Hierarchical Semi-Implicit Variational Inference with Application to Diffusion Model Acceleration

Anonymous Author(s)
Affiliation
Address
email

Abstract

Semi-implicit variational inference (SIVI) has been introduced to expand the analytical variational families by defining expressive semi-implicit distributions in a hierarchical manner. However, the single-layer architecture commonly used in current SIVI methods can be insufficient when the target posterior has complicated structures. In this paper, we propose hierarchical semi-implicit variational inference, called HSIVI, which generalizes SIVI to allow more expressive multi-layer construction of semi-implicit distributions. By introducing auxiliary distributions that interpolate between a simple base distribution and the target distribution, the conditional layers can be trained by progressively matching these auxiliary distributions one layer after another. Moreover, given pre-trained score networks, HSIVI can be used to accelerate the sampling process of diffusion models with the score matching objective. We show that HSIVI significantly enhances the expressiveness of SIVI on several Bayesian inference problems with complicated target distributions. When used for diffusion model acceleration, we show that HSIVI can produce high quality samples comparable to or better than the existing fast diffusion model based samplers with a small number of function evaluations on various datasets.

1 Introduction

Variational inference (VI) is an approximate Bayesian inference method that is gaining in popularity, where one tries to find an approximation to the target posterior distribution using an optimization approach (Jordan et al., 1999; Wainwright & Jordan, 2008; Blei et al., 2016). To do that, it first posits a family of variational distributions and then seeks the closest member from this family that minimizes some statistical distance to the target posterior, usually the Kullback-Leibler (KL) divergence. As the posterior is not analytically available, an equivalent formulation is often adopted in practice where one maximizes the evidence lower bound (ELBO) instead (Jordan et al., 1999).

One classical VI method is mean-field VI, which assumes a factorizable structure of the variational distributions over the parameters or latent variables (Bishop & Tipping, 2000). This often leads to closed-form coordinate-ascent update rules when certain conditional conjugacy conditions are satisfied. In practice, the conditional conjugacy may not hold and the true posterior could be much more complicated than what a factorized variational distribution can accurately approximate. In recent years, several attempts have been made in VI that alleviate these constraints by designing more flexible variational families (Jaakkola & Jordan, 1998; Saul & Jordan, 1996; Giordano et al., 2015; Tran et al., 2015; Rezende & Mohamed, 2015; Dinh et al., 2017; Kingma et al., 2016; Papamakarios et al., 2019), together with generic training algorithms via Monte Carlo gradient estimators (Nott et al., 2012; Paisley et al., 2012; Ranganath et al., 2014; Rezende et al., 2014; Kingma & Welling, 2014). While successful, these approaches all assume tractable densities of variational distributions.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

To further expand the capacity of variational families, one approach is to incorporate the implicit models that have intractable densities but are easy to sample from (Huszár, 2017; Tran et al., 2017; Mescheder et al., 2017; Shi et al., 2018a,b; Song et al., 2019). However, as the densities are intractable for implicit models, one often resorts to density ratio estimation for ELBO evaluation during training, which is known to be difficult in high dimensional settings (Sugiyama et al., 2012). To avoid density ratio estimation, semi-implicit variational inference (SIVI) has been proposed where the variational distributions are formed through a semi-implicit hierarchical construction, and various training criteria have been employed (Yin & Zhou, 2018; Moens et al., 2021; Titsias & Ruiz, 2019; Yu & Zhang, 2023).

While striking a good balance between approximation flexibility and training efficiency, current SIVI methods often use a single conditional layer which can be insufficient when the target posterior possesses complicated structures (e.g., multimodality, see an example in Section 5.1). To enhance the expressiveness of single-layer models, an intuitive but effective approach is to extend them to multi-layer hierarchical models (Vahdat & Kautz, 2020; Ranganath et al., 2016; Sobolev & Vetrov, 2019). In this paper, we propose hierarchical semi-implicit variational inference (HSIVI), which is a generalization of SIVI that allows multiple conditional layers. Instead of training the hierarchical semi-implicit model end to end, we introduce auxiliary distributions that interpolate between a simple base distribution and the target distribution to guide the intermediate semi-implicit distributions toward the target distribution. The conditional layers are then trained sequentially to match these auxiliary bridging distributions given the fitted semi-implicit distributions from the previous layers (Figure 1), using different criteria from before. This way, HSIVI allows progressive learning of the target distribution that significantly reduces the burden of each conditional layer. Moreover, HSIVI with the score matching objective can also be used to accelerate the sampling process of diffusion models where the pre-trained score networks corresponding to different noise levels provide a natural sequence of bridging distributions. In experiments, we demonstrate the effectiveness of HSIVI on both Bayesian inference tasks with complicated target distributions and diffusion model acceleration.

2 Background on semi-implicit variational inference

The semi-implicit variational family (Yin & Zhou, 2018; Titsias & Ruiz, 2019) is defined as

$$q_\phi(\boldsymbol{x}) = \int q_\phi(\boldsymbol{x}|\boldsymbol{z})q(\boldsymbol{z})\mathrm{d}\boldsymbol{z}, \tag{1}$$

where ϕ are the variational parameters, $q_\phi(\boldsymbol{x}|\boldsymbol{z})$ is called the conditional layer, and $q(\boldsymbol{z})$ is called the mixing layer. This variational family is said to be semi-implicit as $q_\phi(\boldsymbol{x}|\boldsymbol{z})$ is required to be explicit and $q(\boldsymbol{z})$ is often implicit. The semi-implicit variational family is capable of capturing more complicated dependencies between variables (Yin & Zhou, 2018; Titsias & Ruiz, 2019; Yu & Zhang, 2023) than explicit variational families without the hierarchical structure. Given the observed data D, the classical VI methods often use the evidence lower bound (ELBO) for training, which is defined as $\text{ELBO} := \mathbb{E}_{q_\phi(\boldsymbol{x})}\left[\log p(D, \boldsymbol{x}) - \log q_\phi(\boldsymbol{x})\right]$. However, as $q_\phi(\boldsymbol{x})$ is no longer tractable in SIVI, alternative training objectives have been introduced.

