Differentiable Marginal Likelihood Estimation and the Perils of Gradient Noise

Anonymous Author(s)
Affiliation
Address
email

Abstract

Annealed importance sampling (AIS) and related algorithms are highly effective tools for marginal likelihood estimation, but are not fully differentiable due to the use of Metropolis-Hastings (MH) correction steps. Differentiability is a desirable property as it would admit the possibility of optimizing marginal likelihood as an objective using gradient-based methods. To this end, we propose a differentiable AIS algorithm by abandoning MH steps, which further unlocks mini-batch computation. We provide a detailed convergence analysis for Bayesian linear regression which goes beyond previous analyses by explicitly accounting for non-perfect transitions. Using this analysis, we prove that our algorithm is consistent in the full-batch setting and provide a sublinear convergence rate. However, we show that the algorithm is inconsistent when mini-batch gradients are used due to a fundamental incompatibility between the goals of last-iterate convergence to the posterior and elimination of the pathwise stochastic error. This result is in stark contrast to our experience with stochastic optimization and stochastic gradient Langevin dynamics, where the effects of gradient noise can be washed out by taking more steps of a smaller size. Our negative result relies crucially on our explicit consideration of convergence to the stationary distribution, and it helps explain the difficulty of developing practically effective AIS-like algorithms that exploit mini-batch gradients.

1 Introduction

Marginal likelihood (ML), sometimes called *evidence*, is a central quantity in Bayesian learning as it measures how well a model can describe a particular data set. It has several appealing properties. First, it can be plugged into Bayes' rule to compute the posterior over models. Second, it is closely related to Occam's Razor [Rasmussen and Ghahramani, 2001] and manages the tradeoff between model complexity and the goodness of fit to the data. Hence, it can be used as a metric to compare different models. For example, it is commonly used to select hyperparameters for Gaussian processes [Rasmussen, 2003] and network structures for Bayesian networks [Teyssier and Koller, 2005], where either closed-form solutions or accurate, tractable approximations are available.

However, it is often the case that computing ML is computationally intractable, as it commonly involves summation or integration over high-dimensional model parameters or latent variables. In this case, one must resort to numerical approximation methods. In the context of model comparison (e.g., evaluating generative models [Wu et al., 2016, Huang et al., 2020]), annealed importance sampling (AIS) [Neal, 2001] is arguably the most popular algorithm. Notably, AIS is closely related to other generic ML estimators that yield accurate estimation [Grosse et al., 2015], including Sequential Monte Carlo (SMC) [Doucet et al., 2001] and nested sampling [Skilling et al., 2006]. Given enough computing time, AIS is able to produce accurate estimates of marginal likelihood (it converges to the true ML value quickly by adding more intermediate distributions).

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

AIS alternates between Markov chain Monte Carlo (MCMC) transitions and importance sampling updates, where the MCMC step typically involves a non-differentiable Metropolis-Hastings (MH) correction. Unfortunately, the MH step is generally incompatible with gradient-based optimization and complicates theoretical analysis. To resolve these issues, we marry AIS with Hamiltonian Monte Carlo (HMC) [Neal et al., 2011] and derive an unbiased yet differentiable ML estimator named differentiable AIS (DAIS) by removing the MH correction step, which further unlocks the possibility of mini-batch computation. Moreover, our algorithm can be made memory-efficient by caching noise and simulating Hamiltonian dynamics in reverse [Maclaurin et al., 2015]. We further prove that DAIS is asymptotically consistent with full-batch gradients in the case of Bayesian linear regression, inheriting the convergence property of AIS. In particular, we provide a sublinear convergence rate by explicitly accounting for non-perfect transitions[1].

Furthermore, motivated by the problem of learning from large-scale datasets, we study a stochastic variant of our algorithm that uses gradients estimated from a subset of the dataset. Given the success of stochastic optimization [Robbins and Monro, 1951] and stochastic gradient MCMC algorithms [Welling and Teh, 2011, Chen et al., 2014], one may presume that stochastic gradient DAIS performs well. Surprisingly, the natural implementation of this algorithm can be arbitrarily bad. In particular, we show that DAIS with stochastic gradients is inconsistent due to a fundamental incompatibility between the goals of last-iterate convergence to the posterior and elimination of the pathwise stochastic error. This is in stark contrast with other settings such as gradient-based optimization and Langevin dynamics, where the gradient noise can be washed out by taking smaller steps. This indicates that ML estimation with stochastic gradients may require new ideas.

We validate our theoretical analysis with simulations. We also demonstrate empirically that DAIS can be applied to variational autoencoders (VAEs) [Kingma and Welling, 2013] for a tighter evidence lower bound, which in turn leads to improved performance compared to vanilla VAE. We also compare to importance weighted autoencoders (IWAE) [Burda et al., 2016]. While IWAE is more effective with a low compute budget, we show that DAIS eventually outperforms IWAE as compute increases. Finally, like AIS, DAIS can be used to evaluating generative models. We show that it performs on par with AIS despite the removal of the MH correction step and outperforms the IWAE bound by a large margin.

2 Background

2.1 Marginal Likelihood Estimation

For a model $\mathcal{M}$ and observed data $\mathcal{D}$, one can define marginal likelihood (ML) as

$$p(\mathcal{D}|\mathcal{M}) = \int p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M}) d\boldsymbol{\theta} = \int p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M}) d\boldsymbol{\theta}, \tag{1}$$

where $\boldsymbol{\theta}$ denotes the parameters of the model. ML estimation is often regarded as the same problem as estimating the partition function of an unnormalized distribution. Given a distribution defined as $p(\boldsymbol{\theta}) = f(\boldsymbol{\theta})/\mathcal{Z}$ where the unnormalized density $f(\boldsymbol{\theta})$ can be efficiently computed, we are interested in estimating the partition function $\mathcal{Z} = \int f(\boldsymbol{\theta}) d\boldsymbol{\theta}$. In (1), the $f(\boldsymbol{\theta})$ is $p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M})$. In this paper, we find it convenient to focus on discussing ML estimation because we later will discuss the effect of noise arising from data subsampling.

It is often the case that computing ML is computationally intractable. One approach is to approximate (1) with Monte Carlo methods. In particular, one can approximate the integration using importance sampling:

$$p(\mathcal{D}|\mathcal{M}) = \mathbb{E}_{q(\boldsymbol{\theta})} \left[\frac{p(\mathcal{D}|\boldsymbol{\theta}, \mathcal{M}) p(\boldsymbol{\theta}|\mathcal{M})}{q(\boldsymbol{\theta})} \right] \approx \frac{1}{S} \sum_{i=1}^{S} \frac{p(\mathcal{D}|\boldsymbol{\theta}_i, \mathcal{M}) p(\boldsymbol{\theta}_i|\mathcal{M})}{q(\boldsymbol{\theta}_i)} \quad \text{with } \boldsymbol{\theta}_i \sim q(\boldsymbol{\theta}) \tag{2}$$

However, this estimation can exhibit high variance for small or medium S when the target distribution $p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M})$ and proposal distribution $q(\boldsymbol{\theta})$ are dissimilar. One important exception is when the proposal distribution is exactly the posterior $p(\boldsymbol{\theta}|\mathcal{D}, \mathcal{M})$.

