legend. The other parameters of the neural network are trained with a learning rate of $1e-3$

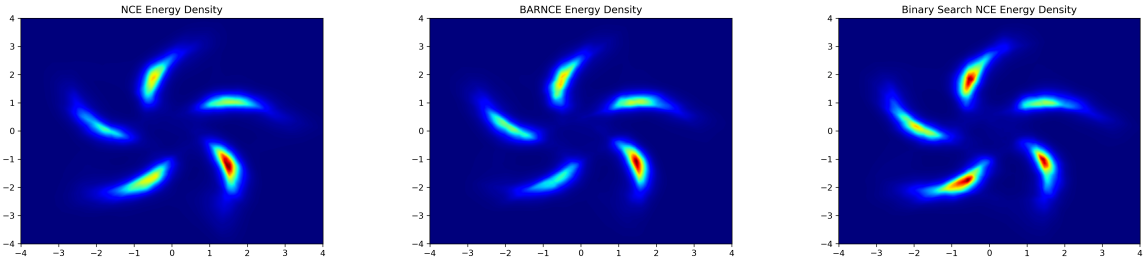

Figure 12. Energy density function on the Pinwheel 2D toy system learnt by the neural network through NCE (left), BAR (center), and Binary search (right). Ground truth energy has the same intensity for all 5 pinwheel petals.

D.2. Example of generated samples after training on MNIST

Figure 13. MNIST images sampled from a learnt energy function

We show in Figure 13 some generated samples via running MCMC chains on the energy function learnt by the neural network that used the BAR update for its log partition function parameter while training.