ELBO related objectives Yin & Zhou (2018) considered a sequence of lower bounds of the ELBO

$$\mathcal{L}_{\text{SIVI-LB}}(p(\boldsymbol{x}|D)\|q_\phi(\boldsymbol{x})) := \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}),\boldsymbol{x}\sim q_\phi(\boldsymbol{x},\boldsymbol{z})}\mathbb{E}_{\{\boldsymbol{z}^{(i)}\}_{i=1}^K \overset{\text{i.i.d.}}{\sim} q(\boldsymbol{z})} \log \frac{p(D,\boldsymbol{x})}{\frac{1}{K+1}\left(q_\phi(\boldsymbol{x}|\boldsymbol{z}) + \sum_{k=1}^K q_\phi(\boldsymbol{x}|\boldsymbol{z}^{(k)})\right)}. \tag{2}$$

It is an asymptotically exact surrogate in the sense that $\lim_{K\to\infty}\mathcal{L}_{\text{SIVI-LB}} = \text{ELBO}$. Titsias & Ruiz (2019) proposed unbiased implicit variational inference (UIVI) which uses samples from the inverse conditional distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ (from an MCMC run, e.g. Hamiltonian Monte Carlo (Neal, 2011)) to provide an unbiased gradient estimator of the exact ELBO. See more details of UIVI in Appendix B.

Score matching objective Besides the ELBO, score based distance measures have also been used for variational inference where the score function $\boldsymbol{S}(\boldsymbol{x}) := \nabla_{\boldsymbol{x}}\log p(\boldsymbol{x}|D) = \nabla_{\boldsymbol{x}}\log p(D,\boldsymbol{x})$ is assumed to be tractable (Liu et al., 2016; Zhang et al., 2018; Hu et al., 2018). Yu & Zhang (2023) considered the following Fisher divergence between the target distribution and the semi-implicit variational distribution

$$\mathcal{D}_{\text{Fisher}}(p(\boldsymbol{x}|D)\|q_\phi(\boldsymbol{x})) := \mathbb{E}_{\boldsymbol{x}\sim q_\phi(\boldsymbol{x})}\|\boldsymbol{S}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}}\log q_\phi(\boldsymbol{x})\|_2^2. \tag{3}$$

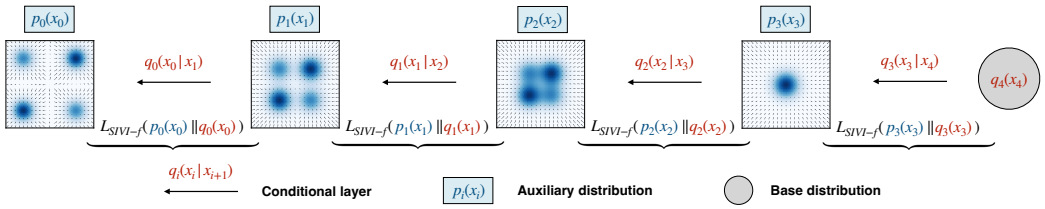


Figure 1: An example for 4-layer HSIVI. The target distribution $p_0(\boldsymbol{x})$ is a Gaussian mixture and the auxiliary distributions $\{p_i(\boldsymbol{x})\}_{i=0}^3$ are constructed using the diffusion bridge. The auxiliary distributions are plotted in the squares, where the blue heatmap describes the probability density and the arrows represent the score functions of the auxiliary distributions.

By reformulating $\mathcal{D}_{\text{Fisher}}$ as the maximum of the following optimization problem

$$\mathcal{D}_{\text{Fisher}}(p(\boldsymbol{x}|D)\|q_\phi(\boldsymbol{x})) = \max_{\boldsymbol{f}(\boldsymbol{x})} \left[2\boldsymbol{f}(\boldsymbol{x})^T(\boldsymbol{S}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q_\phi(\boldsymbol{x})) - \|\boldsymbol{f}(\boldsymbol{x})\|_2^2\right],$$

and using a similar trick as in denoising score matching (Vincent, 2011; Song & Ermon, 2019), one can transform the minimization of $\mathcal{D}_{\text{Fisher}}$ into the following minimax problem which is tractable

$$\min_\phi \max_\psi \mathcal{L}_{\text{SIVI-SM}}(p(\boldsymbol{x}|D)\|q_\phi(\boldsymbol{x})) := \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}),\boldsymbol{x}\sim q_\phi(\boldsymbol{x}|\boldsymbol{z})}\left[2\boldsymbol{f}_\psi(\boldsymbol{x})^T[\boldsymbol{S}(\boldsymbol{x}) - \nabla_{\boldsymbol{x}} \log q_\phi(\boldsymbol{x}|\boldsymbol{z})] - \|\boldsymbol{f}_\psi(\boldsymbol{x})\|_2^2\right]. \tag{4}$$

In practice, $\boldsymbol{f}_\psi(\boldsymbol{x})$ is parametrized using neural networks. The above minimax optimization problem can be efficiently solved by optimizing ψ and ϕ alternately.

3 Hierarchical semi-implicit variational inference

The semi-implicit variational family $q_\phi(\boldsymbol{x})$ in equation (1) is indeed a single-layer model in the sense that it contains only one conditional layer. Our main idea is to expand this single-layer semi-implicit variational family into its multi-layer variants and introduce a sequence of auxiliary distributions to guide the semi-implicit distributions toward the target distribution. This leads to a new SIVI method which we call hierarchical semi-implicit variational inference (HSIVI). We start with the following definition which is motivated by equation (1).

Definition 1 (Hierarchical Semi-Implicit Distribution)**.** *Let $\boldsymbol{x}_T \sim q_T(\boldsymbol{x}_T)$ for some $T \in \mathbb{N}^*$, where $q_T(\boldsymbol{x}_T)$ is called the variational prior. Let $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ be the t-th conditional layer for $0 \le t \le T-1$. Denote $\{\phi_k\}_{k=t}^{T-1}$ by $\phi_{\ge t}$. The t-th layer hierarchical semi-implicit distribution $q_t(\boldsymbol{x}_t;\phi_{\ge t})$ is defined recursively from $T-1$ to 0 by*

$$q_t(\boldsymbol{x}_t;\phi_{\ge t}) = \int q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t) q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\ge t+1})\mathrm{d}\boldsymbol{x}_{t+1}, \quad 0 \le t \le T-1, \tag{5}$$

where $q_T(\boldsymbol{x}_T;\phi_{\ge T}) := q_T(\boldsymbol{x}_T)$. Here, the t-th conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ is required to be explicit and reparametrizable with a tractable score function $\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$.

Compared to the single-layer semi-implicit variational family (1), the family of hierarchical semi-implicit distributions provides a principled way to construct more expressive mixing layers using multi-layer architectures. Also, unlike the hierarchical variational models (Ranganath et al., 2016) which require an extra reverse model and explicit variational prior, hierarchical semi-implicit distributions inherit the advantage of SIVI that allows $q_t(\boldsymbol{x}_t;\phi_{\ge t})$ to be implicit, and as shown next, they do not require a reverse model and can be progressively trained using the simple algorithms of SIVI for each conditional layer, from $t = T-1$ to $t = 0$.