[1] While Neal [2001] analyzed the convergence rate of AIS, his analysis assumed perfect transitions. In this paper, we show that accounting for the convergence of concrete MCMC algorithms is crucial for understanding the practical performance.

2.2 Annealed Importance Sampling

Annealed importance sampling (AIS) is an algorithm which estimates the ML by gradually changing, or "annealing", a distribution. Formally, the algorithm takes in a sequence of distributions $p_0, \dots, p_K$, with $p_k(\boldsymbol{\theta}) = f_k(\boldsymbol{\theta})/\mathcal{Z}_k$ and $\mathcal{Z}_k = \int f_k(\boldsymbol{\theta}) d\boldsymbol{\theta}$. In the context of ML estimation, the starting distribution f_0 is the tractable prior distribution $p(\boldsymbol{\theta}|\mathcal{M})$ with $\mathcal{Z}_0 = 1$, while the target distribution f_K is $p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M})$ with $\mathcal{Z}_K = p(\mathcal{D}|\mathcal{M})$. For each p_k, one must also specify an MCMC transition operator $\mathcal{T}_k$ which leaves p_k invariant.

The output of AIS is an unbiased estimate $\hat{\mathcal{Z}}_K$ of the exact ML $\mathcal{Z}_K$. Importantly, unbiasedness holds for any finite K, as shown in Neal [2001]. Moreover, AIS can be viewed as importance sampling over an extended space [Neal, 2001]. In particular, we have $\mathcal{Z}_K = \mathbb{E}_{q_{\text{fwd}}}[q_{\text{bwd}}/q_{\text{fwd}}]$ with the target and proposal distributions defined as

$$q_{\text{fwd}}(\boldsymbol{\theta}_{0:K}) = p_0(\boldsymbol{\theta}_0)\mathcal{T}_1(\boldsymbol{\theta}_1|\boldsymbol{\theta}_0)\cdots\mathcal{T}_K(\boldsymbol{\theta}_K|\boldsymbol{\theta}_{K-1}) \tag{3}$$

$$q_{\text{bwd}}(\boldsymbol{\theta}_{0:K}) = f_K(\boldsymbol{\theta}_K)\tilde{\mathcal{T}}_K(\boldsymbol{\theta}_{K-1}|\boldsymbol{\theta}_K)\cdots\tilde{\mathcal{T}}_1(\boldsymbol{\theta}_0|\boldsymbol{\theta}_1), \tag{4}$$

where $\mathcal{T}_k(\boldsymbol{\theta}|\boldsymbol{\theta}')$ is a forward MCMC kernel and $\tilde{\mathcal{T}}_k(\boldsymbol{\theta}'|\boldsymbol{\theta}) = \mathcal{T}_k(\boldsymbol{\theta}|\boldsymbol{\theta}')p_k(\boldsymbol{\theta}')/p_k(\boldsymbol{\theta})$ is the corresponding reverse MCMC kernel. Here, q_{fwd} represents the chain of states generated by AIS, and q_{bwd} is a fictitious (unnormalized) reverse chain which begins with a sample from p_K and applies the transitions in reverse order. In practice, the intermediate distributions have to be chosen carefully for a low variance estimate $\hat{\mathcal{Z}}_K$. One typically uses geometric averages of the initial and target distributions:

$$p_k(\boldsymbol{\theta}) = p_{\beta_k}(\boldsymbol{\theta}) = f_{\beta_k}(\boldsymbol{\theta})/\mathcal{Z}_{\beta_k} = f_0(\boldsymbol{\theta})^{1-\beta_k} f_K(\boldsymbol{\theta})^{\beta_k}/\mathcal{Z}_{\beta_k} = p(\boldsymbol{\theta}|\mathcal{M})p(\mathcal{D}|\boldsymbol{\theta},\mathcal{M})^{\beta_k}/\mathcal{Z}_{\beta_k} \tag{5}$$

where $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$ is the annealing schedule. Indeed, AIS gives an unbiased estimate $\hat{\mathcal{Z}}$ of $\mathcal{Z}$. However, as $\mathcal{Z}$ can vary over many orders of magnitude, it is often more meaningful to talk about estimating $\log \mathcal{Z}$. Unfortunately, unbiased estimators of $\mathcal{Z}$ can result in biased estimators of $\log \mathcal{Z}$ because $\mathbb{E} \log \hat{\mathcal{Z}} \leq \log \mathbb{E}\hat{\mathcal{Z}}$ by Jensen's inequality, resulting in only a lower bound. In particular, we have the AIS bound

$$\mathbb{E}_{q_{\text{fwd}}} \log \hat{\mathcal{Z}}_K = \sum_{k=1}^{K} \mathbb{E}_{q_{\text{fwd}}}\left[\log f_{\beta_k}(\boldsymbol{\theta}_{k-1}) - \log f_{\beta_{k-1}}(\boldsymbol{\theta}_{k-1})\right] \tag{6}$$

$$= \sum_{k=1}^{K} (\beta_k - \beta_{k-1}) \mathbb{E}_{q_{\text{fwd}}}\left[\log p(\mathcal{D}|\boldsymbol{\theta}_{k-1}, \mathcal{M})\right] \tag{7}$$

where (5) facilitated the simplification from (6) to (7). Of course, it is not enough to have a lower bound; we would also like the estimates to be close to the true value. Fortunately, AIS is *consistent* in that it converges to the correct value in the limit of infinite intermediate distributions.

3 Differentiable Annealed Importance Sampling

In this section, we motivate and derive a differentiable AIS (DAIS) algorithm for ML estimation. We also discuss its application to variational inference and a memory-efficient implementation.

Ideally, assuming a continuously parameterized model class (e.g., variational autoencoder), we would like to differentiate through the lower bound (7) to find an optimal model $\mathcal{M}$. However, AIS is almost always instantiated with the use of an MCMC transition kernel $\mathcal{T}_k$ that satisfies detailed balance to ensure it leaves p_k invariant. In practice, this is typically achieved by using a MH step, which is generally not differentiable.[2] We thus remove the MH correction and, in particular, specify each transition to consist

Algorithm 1 Differentiable AIS (DAIS)

$\boldsymbol{\theta}_0, \mathbf{v}_0$ sample from $p_0(\boldsymbol{\theta}), \pi \triangleq \mathcal{N}(\mathbf{0}, \mathbf{M})$
$\mathcal{L}_{\text{DAIS}} = -\log p_0(\boldsymbol{\theta}_0)$
for $k = 1, \dots, K$ **do**
  $\boldsymbol{\theta}_{k-\frac{1}{2}} \leftarrow \boldsymbol{\theta}_{k-1} + \frac{\eta}{2}\mathbf{M}^{-1}\mathbf{v}_{k-1}$
  $\hat{\mathbf{v}}_k \leftarrow \mathbf{v}_{k-1} + \eta \nabla \log f_{\beta_k}(\boldsymbol{\theta}_{k-\frac{1}{2}})$
  $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_{k-\frac{1}{2}} + \frac{\eta}{2}\mathbf{M}^{-1}\hat{\mathbf{v}}_k$
  $\mathbf{v}_k \leftarrow \gamma\hat{\mathbf{v}}_k + \sqrt{1-\gamma^2}\boldsymbol{\varepsilon},\ \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{M})$
  $\mathcal{L}_{\text{DAIS}} \mathrel{+}= \log \pi(\hat{\mathbf{v}}_k)) - \log \pi(\mathbf{v}_{k-1})$
end for
return $\mathcal{L}_{\text{DAIS}} \mathrel{+}= \log p(\mathcal{D}, \boldsymbol{\theta}_K|\mathcal{M})$

[2] The discontinuity introduced by the MH step makes it hard to use the reparameterization trick, though this can be done using a delicate gradient estimator [Naesseth et al., 2017]. This is orthogonal to our work and our fix is simpler and more elegant.

of a deterministic leapfrog integration step followed by a stochastic partial momentum refreshment [Horowitz, 1991]. Algorithm 1 details the simulation of Hamiltonian dynamics using such transitions. With $\gamma = 0$, the algorithm is essentially unadjusted Langevin dynamics [Roberts et al., 1996]. In practice, choosing $0 < \gamma < 1$ ($\gamma = 0.9$ is a common default) helps avoid random walk behavior and accelerates mixing [Neal et al., 2011, Chen et al., 2014]. Importantly, we retain the formalism of doing importance sampling on an extended space despite the loss of detailed balance. To show this, we can define the forward and (unnormalized) backward distributions as

$$q_{\text{fwd}}(\boldsymbol{\theta}_{0:K}, \mathbf{v}_{0:K}) = p_0(\boldsymbol{\theta}_0)\pi(\mathbf{v}_0)\mathcal{T}_1(\boldsymbol{\theta}_1, \mathbf{v}_1|\boldsymbol{\theta}_0, \mathbf{v}_0)\cdots\mathcal{T}_K(\boldsymbol{\theta}_K, \mathbf{v}_K|\boldsymbol{\theta}_{K-1}, \mathbf{v}_{K-1}) \tag{8}$$

$$q_{\text{bwd}}(\boldsymbol{\theta}_{0:K}, \mathbf{v}_{0:K}) = f_K(\boldsymbol{\theta}_K)\pi(\mathbf{v}_K)\tilde{\mathcal{T}}_K(\boldsymbol{\theta}_{K-1}, \mathbf{v}_{K-1}|\boldsymbol{\theta}_K, \mathbf{v}_K)\cdots\tilde{\mathcal{T}}_1(\boldsymbol{\theta}_0, \mathbf{v}_0|\boldsymbol{\theta}_1, \mathbf{v}_1) \tag{9}$$

where the transition operator $\mathcal{T}_k(\boldsymbol{\theta}_k, \mathbf{v}_k|\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}) = \mathcal{T}'_k(\boldsymbol{\theta}_k, \hat{\mathbf{v}}_k|\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1})\mathcal{T}''_k(\mathbf{v}_k|\hat{\mathbf{v}}_k)$ is the composition of a leapfrog step and momentum refreshment step. We define the reverse chain by starting with an exact sample and executing each of the above steps of Algorithm 1 in the reverse order, which leads to a surprisingly simple expression for our estimator. In particular, the backward transition operator is defined by $\tilde{\mathcal{T}}_k(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}|\boldsymbol{\theta}_k, \mathbf{v}_k) = \mathcal{T}''_k(\hat{\mathbf{v}}_k|\mathbf{v}_k)\mathcal{T}'_k(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}|\boldsymbol{\theta}_k, -\hat{\mathbf{v}}_k)$. Note that we need to flip the sign of $\hat{\mathbf{v}}_k$ in the reverse chain to account for time reversal. As a consequence of the above definitions, we have

$$\mathcal{T}''_k(\mathbf{v}_k|\hat{\mathbf{v}}_k) = \mathcal{T}''_k(\hat{\mathbf{v}}_k|\mathbf{v}_k)\pi(\mathbf{v}_k)/\pi(\hat{\mathbf{v}}_k). \tag{10}$$

This is because $\mathcal{T}''_k(\mathbf{v}_k|\hat{\mathbf{v}}_k) = \mathcal{N}(\gamma\hat{\mathbf{v}}_k, (1-\gamma^2)\mathbf{M})$ and $\mathcal{T}''_k(\hat{\mathbf{v}}_k|\mathbf{v}_k) = \mathcal{N}(\gamma\mathbf{v}_k, (1-\gamma^2)\mathbf{M})$. Furthermore, since $\mathcal{T}'_k$ is a deterministic leapfrog update, it is reversible and volume preserving, so we have $\mathcal{T}'_k(\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1}|\boldsymbol{\theta}_k, -\hat{\mathbf{v}}_k) = \mathcal{T}'_k(\boldsymbol{\theta}_k, \hat{\mathbf{v}}_k|\boldsymbol{\theta}_{k-1}, \mathbf{v}_{k-1})$. With this, we can derive the DAIS bound:

$$\begin{aligned}\mathcal{L}_{\text{DAIS}} &= \mathbb{E}_{q_{\text{fwd}}}\left[\log q_{\text{bwd}}(\boldsymbol{\theta}_{0:K}, \mathbf{v}_{0:K}) - \log q_{\text{fwd}}(\boldsymbol{\theta}_{0:K}, \mathbf{v}_{0:K})\right] \\ &= \mathbb{E}_{q_{\text{fwd}}}\left[\log p(\mathcal{D}, \boldsymbol{\theta}_K|\mathcal{M}) - \log p_0(\boldsymbol{\theta}_0) + \sum_{k=1}^{K}\log\frac{\pi(\hat{\mathbf{v}}_k)}{\pi(\mathbf{v}_{k-1})}\right]\end{aligned} \tag{11}$$

We remark that this bound supports the computation of pathwise derivatives. Intuitively, $\log\pi(\mathbf{v}_k)$ is the negative kinetic energy (plus a constant), hence each summand in the last term of (11) is measuring whether kinetic energy increases or decreases over the course of a leapfrog step. Leapfrog steps are energy preserving up to discretization error, which can be eliminated with the MH step. With the MH step, this term also measures the change of potential energy. Therefore, one can conclude that the AIS bound (6) is a special case of the DAIS bound when detailed balance holds. In particular, detailed balance implies energy conservation (see Neal et al. [2011]), i.e. that $f_k(\boldsymbol{\theta}_{k-1})\pi(\mathbf{v}_{k-1}) = f_k(\boldsymbol{\theta}_k)\pi(\hat{\mathbf{v}}_k)$, with which we recover (6).