3.1 Progressive approximation with the auxiliary bridge

In this section, we introduce a bridging technique for progressively approximating the target distribution $p(\boldsymbol{x})$ using hierarchical semi-implicit distributions. Rather than approximating $p(\boldsymbol{x})$ with $q_0(\boldsymbol{x};\phi_{\ge 0})$ directly, we construct a sequence of intermediate auxiliary distributions $\{p_t(\boldsymbol{x})\}_{t=0}^{T-1}$ as a bridge between the target distribution $p_0(\boldsymbol{x}) := p(\boldsymbol{x})$ and an easy-to-approximate distribution $p_{T-1}(\boldsymbol{x})$, to amortize the difficulty of one-pass fitting. A typical example of an auxiliary bridge is the geometric interpolation as described below.

Algorithm 1 Hierarchical semi-implicit variational inference (sequential training)

Input: Auxiliary bridge $\{p_t(\boldsymbol{x})\}_{t=0}^{T-1}$; initial value of parameters $\phi^{(0)} = \{\phi_i^{(0)}\}_{t=0}^{T-1}$.
Output: The optimal parameters ϕ^*.
Initialization: $\phi \leftarrow \phi^{(0)}$.
for $t = T-1$ **to** 0 **do**
  while not converge **do**
    Sample a minibatch $\{\boldsymbol{x}_T^{(k)}\}_{k=1}^K$ from the variational prior $q_T(\boldsymbol{x}_T)$.
    if $t < T-1$ **then**
      Sequentially sample $\{\boldsymbol{x}_{t+1}^{(k)}\}_{k=1}^K$ through $q(\boldsymbol{x}_i|\boldsymbol{x}_{i+1};\phi_i)$ from $i = T-1$ to $i = t+1$.
      Detach the computation graphs from $\{\boldsymbol{x}_{t+1}^{(k)}\}_{k=1}^K$.
    end if
    Update ϕ_t by optimizing the $\mathcal{L}_{\text{SIVI-}f}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ based on the minibatch $\{\boldsymbol{x}_{t+1}^{(k)}\}_{k=1}^K$.
  end while
  $\phi_t^* \leftarrow \phi_t$.
end for
$\phi^* \leftarrow \{\phi_t^*\}_{t=0}^{T-1}$.

Example 1 (Geometric Interpolation)**.** *Let $\boldsymbol{S}(\boldsymbol{x}) := \nabla \log p(\boldsymbol{x})$ be the score function of target distribution $p(\boldsymbol{x})$ and $\boldsymbol{S}_{base}(\boldsymbol{x}) := \nabla \log p_{base}(\boldsymbol{x})$ be the score function of a base distribution $p_{base}(\boldsymbol{x})$. In geometric interpolation (Neal, 2001; Bernton et al., 2019), each auxiliary distribution $p_t(\boldsymbol{x})$ for $0 \leq t \leq T-1$ has the following probability density function (pdf) and score function*

$$p_t(\boldsymbol{x}) \propto p_{base}(\boldsymbol{x})^{1-\lambda_t} p(\boldsymbol{x})^{\lambda_t},\ \boldsymbol{S}_t(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) = (1-\lambda_t)\boldsymbol{S}_{base}(\boldsymbol{x}) + \lambda_t \boldsymbol{S}(\boldsymbol{x}), \quad (6)$$

where $\{\lambda_t\}_{t=0}^{T-1}$ is a non-negative decreasing sequence satisfying $\lambda_0 = 1$.

Intuitively, we expect the distance between two neighboring distributions $p_t(\boldsymbol{x})$ and $p_{t+1}(\boldsymbol{x})$ to be not too large so that it would be easy to construct a conditional distribution $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1})$ such that $p_t(\boldsymbol{x}_t) \approx \int q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}) p_{t+1}(\boldsymbol{x}_{t+1}) \mathrm{d}\boldsymbol{x}_{t+1}$. Note that the auxiliary bridge $\{p_t(\boldsymbol{x})\}_{t=0}^{T-1}$ does not necessarily need to have analytical pdfs (up to a constant). In fact, it suffices if they have tractable score functions $\{\boldsymbol{S}_t(\boldsymbol{x})\}_{t=0}^{T-1}$ which lead to another type of auxiliary bridge (Example 2 in Section 4).

3.2 Sequential training of HSIVI

Given the auxiliary distributions $\{p_t(\boldsymbol{x}_t)\}_{t=0}^{T-1}$, a natural approach is to progressively train the hierarchical semi-implicit distribution $q_t(\boldsymbol{x}_t;\phi_{\geq t})$ to match $p_t(\boldsymbol{x}_t)$ from $t = T-1$ to $t = 0$. Let the parameters ϕ_t in the t-th conditional layer be independent across different ts. We first train $q_{T-1}(\boldsymbol{x}_{T-1};\phi_{T-1})$ to match $p_{T-1}(\boldsymbol{x}_{T-1})$ by optimizing ϕ_{T-1} w.r.t. the single-layer SIVI objective $\mathcal{L}_{\text{SIVI-}f}\left(p_{T-1}(\boldsymbol{x}_{T-1})\|q_{T-1}(\boldsymbol{x}_{T-1};\phi_{T-1})\right)$. For $t = T-2, \ldots, 0$, given the trained semi-implicit distribution $q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\geq t+1})$, we can fix it as the mixing layer and train the t-th conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ by optimizing ϕ_t w.r.t. the single-layer SIVI objective $\mathcal{L}_{\text{SIVI-}f}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ as well. Note this is fine as the mixing layer can be implicit in SIVI. Here, f is some distance criterion, e.g. $\mathcal{L}_{\text{SIVI-LB}}$ in equation (2) or $\mathcal{L}_{\text{SIVI-SM}}$ in equation (4). In this article, we mainly focus on $\mathcal{L}_{\text{SIVI-LB}}$ and $\mathcal{L}_{\text{SIVI-SM}}$, while other distance criteria can also be applied. We summarize this sequential training procedure in Algorithm 1.