3.1 Differentiable Annealed Variational Inference

DAIS can be applied to variational inference for a tighter bound; we name this differentiable annealed variational inference (DAVI). We note that the general idea of incorporating auxiliary MCMC states into a variational approximation was discussed in Salimans et al. [2015], but their formulation requires the specification and learning of a reverse transition model, whereas ours does not.

Recall that we can lower bound the log ML by choosing a tractable variational distribution and optimizing the bound. This has been widely adopted in variational autoencoders [Kingma and Welling, 2013, Rezende et al., 2014] and Bayesian neural networks [Blundell et al., 2015, Zhang et al., 2018]. The lower bound has the following form:

$$\mathcal{L} \equiv \mathbb{E}_{q_\phi}\left[\log p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M}) - \log q_\phi(\boldsymbol{\theta})\right] \tag{12}$$

However, the lower bound can be quite loose if the variational posterior family $q_\phi(\boldsymbol{\theta})$ is restrictive, e.g. Gaussian. To improve the bound, we can define a new variational distribution on an extended space as in (8), but starting from q_ϕ rather than p_0:

$$q_{\text{fwd}}(\boldsymbol{\theta}_{0:K}, \mathbf{v}_{0:K}) = q_\phi(\boldsymbol{\theta}_0)\pi(\mathbf{v}_0)\mathcal{T}_1(\boldsymbol{\theta}_1, \mathbf{v}_1|\boldsymbol{\theta}_0, \mathbf{v}_0)\cdots\mathcal{T}_K(\boldsymbol{\theta}_K, \mathbf{v}_K|\boldsymbol{\theta}_{K-1}, \mathbf{v}_{K-1}). \tag{13}$$

We also define associated intermediate distributions $p_k(\boldsymbol{\theta}) = q_\phi(\boldsymbol{\theta})^{1-\beta_k}p(\mathcal{D}, \boldsymbol{\theta}|\mathcal{M})^{\beta_k}$. This gives a new lower bound:

$$\mathcal{L}_{\text{DAVI}} \equiv \mathbb{E}_{q_{\text{fwd}}}\left[\log p(\mathcal{D}, \boldsymbol{\theta}_K|\mathcal{M}) - \log q_\phi(\boldsymbol{\theta}_0) + \sum_{k=1}^{K}\log\frac{\pi(\hat{\mathbf{v}}_k)}{\pi(\mathbf{v}_{k-1})}\right]. \tag{14}$$

We can maximize this lower bound over model parameters of $\mathcal{M}$, all parameters of AIS (e.g., annealing schedule β_k) as well as variational parameters ϕ.

3.2 Memory-Efficient Implementation

Naively optimizing instantiations of (11) or (14) w.r.t. parameters using reverse-mode differentiation involves storing the entire sequence of sampled states $\boldsymbol{\theta}_0, \mathbf{v}_0, \ldots, \boldsymbol{\theta}_K, \mathbf{v}_K$. This can be problematic in cases when K is large due to the large memory overhead. However, DAIS is compatible with the idea of reversible learning [Maclaurin et al., 2015], which ameliorates this problem. Instead of storing the states in memory, we can compute the previous state given the current state by reversing the dynamics. Recall that each DAIS transition is deterministic and reversible other than the use of noise $\boldsymbol{\varepsilon}_k$ for momentum refreshment. The exact noise samples can also be computed in reverse if one uses a deterministic and reversible scheme (e.g. the linear congruential generator) for managing pseudorandom number generator seeds. Assuming exact arithmetic, this means that the memory footprint of DAIS can be made constant with respect to K. Similar memory-efficiency tricks have also been used in other applications [Li et al., 2020, Ruan et al., 2021].

Table 1: Memory and time usage of DAIS implementations. B is 32 for single-precision floating-point format.

Scheme	Precision	Memory	Time
Naive	finite	$\mathcal{O}(BK)$	$\mathcal{O}(K)$
Rev. Learning	exact	$\mathcal{O}(1)$	$\mathcal{O}(K)$
Rev. Learning	finite	$\mathcal{O}(\log_2(1/\gamma)K)$	$\mathcal{O}(K)$

However, as discussed by Maclaurin et al. [2015], reversible learning with finite arithmetic precision requires some storage to counteract compounding round-off error. For $\gamma \neq 0$ ($\gamma = 0.9$ is a common default), we need on average $\log_2(1/\gamma)$ bits per parameter per step, which is still small compared to naive storage. We defer further exposition on memory-efficient DAIS to Appendix D. We remark that reversible learning is a potentially crucial property of DAIS as it affords some degree of scaling to longer chain lengths and, indirectly, bigger models.

4 Convergence Analysis for Bayesian Linear Regression

Neal [2001] and others have pointed that AIS is consistent, i.e. that it converges to the true value in log space in the limit of infinite intermediate distributions. However, these existing consistency results largely depend on the assumption that each transition $\mathcal{T}_k$ generates a state from p_k, independent of the previous state. This is unrealistic in practice.

Here, we analyze DAIS without the assumption of perfect transitions. In particular, we focus on the Bayesian linear regression setting and adopt the following model:

$$\begin{aligned}
\text{prior: } \boldsymbol{\theta} &\sim \mathcal{N}(\boldsymbol{\mu}_p, \boldsymbol{\Lambda}_p^{-1}) \\
\text{likelihood: } \boldsymbol{\theta} &\sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Lambda}_{\text{lld}}^{-1}) \text{ where } \boldsymbol{\Lambda}_{\text{lld}} = \sigma^{-2}\mathbf{X}^\top\mathbf{X} \text{ and } \boldsymbol{\mu}_* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\
\text{posterior: } \boldsymbol{\theta} &\sim \mathcal{N}(\boldsymbol{\mu}_{\text{pos}}, \boldsymbol{\Lambda}_{\text{pos}}^{-1}) \text{ where } \boldsymbol{\mu}_{\text{pos}} = \boldsymbol{\Lambda}_{\text{pos}}^{-1}(\boldsymbol{\Lambda}_p\boldsymbol{\mu}_p + \boldsymbol{\Lambda}_{\text{lld}}\boldsymbol{\mu}_*) \text{ and } \boldsymbol{\Lambda}_{\text{pos}} = \boldsymbol{\Lambda}_p + \boldsymbol{\Lambda}_{\text{lld}}
\end{aligned}$$

with $\mathbf{X} \in \mathbb{R}^{n \times d}$ denoting the input features and $\mathbf{y} \in \mathbb{R}^{n \times 1}$ the targets. We choose Bayesian linear regression because it enables us to analyze the dynamics analytically in a similar manner as done by the noisy quadratic model (NQM) [Zhang et al., 2019] in the context of optimization. We adopt the leapfrog step (we assume an identity mass matrix without loss of generality because we can absorb $\mathbf{M}$ into the input matrix $\mathbf{X}$ in Algorithm 1) and obtain the following update rule (see Appendix B.1 for derivation):

$$\begin{aligned}
\boldsymbol{\theta}_k &\leftarrow \left(\mathbf{I} - \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\boldsymbol{\theta}_{k-1} + \left(\eta_k\mathbf{I} - \frac{\eta_k^3}{4}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\mathbf{v}_{k-1} + \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\mu}_{\text{pos}}^{\beta_k} \\
\hat{\mathbf{v}}_k &\leftarrow -\eta_k\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\theta}_{k-1} + \left(\mathbf{I} - \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\mathbf{v}_{k-1} + \eta_k\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\mu}_{\text{pos}}^{\beta_k}
\end{aligned} \tag{15}$$

where $\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k} = \boldsymbol{\Lambda}_p + \beta_k\boldsymbol{\Lambda}_{\text{lld}}$ and $\boldsymbol{\mu}_{\text{pos}}^{\beta_k} = (\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k})^{-1}(\boldsymbol{\Lambda}_p\boldsymbol{\mu}_p + \beta_k\boldsymbol{\Lambda}_{\text{lld}}\boldsymbol{\mu}_*)$. With these iterative updates, we can compute the expectation and covariance of $\boldsymbol{\theta}_k$ and $\mathbf{v}_k$ at any time k, which suffices to compute the lower bound in closed-form.

4.1 Sublinear Convergence in the Full-Batch Setting

With the model defined, we now show that our algorithm is asymptotically consistent, i.e., the bound (11) converges to exact log ML as K goes to infinity. For Bayesian linear regression, the

update rules in (15) are affine transformations of Gaussian random variables, so the distribution of $\boldsymbol{\theta}_k$ is also Gaussian in the form of $\mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. We can compute the gap between the log ML and our lower bound in closed-form (see Appendix B.2 for derivation):

$$\log p(\mathcal{D}) - \mathcal{L}_{\text{DAIS}} = \underbrace{\frac{1}{2}\|\boldsymbol{\mu}_K - \boldsymbol{\mu}_{\text{pos}}\|^2_{\boldsymbol{\Lambda}_{\text{pos}}}}_{①} + \underbrace{\frac{1}{2}\text{Tr}(\boldsymbol{\Lambda}_{\text{pos}}\boldsymbol{\Sigma}_K) - \frac{d}{2}}_{②} + \underbrace{\frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_{\text{pos}}|}{|\boldsymbol{\Sigma}_p|} - \mathbb{E}_q\left[\sum_{k=1}^{K}\log\frac{\pi(\hat{\mathbf{v}}_k)}{\pi(\mathbf{v}_{k-1})}\right]}_{③} \tag{16}$$

where d is the feature dimension. Here, ① and ② measure the error of last-iterate Markov chain convergence and will both vanish as long as $\boldsymbol{\mu}_K \to \boldsymbol{\mu}_{\text{pos}}$ and $\boldsymbol{\Sigma}_K \to \boldsymbol{\Sigma}_{\text{pos}}$. We will show later that they converge with a rate of $\mathcal{O}(\frac{1}{\eta^2 K})$. The key is to show that $\boldsymbol{\mu}_k$ (resp. $\boldsymbol{\Lambda}_k$) lags behind $\boldsymbol{\mu}_{\text{pos}}^{\beta_k}$ (resp. $\boldsymbol{\Lambda}_{pos}^{\beta_k}$) with roughly $\frac{1}{\eta^2}$ steps. Formally, we have the following.

Lemma 1. *Given equally spaced β_k, running DAIS with $\gamma = 0$ and $\eta \sim \frac{1}{K^c}$ where $c \geq \frac{1}{4}$ yields*

$$\|\boldsymbol{\mu}_{k-1} - \boldsymbol{\mu}_{pos}^{\beta_k}\|_2 = \mathcal{O}(K^{2c-1}),\ \|\boldsymbol{\Lambda}_{k-1} - \boldsymbol{\Lambda}_{pos}^{\beta_k}\|_2 = \mathcal{O}(K^{2c-1}). \tag{17}$$

We remark that the assumption of β_k being equally spaced is not essential and can be relaxed as long as they are chosen by a scheme that leads to $\beta_k - \beta_{k-1}$ going down approximately in inverse proportion to K. In addition, we note that the assumption of full momentum refreshment is for convenience and we believe a similar result holds for $\gamma > 0$.

Importantly, this lemma implies that both ① and ② vanish sublinearly if we choose $c < \frac{1}{2}$. The analysis of error term ③ is more nuanced. In particular, this error could either come from using transitions for each of these intermediate distributions that do not bring the distribution close to equilibrium, or from using a finite number of distributions to anneal from p_0 to p_K. Surprisingly, the error ③ decays as fast as the other two terms if the step size scales as $1/K^c$ with $c \geq \frac{1}{4}$. In summary, we have the following theorem.

Theorem 1. *Given equally spaced β_k, running DAIS with $\gamma = 0$ and $\eta \sim \frac{1}{K^c}$ where $c \geq \frac{1}{4}$ yields*

$$\log p(\mathcal{D}) - \mathcal{L}_{DAIS} = \mathcal{O}(K^{2c-1}).$$

With $c = \frac{1}{4}$, we have the optimal convergence rate $\mathcal{O}(1/\sqrt{K})$.

We remark that with perfect transitions, the requirement of $c \geq 1/4$ is not necessary and we can achieve $\mathcal{O}(1/K)$ convergence, as also shown in [Grosse et al., 2013]. The gap between $\mathcal{O}(1/\sqrt{K})$ and $\mathcal{O}(1/K)$ highlights the importance of considering convergence to the stationary distribution.

4.2 Inconsistency in the Stochastic Setting

We have shown that our algorithm is asymptotically consistent in the full-batch setting. Often, a consistent/convergent algorithm in the deterministic setting readily implies a similar convergence result in the stochastic setting. For example, SGD [Robbins and Monro, 1951] and SGMCMC [Chen et al., 2014, Ma et al., 2015] are both convergent in the presence of noise. This begs the question of whether DAIS is consistent when we only have access to stochastic gradients. Here, we adopt a additive noise model[3] $\tilde{\nabla}\log f_k(\boldsymbol{\theta}) = \nabla \log f_k(\boldsymbol{\theta}) + \boldsymbol{\varepsilon}$. This model is commonly used in the stochastic approximation literature, and such a model has also been adopted in Welling and Teh [2011]. With such a noise model, we have the following dynamics:

$$\begin{aligned}
\boldsymbol{\theta}_k &\leftarrow \left(\mathbf{I} - \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\boldsymbol{\theta}_{k-1} + \left(\eta_k\mathbf{I} - \frac{\eta_k^3}{4}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\mathbf{v}_{k-1} + \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\mu}_{\text{pos}}^{\beta_k} + \frac{\eta_k^2}{2}\boldsymbol{\varepsilon} \\
\hat{\mathbf{v}}_k &\leftarrow -\eta_k\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\theta}_{k-1} + \left(\mathbf{I} - \frac{\eta_k^2}{2}\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)\mathbf{v}_{k-1} + \eta_k\boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\boldsymbol{\mu}_{\text{pos}}^{\beta_t} + \eta_k\boldsymbol{\varepsilon}
\end{aligned} \tag{18}$$

where stochastic noise $\boldsymbol{\varepsilon}$ has variance lowered bounded by $\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}$. Further, we let $\boldsymbol{\mu}_k^{\mathbf{v}} = \mathbb{E}[\hat{\mathbf{v}}_k]$ and $\boldsymbol{\Sigma}_k^{\mathbf{v}}$ be the covariance of $\hat{\mathbf{v}}_k$. Surprisingly, we find that DAIS is incompatible with stochastic gradients, as summarized in the following theorem:

[3] In reality, the noise consists of two parts: multiplicative input subsampling noise and additive label noise. We can assume that the noise is lower bounded by the additive part. See Appendix B.4 for justifications.