Score based training In addition to the common assumption that $p_t(\boldsymbol{x})$ is known up to a constant, it is worth noting that $\mathcal{L}_{\text{SIVI-LB}}$ is also applicable when only the score functions $\{\boldsymbol{S}_t(\boldsymbol{x})\}_{t=0}^{T-1}$ are available which is important for the diffusion bridge construction of auxiliary distributions in Example 2. Concretely, assume $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ is induced by a parametrized transform $\boldsymbol{x}_t = \boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon};\phi_t)$ where $\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$ is a random noise. The only term in $\mathcal{L}_{\text{SIVI-LB}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ containing $p_t(\boldsymbol{x}_t)$ is $\mathbb{E}_{q_t(\boldsymbol{x}_t;\phi_{\geq t})} \log p_t(\boldsymbol{x}_t)$ (see equation (2)) whose gradient takes the form

$$\nabla_{\phi_t} \mathbb{E}_{q_t(\boldsymbol{x}_t;\phi_{\geq t})} \log p_t(\boldsymbol{x}_t) = \mathbb{E}_{q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\geq t+1}) p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})} \boldsymbol{S}_t\left(\boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon};\phi_t)\right) \nabla_{\phi_t} \boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon};\phi_t). \quad (7)$$

In the training of HSIVI-SM, each term $\mathcal{L}_{\text{SIVI-SM}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ involves a nested optimization of $\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t)$. When the score functions are computationally expensive, we find that an alternative parametrization $\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t) := \boldsymbol{S}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)$ is useful to avoid the time-consuming evaluation of $\boldsymbol{S}_t(\boldsymbol{x}_t)$ when optimizing ψ_t in equation (4). The reason for this lies in Proposition 1. See Appendix C.2 for the proof of Proposition 1.

Proposition 1. *Let $q_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \phi_{\geq t}) = q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi_t) q_{t+1}(\boldsymbol{x}_{t+1}; \phi_{\geq t+1})$. The minimax optimization of $\mathcal{L}_{\text{SIVI-SM}}\left(p_t(\boldsymbol{x}_t) \| q_t(\boldsymbol{x}_t; \phi_{\geq t})\right)$ is equivalent to*

$$\min_{\phi_t} \quad \mathbb{E}_{q_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \phi_{\geq t})} \left[\boldsymbol{S}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t; \psi_t)\right]^T \left[\boldsymbol{S}_t(\boldsymbol{x}_t) + \boldsymbol{g}_t(\boldsymbol{x}_t; \psi_t) - 2\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi_t)\right],$$

$$\min_{\psi_t} \quad \mathbb{E}_{q_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}; \phi_{\geq t})} \|\boldsymbol{g}_t(\boldsymbol{x}_t; \psi_t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi_t)\|_2^2.$$

Marginal approximation v.s. joint approximation Previous works (Bernton et al., 2019; Bao et al., 2022) often construct a joint distribution $p(\boldsymbol{x}_{0:T})$ and minimize $\mathrm{KL}(p(\boldsymbol{x}_{0:T-1}) \| q(\boldsymbol{x}_{0:T-1}))$ where $q(\boldsymbol{x}_{0:T-1})$ is a variational distribution. In HSIVI, we directly approximate $p_t(\boldsymbol{x}_t)$ using the semi-implicit variational distributions. When $p(\boldsymbol{x}_{0:T-1})$ is complex and T is small, the variational distribution $q(\boldsymbol{x}_{0:T-1})$ may be insufficient to fully capture the joint distribution $p(\boldsymbol{x}_{0:T-1})$. For example, the optimal fit of the joint distribution for diffusion models established by Analytic-DPM (Bao et al., 2022) does not guarantee that the marginal distributions would be approximated well (see Table 2 for comparison).

4 Application to diffusion model acceleration

4.1 Review of diffusion models

Recently, diffusion models have shown great success on many generative modeling benchmarks, including image generation (Ho et al., 2020; Song et al., 2020a,b), graph generation (Niu et al., 2020), and text generation (Austin et al., 2021). Diffusion models work by adding noise to the training data in the forward process and then removing the noise to recover the data in the backward process, which can be integrated into a general stochastic differential equation (SDE) framework. The forward process $\{\boldsymbol{u}_s\}_{s\in[0,L]}$ is usually described by

$$\mathrm{d}\boldsymbol{u}_s = \boldsymbol{f}(\boldsymbol{u}_s, s)\mathrm{d}s + g(s)\mathrm{d}\boldsymbol{w}_s, \quad \boldsymbol{u}_0 \sim p_0(\cdot), \tag{8}$$

where $p_0(\cdot)$ is the data distribution, $\boldsymbol{w}_s$ is a standard Brownian motion, and $\boldsymbol{f}(\boldsymbol{u}_s, s)$ and $g(s)$ are the drift and diffusion coefficients respectively. To generate samples from the data distribution, one can run the following backward process

$$\mathrm{d}\boldsymbol{u}_s = [\boldsymbol{f}(\boldsymbol{u}_s, s) - g^2(s)\nabla_{\boldsymbol{u}_s} \log p_s(\boldsymbol{u}_s)]\mathrm{d}s + g(s)\mathrm{d}\bar{\boldsymbol{w}}_s, \quad \boldsymbol{u}_L \sim p_L(\cdot), \tag{9}$$

where $p_s(\cdot)$ is the pdf of $\boldsymbol{u}_s$ and $\bar{\boldsymbol{w}}_s$ is a standard Brownian motion when time flows from L to 0. As the score function $\nabla_{\boldsymbol{u}_s} \log p_s(\boldsymbol{u}_s)$ is intractable, we need to estimate it by denoising score matching (Vincent, 2011; Song et al., 2020b). See more details of diffusion models and the training objectives in Appendix A.

4.2 Diffusion model acceleration via HSIVI

While diffusion models prove effective for generative modeling, it often takes a large number of discretization steps in the backward process (9) to produce high quality samples, which caps their potential for real time applications. Note that the forward process (8) naturally provides another type of auxiliary bridge, which combined with HSIVI, can be used to accelerate the sampling process of diffusion models.

Example 2 (Diffusion Bridge)**.** *Consider the forward process $\{\boldsymbol{u}_s\}_{s\in[0,L]}$ with $L > 0$ (defined in equation (8)) in diffusion models. We choose T discrete time steps $0 \approx s_0 < \cdots < s_{T-1} \leq L$ and let $\boldsymbol{x}_t := \boldsymbol{u}_{s_t}$ with probability density function $p_t(\cdot)$. Assume each auxiliary distributions $p_t(\cdot)$ for $0 \leq t \leq T-1$ admits a score function as*

$$\boldsymbol{S}_t(\boldsymbol{x}) := \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}) \approx \boldsymbol{S}^*(\boldsymbol{x}, s_t),\ 0 \leq t \leq T-1,$$

where $\boldsymbol{S}^*(\boldsymbol{x}, s)$ is a pre-trained score model with the denoising score matching loss (equation (13) in Appendix A). Let us denote $\boldsymbol{S}^*(\boldsymbol{x}, s_t)$ by $\boldsymbol{S}_t^*(\boldsymbol{x})$ for short. With sufficient samples from the data distribution $p_0(\boldsymbol{x})$ and model capacity, the approximation $\boldsymbol{S}_t^*(\boldsymbol{x})$ can be reasonably accurate for almost all $\boldsymbol{x}$ and t (Song et al., 2020b).

As the pre-trained score model provides a diffusion bridge from the simple distribution p_{T-1} (e.g., standard Gaussian) to the data distribution, we can train the hierarchical semi-implicit distributions to

approximate the diffusion bridge within the HSIVI framework. Given the expressiveness of hierarchical semi-implicit distributions, we may expect an accurate approximation of the data distribution with a small number T and hence acceleration can be achieved.

However, the memory usage during the sequential training process for HSIVI might be large because of the necessity for independent parameters. Therefore, we may employ a parameter sharing scheme which is commonly assumed in diffusion models (Song & Ermon, 2019; Ho et al., 2020) such that different conditional layers share the same parameters ϕ. Note that sequential training is not suitable in this setting. Therefore, we propose a joint training procedure that minimizes a weighted sum of the SIVI objectives

$$\mathcal{L}_{\text{HSIVI-}f}(\phi) = \sum_{t=0}^{T-1} \beta(t) \mathcal{L}_{\text{SIVI-}f}\left(p_t(\boldsymbol{x}_t) \| q_t(\boldsymbol{x}_t; \phi)\right), \tag{10}$$

where $\beta(t) : \{0, \dots, T-1\} \to \mathbb{R}_+$ is a positive weighting function and f is some distance criterion. See Algorithm 2 in Appendix C.3 for more details of joint training.

More specifically, in this work, we mainly focus on building the diffusion bridge with variance preserving SDE (VP-SDE) (Song et al., 2020b) such that $\boldsymbol{u}_s | \boldsymbol{u}_0 \sim \mathcal{N}(\sqrt{\alpha(s)}\boldsymbol{u}_0, (1-\alpha(s))\mathbf{I})$ with a decreasing function $\alpha(s)$ of s. We use $\mathcal{L}_{\text{HSIVI-SM}}$ in equation (10) for training and set the weighting function $\beta(t) = 1 - \alpha(s_t)$ as recommended in Song et al. (2020b), which tends to train layers that are far from $t = 0$ first during the training, resembling the sequential training. Another popular formulation of diffusion models is to fit a noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ that predicts the noise added to a noisy sample $\boldsymbol{x}$ at time s (Ho et al., 2020). HSIVI-SM also generalizes to the case where a pre-trained noise model is available. The pre-trained noise model forms a (generalized) diffusion bridge by letting $\boldsymbol{\epsilon}_t^*(\boldsymbol{x}) = \boldsymbol{\epsilon}^*(\boldsymbol{x}, s_t)$, and we call the corresponding training method "ϵ-training". We provide a reparametrized objective function $\tilde{\mathcal{L}}_{\text{HSIVI-SM}}$ for ϵ-training in Appendix C.4.

Several efforts have been made to accelerate the sampling process of diffusion models, including faster numerical ordinary differential equation (ODE) solvers (Song et al., 2020a; Zhang & Chen, 2022; Lu et al., 2022) and distillation techniques (Luhman & Luhman, 2021; Salimans & Ho, 2022; Zheng et al., 2022). Our approach is different from these previous efforts in that we accelerate the stochastic diffusion model directly (hence would provide more diverse samples (Figure 6)) and do not require sampling datasets from the diffusion models prior to distillation which is computationally expensive. From a Bayesian perspective, HSIVI is related to Song & Ermon (2019), where the authors used the annealed Langevin dynamics guided by a pre-trained score model to sample from the data distribution. By solving this problem using a variational inference approach, HSIVI enjoys faster sampling speed and scales better to high-dimensional data.

5 Experiments

In this section, we first compare HSIVI to its single-layer counterpart, SIVI, on two inference tasks. We use the sequential training method where each conditional layer in the hierarchical semi-implicit variational distributions has independent parameters. We then apply HSIVI-SM to diffusion model acceleration on various datasets. As the memory consumption for generative models is large, we use the joint training method where the conditional layers in hierarchical semi-implicit distributions have shared parameters across different ts. For all experiments, each conditional layer is modeled as a Gaussian distribution with parametrized mean and variance. More details of the model architectures and hyper-parameters are included in Appendix E.

5.1 Target distribution approximation

Gaussian mixture model We first evaluate HSIVI and SIVI on a two-dimensional Gaussian mixture model. The target distribution $p(\boldsymbol{x})$ takes the form $p(\boldsymbol{x}) = \sum_{i=1}^{8} 1/8 \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \sigma^2 \mathbf{I})$ where $\boldsymbol{\mu}_i = [10\cos(\frac{i\pi}{4}), 10\sin(\frac{i\pi}{4})]^T$, $\sigma = 1$. For HSIVI, we construct an auxiliary bridge of $T = 5$ with geometric interpolation in Example 1, where $p_{\text{base}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \mathbf{0}, \mathbf{I})$ and $\lambda_t = 1 - t/5$. The results are presented in Figure 2. Note that the modes in this Gaussian mixture model are far apart from each other, and both SIVI-LB and SIVI-SM are trapped in local modes. In contrast, both HSIVI-LB and HSIVI-SM discover all modes and provide an accurate approximation of the target distribution with HSIVI-SM being better for recovering the right scale of variance.

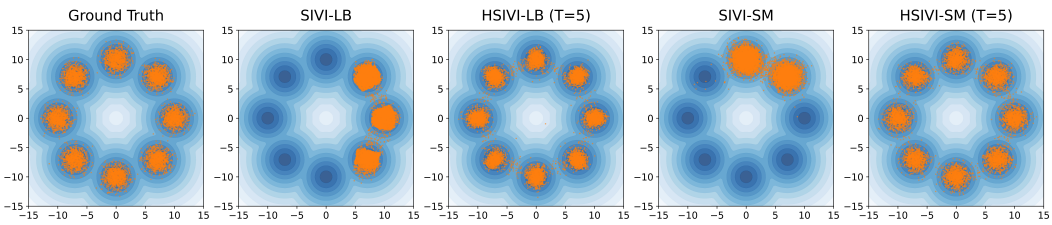


Figure 2: Comparison of 10,000 generated samples from SIVI and 5-layer HSIVI on a two-dimensional Gaussian mixture model (blue).