Theorem 2. *For stochastic DAIS with full momentum refreshment ($\gamma = 0$ in Algorithm 1) and any stepsize scheme, we have*

$$\liminf_{K\to\infty} |\log p(\mathcal{D}) - \mathcal{L}_{DAIS}| > 0. \tag{19}$$

Here, we give some intuition why DAIS fails in the stochastic setting. To ensure convergence of $\boldsymbol{\theta}_K$ to $\mathcal{N}(\boldsymbol{\mu}_{\text{pos}}, \boldsymbol{\Sigma}_{\text{pos}})$, a major requirement is for the step sizes to satisfy the property [Robbins and Monro, 1951] of $\lim_{K\to\infty} \sum_{k=1}^{K} \eta_k^2 = \infty$. However, the randomness of mini-batching sampling would contribute to the variance of $\hat{\mathbf{v}}_k$. In particular, we have the following recursion:

$$\tilde{\boldsymbol{\Sigma}}_k^{\mathbf{v}} = \eta_k^2 \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k} \hat{\boldsymbol{\Sigma}}_{k-1} \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k} + \left(\mathbf{I} - \frac{\eta_k^2}{2} \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k}\right)^2 + \eta_k^2 \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}. \tag{20}$$

For notational convenience, we let $\hat{\boldsymbol{\Sigma}}_k^{\mathbf{v}} \triangleq \eta_k^2 \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k} \hat{\boldsymbol{\Sigma}}_{k-1} \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k} + (\mathbf{I} - \frac{\eta_k^2}{2} \boldsymbol{\Lambda}_{\text{pos}}^{\beta_k})^2$. Here, we used $\tilde{\boldsymbol{\Sigma}}_k^{\mathbf{v}}$, $\hat{\boldsymbol{\Sigma}}_k^{\mathbf{v}}$ and $\hat{\boldsymbol{\Sigma}}_k$ to avoid confusion with $\boldsymbol{\Sigma}_k^{\mathbf{v}}$ and $\boldsymbol{\Sigma}_k$ in the full-batch setting. In this case, if we follow Stephan et al. [2017] and Chen et al. [2014] in assuming $\boldsymbol{\varepsilon}$ is Gaussian,[4] we have

$$\mathbb{E}_q \left[\sum_{k=1}^{K} \log \frac{\pi(\hat{\mathbf{v}}_k)}{\pi(\mathbf{v}_{k-1})} \right] = \sum_{k=1}^{K} \left[-\frac{1}{2} \|\boldsymbol{\mu}_k^{\mathbf{v}}\|_2^2 - \frac{1}{2} \mathrm{Tr}(\hat{\boldsymbol{\Sigma}}_k^{\mathbf{v}}) + \frac{d}{2} \right] - \sum_{k=1}^{K} \left[\frac{1}{2} \eta_k^2 \mathrm{Tr}(\boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}}) \right]. \tag{21}$$

The second term of (21) goes to infinity as $\lim_{K\to\infty} \sum_{k=1}^{K} \eta_k^2 = \infty$. Intuitively, the gradient noise adds to the kinetic energy, and the size of this contribution is proportional to η_k^2. Since this effect is cumulative over all K steps, η_k has to be reduced at least as $1/\sqrt{K}$ for the kinetic energy term to go down. However, this contradicts the requirement of last-iterate convergence.

One may wonder why gradient noise does not hurt the convergence of SGLD [Welling and Teh, 2011] or SGMCMC [Ma et al., 2015]. Generally speaking, these algorithms are only concerned with the last iteration convergence to the true posterior, hence one can eliminate stochastic error by taking more steps of a smaller size. In contrast, our bound (and potentially other AIS-style algorithms) relies on all intermediate distributions, and so the error induced by stochastic gradient noise accumulates over the whole trajectory.

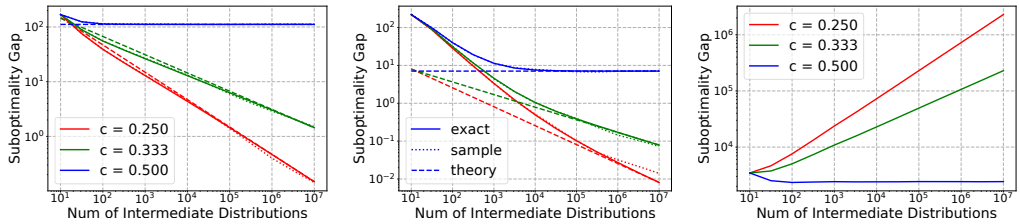
5 Related Works

For marginal likelihood estimation (or partition function estimation), Sequential Monte Carlo (SMC) [Doucet et al., 2001, Del Moral et al., 2006] is another popular method which is derived from partial filtering method. While SMC is based on a different intuition from AIS, the underlying mathematics is equivalent. In SMC, the intermediate distributions are defined by conditioning on a sequence of increasing subsets of data. Therefore, we expect our analysis in Section 4 would also apply to SMC.

Besides, both AIS and SMC are closely related to a broader family of techniques for partition function estimation, all based on the following identity from statistical physics: $\log \mathcal{Z}_K - \log \mathcal{Z}_0 = \int_0^1 \mathbb{E}_{\boldsymbol{\theta}\sim p_\beta} \left[\frac{d}{d\beta} \log f_\beta(\boldsymbol{\theta}) \right] d\beta$. In particular, the weight update in AIS can be seen as a finite difference approximation. In comparison, thermodynamic integration (TI) [Frenkel and Smit, 2001] estimates this integration using numerical quadrature, and path sampling [Gelman and Meng, 1998] does so with Monte Carlo integration. Recently, Masrani et al. [2019] connected TI and variational inference for a tighter bound on the log ML.