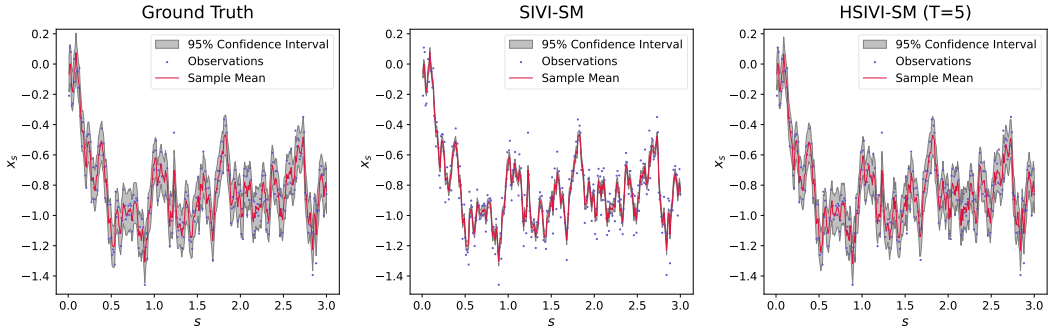


Figure 3: The posterior estimates obtained by different methods. For each method, we collect 100,000 samples to calculate the sample mean and confidence interval.

High-dimensional conditioned diffusion The second example is a high-dimensional Bayesian inference problem arising from the following Langevin SDE

$$\mathrm{d}x_s = 10x_s(1 - x_s^2)\mathrm{d}s + \mathrm{d}w_s, \tag{11}$$

where $x_0 = 0$ and w_s is a one-dimensional standard Brownian motion. This system describes the motion of a particle with negligible mass trapped in an energy potential with thermal fluctuations represented by the Brownian forcing (Cui et al., 2016). Using an Euler-Maruyama scheme with step size $\Delta s = 0.01$ on a time interval $[0, 3]$, we discretize the SDE (11) into $\boldsymbol{x} = (x_{d_1}, \ldots, x_{d_{300}})$ where $d_i = 0.01i$, which gives the prior distribution $p_{\text{prior}}(\boldsymbol{x})$ of the 300-dimensional variable $\boldsymbol{x}$. The noisy observations $\boldsymbol{y}$ is obtained by $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\xi}$, where $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ with $\sigma = 0.1$. Our goal is to infer the posterior distribution of the latent states $p(\boldsymbol{x}|\boldsymbol{y}) \propto p_{\text{prior}}(\boldsymbol{x})p(\boldsymbol{y}|\boldsymbol{x})$. The ground truth is formed by running 100,000 independent stochastic gradient Langevin dynamics (SGLD) chains with a step size of 0.0001 and collecting the results after 10,000 iterations.

For HSIVI, we form the auxiliary bridge using geometric interpolation with $p_{\text{base}}(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{y}, \sigma^2\mathbf{I})$ and $\lambda_t = 1 - t/(T-1)$ for $t = 0, \ldots, T-1$. Figure 3 shows the estimated posteriors obtained by different methods. We see that SIVI-SM severely underestimates the variance. With $T = 5$ layers, HSIVI-SM fits the variance better and hence provides more accurate posterior estimates. For both HSIVI-SM and HSIVI-LB, the estimated covariance matrix becomes more accurate as T increases (Table 3 in Appendix D.2), demonstrating the effectiveness of hierarchical models for fitting complicated distributions.

5.2 Diffusion model acceleration

2D toy examples In this toy model example, we test four synthetic 2D datasets: Checkerboard, Circles, Moons, and Swissroll (Pedregosa et al., 2011). We first pre-train the score model $\boldsymbol{S}^*(\boldsymbol{x}, s)$ for $s \in [0, 1]$ with quadratic noise schedule $1 - \alpha(s) = s^2$. For constructing the T-layer diffusion bridge, we select $\{s_t\}_{t=0}^{T-1}$ so that $1 - \alpha(s_t) = [0.01 + (\sqrt{0.8} - 0.01)t/T]^2$. Figure 4 shows the sample trajectories ($\boldsymbol{x}_9$, $\boldsymbol{x}_7$, $\boldsymbol{x}_5$ and $\boldsymbol{x}_0$) progressively generated from 10-layer HSIVI-SM. We see clearly how the semi-implicit distributions are guided towards the target distribution and all modes are discovered. We also report the Jensen-Shannon (JS) divergence between the target distributions and the estimated distributions in Table 1. We see that HSIVI-SM significantly improves upon DDIM and

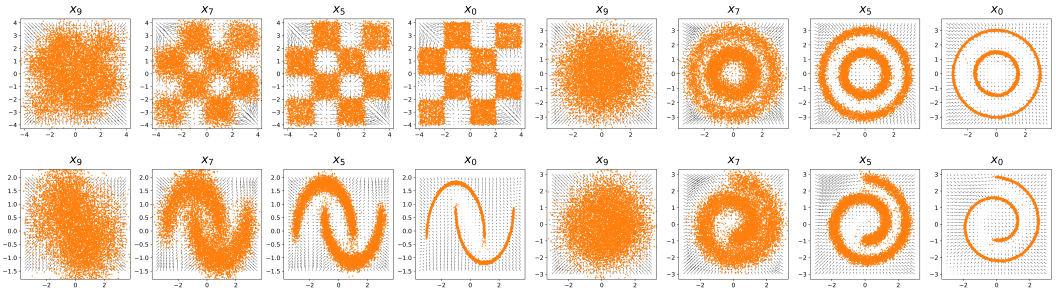


Figure 4: Sample trajectories generated from 10-layer HSIVI-SM on four 2D toy examples. The arrows represent the estimated score function in HSIVI-SM. The sample size is 10,000.

Table 1: JS divergences between the target distribution and the variational approximation on the four toy datasets. The results of HSIVI-SM are averaged by 5 independent runs with standard deviation in the subscripts. JS divergences are calculated by the ITE package (Szabó, 2014) with 10,000 samples.

Name	$T = 5$			$T = 10$			$T = 1000$
	DDPM	DDIM	HSIVI-SM	DDPM	DDIM	HSIVI-SM	DDPM
Checkerboard	0.891	0.591	$\mathbf{0.068}_{\pm \mathbf{0.006}}$	0.521	0.373	$\mathbf{0.030}_{\pm \mathbf{0.005}}$	0.058
Swissroll	1.037	0.332	$\mathbf{0.126}_{\pm \mathbf{0.006}}$	0.334	0.164	$\mathbf{0.082}_{\pm \mathbf{0.003}}$	0.042
Circles	0.907	0.397	$\mathbf{0.083}_{\pm \mathbf{0.015}}$	0.364	0.201	$\mathbf{0.073}_{\pm \mathbf{0.005}}$	0.032
Moons	0.961	0.355	$\mathbf{0.096}_{\pm \mathbf{0.013}}$	0.352	0.137	$\mathbf{0.059}_{\pm \mathbf{0.007}}$	0.036

DDPM in both cases with 5 and 10 steps. Also, 10-layer HSIVI-SM is comparable to DDPM with 1000 full steps. See Figure 9 in Appendix D.3 for visualization of samples from different methods.

MNIST On MNIST, we use the noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ instead of the score model and use ϵ-training to train HSIVI-SM. The structure of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ follows the UNet in Ho et al. (2020) by reducing the number of input and output channels to one. With the same noise schedule employed in Song et al. (2020a), we first pre-train the noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ with 1000 discretization steps and then form the T-layer diffusion bridge for HSIVI-SM by selecting T discrete time steps. Figure 5 shows the samples from DDPM, DDIM, and HSIVI-SM with $T = 5$ steps. We see that the samples produced by HSIVI-SM are much cleaner and more recognizable than those produced by DDPM and DDIM.

CIFAR-10 & CelebA On both CIFAR-10 and CelebA, the structure of our pre-trained noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ follows the UNet structure[1](Ronneberger et al., 2015) employed by Ho et al. (2020), instead of the huge VP deep continuous-time model (Song et al., 2020b) that has more channels and layers. Since this generative modeling has been formulated as a score-based VI problem, we do not have to use any training data for training HSIVI-SM. Following the noise schedule employed in Song et al. (2020a), we first pre-train the noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ with 1000 discretization steps and then form the T-layer diffusion bridge for HSIVI-SM by selecting T discrete time steps as before. For HSIVI-SM with ϵ-training, the conditional layer $q_t(\cdot|\boldsymbol{x}_{t+1}; \phi)$ is modeled as a Gaussian distribution with mean $\boldsymbol{\mu}_t(\boldsymbol{x}_{t+1}; \phi^\mu)$ and diagonal variance matrix $\boldsymbol{\Sigma}_t(\phi^\sigma)$ where $\{\phi^\mu, \phi^\sigma\} = \phi$ are the variational parameters. In our implementations, both $\boldsymbol{\mu}_t(\boldsymbol{x}_{t+1}; \phi^\mu)$ and $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ use the same architecture as $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$. The number of layers, which is also the number of function evaluations (NFE), is set to be $T = 5, 10, 15$ in our experiments. We train HSIVI-SM with the same setting for $T = 10, 15$. The 5-layer HSIVI-SM is trained by further fine-tuning the well-trained 15-layer HSIVI-SM and we find this strategy leads to better results. During each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$, we update ψ 20 times before each update of ϕ, since we find $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ needs more training empirically to provide reliable guidance.

[1]In our implementations, we reduce one downsampling block and one upsampling block in the UNet for CelebA so that the UNets used for the two datasets have the same structure. Therefore, the number of parameters for HSIVI-SM is 0.49× less than other methods on CelebA in Table 2. See Figure 12 and Table 4 in Appendix D.5 for the sampling time and the number of parameters comparison.

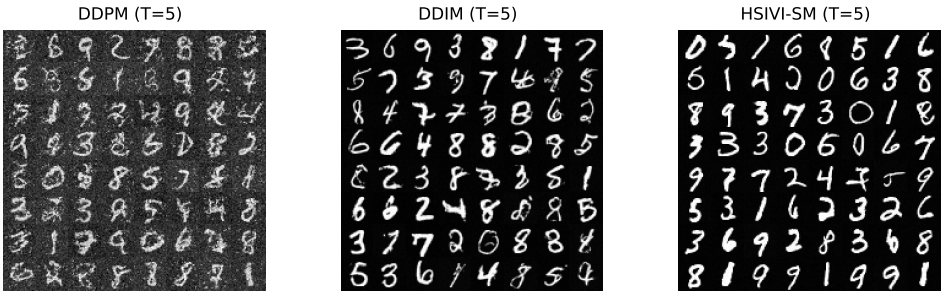


Figure 5: Comparison of the quality of uncurated samples generated by DDPM, DDIM, and HSIVI-SM with 5 discrete time steps on MNIST.

Table 2: Sample quality measured by FID (↓) on CIFAR-10 and CelebA, with a varying number of function evaluations (NFE). Results of baselines are calculated by running their official codes, where the architectures of score model (or noise model) are the UNet employed in Ho et al. (2020).

Dataset	CIFAR-10 (32×32)			CelebA (64×64)		
NFE	5	10	15	5	10	15
DDPM (Ho et al., 2020)	320.16	278.65	198.00	366.10	309.95	206.92
DDIM (Song et al., 2020a)	41.53	13.73	8.78	27.38	10.89	7.78
FastDPM (Kong & Ping, 2021)	67.64	9.85	6.16	27.63	15.44	12.05
Analytic-DDPM (Bao et al., 2022)	93.16	34.54	20.03	50.92	28.93	21.84
Analytic-DDIM (Bao et al., 2022)	51.86	14.08	8.65	29.40	15.74	12.25
DPM-Solver-fast (Lu et al., 2022)	329.13	10.89	4.67	355.96	6.76	**2.98**
HSIVI-SM (ours)	**6.27**	**4.31**	**4.17**	**8.29**	**4.95**	4.66

For each method, we draw 50,000 samples and use the Fréchet inception distance (FID) score (Karras et al., 2022) to evaluate the sample quality (Table 2). We find that HSIVI-SM performs on par or better than the other baselines on both CIFAR-10 and CelebA, and the advantage is evident when the NFE is small. The sampling trajectories of 10-layer HSIVI-SM on CelebA with the same starting point but different random seeds are shown in Figure 6. We see that HSIVI-SM is capable of producing more diverse samples due to its stochastic nature, which is different from existing ODE based fast diffusion model samplers.

Figure 6: Sample trajectories of 10-layer HSIVI-SM with the same starting point $\boldsymbol{x}_{10}$ on CelebA.

6 Conclusions

We introduced HSIVI, a hierarchical semi-implicit variational inference method that enables more expressive multi-layer construction of semi-implicit distributions. Given appropriate auxiliary distributions that interpolate between a simple base distribution and the target distribution, the conditional layers in hierarchical semi-implicit distributions can be progressively trained one layer after another. In experiments, we showed that HSIVI outperforms previous single-layer SIVI methods on several Bayesian inference tasks with complicated posteriors. HSIVI can also be used to accelerate the sampling process of diffusion models, where pre-trained score networks serve as a natural sequence of bridging distributions, which allows for direct acceleration of the stochastic diffusion model and does not require expensive sampling from the diffusion models during training. We showed that HSIVI can produce high quality samples comparable to or better than existing fast diffusion model samplers with few function evaluations on various datasets. Limitations are discussed in Appendix F.

References

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *arXiv preprint arXiv:2201.06503*, 2022.