In the context of variational inference, many papers have also investigated tighter lower bounds for the log ML. Burda et al. [2016] proposed a strictly tighter log-likelihood lower bound derived from importance weighting. Salimans et al. [2015] proposed to incorporate MCMC iterations into the variational approximation. However, the proposed methods require learning reverse kernels which have a large impact on performance. The same authors also briefly discussed annealed variational inference, which combines variational inference and AIS. However, their derivation relies on the detailed balance assumption and is therefore not amenable to gradient-based optimization. Later, Caterini et al. [2019] proposed the Hamiltonian VAE, which improves Hamiltonian variational inference [Salimans et al., 2015] with an optimally chosen reverse MCMC kernel. In particular, they

[4] This assumption is non-essential, though Gaussian noise is reasonable by the central limit theorem.



(a) Full refreshment **(b)** Partial refreshment **(c)** Mini-batch gradient

Figure 1: Gap between true log ML and our DAIS bound as a function of number of intermediate distributions. Solid lines are exact computation of our DAIS bound; dotted lines are sample-based simulation (Monte Carlo method with 100 samples); dashed lines are theoretical predictions based on Theorem 1 with slope $2c - 1$. For the rightmost figure, we use a batch size of 100.

removed the momentum sampling step and used deterministic forward transitions. The resulting algorithm can be thought of as a normalizing flow scheme in which the flow depends explicitly on the target distribution. Along this line, Le et al. [2018], Naesseth et al. [2018], Maddison et al. [2017] proposed to meld variational inference and SMC for time-series models.

Finally, stochastic gradient variants of several MCMC algorithms [Welling and Teh, 2011, Chen et al., 2014, Ma et al., 2015] have been proposed over the last decade. In particular, they showed that adding the "right amount" of noise to the parameter updates leads to samples from the target posterior as long as the step size is annealed. Importantly, the convergence rates of these algorithms are established in both the full-batch setting [Dalalyan, 2017, Cheng et al., 2018] and the stochastic setting [Chen et al., 2015, Teh et al., 2016, Raginsky et al., 2017, Zou et al., 2020]. By contrast, the convergence properties for AIS and related algorithms were largely unknown even for the deterministic case, and it remains largely unexplored whether AIS can be made compatible with stochastic gradients.

6 Experiments

In this section, we discuss the experiments used to validate our algorithm and theory. Importantly, we do *not* aim to achieve state-of-the-art on these tasks.

6.1 Bayesian Linear Regression

In Section 4, we proved for the Bayesian linear regression setting that while DAIS is asymptotically consistent with full-batch gradient, the noise injected into the system via stochastic gradients precludes convergence. Here, we verify our theory with numerical simulations. The n input vectors $\mathbf{X} \in \mathbb{R}^{n \times d}$ and targets $\mathbf{y} \in \mathbb{R}^n$ respectively consist of entries sampled from $\mathcal{N}(0, 0.01)$ and $\mathcal{N}(0, 1)$. In particular, we choose $n = 10,000$ and $d = 10$ for our simulations (the results are qualitatively same with different n and d). In addition, we set the observation variance $\sigma^2 = 1$. For convenience, we set the linear annealing scheme $\beta_k = \frac{k}{K}$.

In Figure 1, we report the gap between exact log ML and our bound as a function of number of intermediate distributions. With full-batch gradients, the simulations (solid and dotted lines) align well with our theoretical predictions (dashed lines) for different step-size scaling schemes, suggesting our bound in Theorem 1 is tight. In addition, we observe in Figure 1c that the gap fails to vanish with mini-batch gradients for all step-size scaling schemes. Interestingly, with $c = 1/2$, the gap stays constant. This matches our predictions that the deterministic error decays as $\mathcal{O}(K^{2c-1}) = \mathcal{O}(1)$ while the stochastic error is proportional to $\sum_{k=1}^{K} \eta_k^2 = \mathcal{O}(K^{2c-1}) = \mathcal{O}(1)$.

6.2 Variational Autoencoder

We compare the performance of DAVI to vanilla VAE [Kingma and Welling, 2013] and IWAE [Burda et al., 2016] on density modeling tasks. We use the dynamically binarized MNIST [LeCun et al., 1998] dataset. We use the same architecture as in IWAE paper. The prior $p(\mathbf{z})$ is a 50-dimensional standard Gaussian distribution. The conditional distributions $p(\mathbf{x}_i|\mathbf{z})$ are independent Bernoulli, with the decoder parameterized by two hidden layers, each with 200 tanh units. The variational posterior $q(\mathbf{z}|\mathbf{x})$ is also a 50-dimensional Gaussian with diagonal covariance, whose mean and variance are both parameterized by two hidden layers with 200 tanh units (see other details in Appendix C.1).

Table 2: Test negative log-likelihood of the trained model, estimated using AIS with 10,000 intermediate distribution and 10 particles. For VAE/IWAE, we used $S \times K$ samples. The numbers reported are averaged over three runs. The standard deviations are fairly small over three runs (< 0.06).

Objective	$S \times K = 1$	$S \times K = 5$	$S \times K = 10$		$S \times K = 50$			$S \times K = 500$	
	$K = 1$	$K = 5$	$K = 5$	$K = 10$	$K = 5$	$K = 10$	$K = 50$	$K = 10$	$K = 50$
VAE	86.93	86.95	86.91			86.94		86.89	
IWAE	86.93	**85.43**	**85.09**			84.46		83.87	
DAVI	-	86.51	85.45	85.76	84.49	84.45	85.23	83.62	83.65
DAVI (adapt)	-	86.49	85.35	85.73	84.42	**84.39**	85.00	**83.56**	83.69

In the first set of experiments, we investigate the effect of number of intermediate distributions K and combine it with importance sampling (as done in IWAE) with S samples in DAVI. To be specific, we define the bound as follows:

$$\log \frac{1}{S} \sum_{i=1}^{S} \left(\frac{p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}_K^i)}{q_\phi(\mathbf{z}_0^i|\mathbf{x})} \prod_{k=1}^{K} \frac{\pi(\hat{\mathbf{v}}_k^i)}{\pi(\mathbf{v}_{k-1}^i)} \right), \tag{22}$$

where we sample $(\mathbf{z}_0^i, \mathbf{v}_0^i, \hat{\mathbf{v}}_1^i, \dots)$ independently from q_{fwd}. By default, we use partial momentum refreshment with $\gamma = 0.9$ and equally spaced annealing parameters $\beta_k = k/K$. We compare it to vanilla VAE and IWAE bounds with $S \times K$ samples. As shown in Table 2, increasing K gives strictly better models with lower test negative log-likelihood. However, IWAE achieves slightly better performance with roughly the same computation if $S \times K$ is small. On the other hand, DAVI is more effective with more compute budget (i.e., $S \times K$ is large) and eventually outperforms IWAE.

In the second set of experiments, we learn the annealing scheme of DAVI together with the parameters of encoder and decoder. Comparing the third and fourth rows of Table 2, one can see that learning the annealing scheme improves the performance slightly.

Lastly, we also compare our algorithm with IWAE, AIS, and Hamiltonian AIS (HAIS) [Sohl-Dickstein and Culpepper, 2012] in evaluating the log-likelihood of trained models. To be noted, IWAE and AIS have been widely used in evaluating VAE, see e.g. Wu et al. [2016], Huang et al. [2020]. For HAIS and DAIS, we employ the optimal step-size scaling scheme derived in Theorem 1 with $c = 1/4$ and only tune the step size for the case of $K = 10$. For all implementation details, please see Appendix C.2. In particular, we choose the vanilla VAE model to compare all different algorithms (see results of other models in Appendix C.3). In Figure 2, we report the estimated negative log-likelihood as a function of the number of particles (for IWAE) or gradient updates (for AIS, HAIS and DAIS). Interestingly, we observe that IWAE performs better when we have limited computation and AIS/HAIS/DAIS win out if we increase K. Moreover, DAIS performs on par with AIS/HAIS but without requiring the MH correction steps.

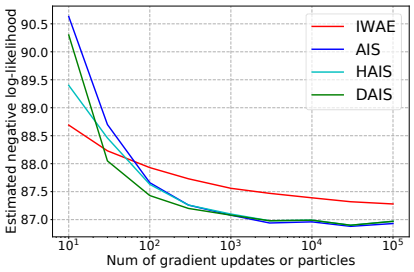


Figure 2: Results of different algorithms in evaluating a trained VAE model.

7 Conclusion

In this paper, we proposed a differentiable AIS (DAIS) algorithm for marginal likelihood estimation. We provided a detailed convergence analysis for Bayesian linear regression which goes beyond existing analyses. Using this analysis, we proved a sublinear convergence rate of DAIS in the full-batch setting. However, we showed that DAIS is inconsistent when mini-batch gradients are used due to a fundamental incompatibility between the goals of last-iterate convergence to the posterior and elimination of the pathwise stochastic error. This comprises an interesting counterexample to the general trend of algorithms consistent in the deterministic setting remaining consistent in the stochastic setting. Our negative result helps explain the difficulty of developing practically effective AIS-like algorithms that exploit mini-batch gradients. Our numerical experiments validate our claims.

Social Impact

The main contribution in this work is methodological and theoretical. We do not expect there to be direct negative societal impacts from our work.

References

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International Conference on Machine Learning*, pages 1613–1622. PMLR, 2015.

Yuri Burda, Roger B Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *ICLR (Poster)*, 2016.

AL Caterini, A Doucet, and D Sejdinovic. Hamiltonian variational auto-encoder. *Advances in Neural Information Processing Systems*, 31, 2019.

Changyou Chen, Nan Ding, and Lawrence Carin. On the convergence of stochastic gradient mcmc algorithms with high-order integrators. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2278–2286, 2015.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. In *International conference on machine learning*, pages 1683–1691. PMLR, 2014.

Xiang Cheng, Niladri S Chatterji, Peter L Bartlett, and Michael I Jordan. Underdamped langevin mcmc: A non-asymptotic analysis. In *Conference on Learning Theory*, pages 300–323. PMLR, 2018.

Arnak S Dalalyan. Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(3): 651–676, 2017.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3):411–436, 2006.

Arnaud Doucet, Nando De Freitas, and Neil Gordon. An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pages 3–14. Springer, 2001.

Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Elsevier, 2001.

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

Roger B Grosse, Chris J Maddison, and Ruslan Salakhutdinov. Annealing between distributions by averaging moments. In *NIPS*, pages 2769–2777. Citeseer, 2013.

Roger B Grosse, Zoubin Ghahramani, and Ryan P Adams. Sandwiching the marginal likelihood using bidirectional monte carlo. *arXiv preprint arXiv:1511.02543*, 2015.

Alan M Horowitz. A generalized guided monte carlo algorithm. *Physics Letters B*, 268(2):247–252, 1991.

Sicong Huang, Alireza Makhzani, Yanshuai Cao, and Roger Grosse. Evaluating lossy compression rates of deep generative models. In *International Conference on Machine Learning*, pages 4444–4454. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Tuan Anh Le, Maximilian Igl, Tom Rainforth, Tom Jin, and Frank Wood. Auto-encoding sequential monte carlo. In *International Conference on Learning Representations*, 2018.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in Neural Information Processing Systems*, 28:2917–2925, 2015.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International conference on machine learning*, pages 2113–2122. PMLR, 2015.

Chris J Maddison, Dieterich Lawson, George Tucker, Nicolas Heess, Mohammad Norouzi, Andriy Mnih, Arnaud Doucet, and Yee Whye Teh. Filtering variational objectives. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6576–6586, 2017.

Vaden Masrani, Tuan Anh Le, and Frank Wood. The thermodynamic variational objective. *arXiv preprint arXiv:1907.00031*, 2019.

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

Christian Naesseth, Scott Linderman, Rajesh Ranganath, and David Blei. Variational sequential monte carlo. In *International Conference on Artificial Intelligence and Statistics*, pages 968–977. PMLR, 2018.

Radford M Neal. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Maxim Raginsky, Alexander Rakhlin, and Matus Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In *Conference on Learning Theory*, pages 1674–1703. PMLR, 2017.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003.

Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. *Advances in neural information processing systems*, pages 294–300, 2001.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

Gareth O Roberts, Richard L Tweedie, et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Yangjun Ruan, Karen Ullrich, Daniel Severo, James Townsend, Ashish Khisti, Arnaud Doucet, Alireza Makhzani, and Chris J Maddison. Improving lossless compression rates via monte carlo bits-back coding. *arXiv preprint arXiv:2102.11086*, 2021.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain monte carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226, 2015.

John Skilling et al. Nested sampling for general bayesian computation. *Bayesian analysis*, 1(4): 833–859, 2006.

Jascha Sohl-Dickstein and Benjamin J Culpepper. Hamiltonian annealed importance sampling for partition function estimation. *arXiv preprint arXiv:1205.1925*, 2012.

M Stephan, Matthew D Hoffman, David M Blei, et al. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

Yee Whye Teh, Alexandre H Thiery, and Sebastian J Vollmer. Consistency and fluctuations for stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 17, 2016.

Marc Teyssier and Daphne Koller. Ordering-based search: a simple and effective algorithm for learning bayesian networks. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 584–590, 2005.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5852–5861. PMLR, 2018.

Guodong Zhang, Lala Li, Zachary Nado, James Martens, Sushant Sachdeva, George E Dahl, Christopher J Shallue, and Roger Grosse. Which algorithmic choices matter at which batch sizes? insights from a noisy quadratic model. *arXiv preprint arXiv:1907.04164*, 2019.

Difan Zou, Pan Xu, and Quanquan Gu. Faster convergence of stochastic gradient langevin dynamics for non-log-concave sampling. *arXiv preprint arXiv:2010.09597*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discussed the limitations of DAIS in the stochastic setting.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In the appendices.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Will submit along with supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provided implementation details in the Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] For VAE experiments, the variations between runs are fairly small, we mentioned that in the main paper and only reported the mean value over different runs.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]