Espen Bernton, Jeremy Heng, Arnaud Doucet, and Pierre E. Jacob. Schrödinger bridge samplers. *arXiv preprint arXiv:1912.13170*, 2019.

Christopher M. Bishop and Michael E Tipping. Variational relevance vector machines. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 46–53, 2000.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112:859 – 877, 2016.

Tiangang Cui, Kody JH Law, and Youssef M Marzouk. Dimension-independent likelihood-informed mcmc. *Journal of Computational Physics*, 304:109–137, 2016.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. In *International Conference on Learning Representations*, 2017.

R. J. Giordano, T. Broderick, and M. I. Jordan. Linear response methods for accurate covariance estimates from mean field variational bayes. In *Advances in Neural Information Processing Systems*, 2015.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Tianyang Hu, Zixiang Chen, Hanxi Sun, Jincheng Bai, Mao Ye, and Guang Cheng. Stein neural sampler. *arXiv preprint arXiv:1810.03545*, 2018.

Ferenc Huszár. Variational inference using implicit distributions. *arXiv preprint arXiv: 1702.08235*, 2017.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4):695–709, 2005. URL `http://jmlr.org/papers/v6/hyvarinen05a.html`.

T. S. Jaakkola and M. I. Jordan. Improving the mean field approximation via the use of mixture distributions. In *Learning in Graphical Models*, pp. 173–173, 1998.

Michael I. Jordan, Zoubin Ghahramani, T. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Proc. NeurIPS*, 2022.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pp. 4743–4751, 2016.

Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. *arXiv preprint arXiv:2106.00132*, 2021.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pp. 276–284. PMLR, 2016.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

Eric Luhman and Troy Luhman. Knowledge distillation in iterative generative models for improved sampling speed. *arXiv preprint arXiv:2101.02388*, 2021.

L. M. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017.

V. Moens, H. Ren, A. Maraval, R. Tutunov, J. Wang, and H. Ammar. Efficient semi-implicit variational inference. *arXiv preprint arXiv:2101.06070*, 2021.

Radford Neal. MCMC using hamiltonian dynamics. In S Brooks, A Gelman, G Jones, and XL Meng (eds.), *Handbook of Markov Chain Monte Carlo*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods. Taylor & Francis, 2011. ISBN 9781420079425. URL `http://books.google.com/books?id=qfRsAIKZ4rIC`.

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

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *International Conference on Artificial Intelligence and Statistics*, 2020.

D. J. Nott, S. L. Tan, M. Villani, and R. Kohn. Regression density estimation with variational methods and stochastic approximation. *Journal of Computational and Graphical Statistics*, 21(3):797–820, 2012.

J. W. Paisley, D. M. Blei, and M. I. Jordan. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning ICML*, 2012.

G. Papamakarios, E. Nalisnick, D. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *ArXiv Preprint arXiv:1912.02762*, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

R. Ranganath, S. Gerrish, and D. M. Blei. Black box variational inference. In *AISTATS*, pp. 814–822, 2014.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International conference on machine learning*, pp. 324–333. PMLR, 2016.

D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1530–1538, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *ArXiv*, abs/2202.00512, 2022.

L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In *Advances in Neural Information Processing Systems*, 1996.

J. Shi, S. Sun, and J. Zhu. Kernel implicit variational inference. In *International Conference on Learning Representations*, 2018a.

J. Shi, S. Sun, and J. Zhu. A spectral approach to gradient estimation for implicit distributions. In *International Conference on Machine Learning*, 2018b.

Artem Sobolev and Dmitry P. Vetrov. Importance weighted hierarchical variational inference. In *Neural Information Processing Systems*, 2019.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Y. Song, S. Garg, J. Shi, and S. Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2019.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012.

Zoltán Szabó. Information theoretical estimators toolbox. *Journal of Machine Learning Research*, 15:283–287, 2014.

Michalis K. Titsias and Francisco J. R. Ruiz. Unbiased implicit variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 167–176. PMLR, 2019.

D. Tran, D. M. Blei, and E. M. Airoldi. Copula variational inference. In *Advances in Neural Information Processing Systems*, 2015.

D. Tran, R. Ranganath, and D. M. Blei. Hierarchical implicit models and likelihood-free variational inference. In *Advances in Neural Information Processing Systems*, 2017.

Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Neural Information Processing Systems (NeurIPS)*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Maching Learning*, 1(1-2):1–305, 2008.

Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. In *International Conference on Machine Learning*, pp. 5646–5655, 2018.

Longlin Yu and Cheng Zhang. Semi-implicit variational inference via score matching. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=sd90a2ytrt`.

C. Zhang, B. Shahbaba, and H. Zhao. Variational hamiltonian monte carlo via score matching. *Bayesian Analysis*, 13(2):485–506, 2018.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

Hongkai Zheng, Weili Nie, Arash Vahdat, Kamyar Azizzadenesheli, and Anima Anandkumar. Fast sampling of diffusion models via operator learning. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022.

A Details of diffusion models

Diffusion models work by adding noise to the training data in the forward process and then removing the noise to recover the data in the backward process, which can be integrated into a general stochastic differential equation (SDE) framework (Song et al., 2020b). The forward process $\{\boldsymbol{u}_s\}_{s\in[0,L]}$ is usually described by the SDE

$$\mathrm{d}\boldsymbol{u}_s = \boldsymbol{f}(\boldsymbol{u}_s, s)\mathrm{d}s + g(s)\mathrm{d}\boldsymbol{w}_s, \quad \boldsymbol{u}_0 \sim p_0(\cdot),$$

where $p_0(\cdot)$ is the data distribution, $\boldsymbol{w}_s$ is a standard Brownian motion, $\boldsymbol{f}(\boldsymbol{u}_s, s)$ and $g(s)$ are the drift and diffusion coefficient respectively. To generate samples from the data distribution, one can run the following reversed SDE

$$\mathrm{d}\boldsymbol{u}_s = [\boldsymbol{f}(\boldsymbol{u}_s, s) - g^2(s)\nabla_{\boldsymbol{u}_s}\log p_s(\boldsymbol{u}_s)]\mathrm{d}s + g(s)\mathrm{d}\bar{\boldsymbol{w}}_s, \quad \boldsymbol{u}_L \sim p_L(\cdot),$$

where $p_s(\cdot)$ is the probability density function (pdf) of $\boldsymbol{u}_s$ and $\bar{\boldsymbol{w}}_s$ is a standard Brownian motion when time flows from L to 0. There exists deterministic process shares the same marginal probability densities $\{p_s(\cdot)\}_{s\in[0,L]}$ described by the following ordinary differential equation (ODE)

$$\mathrm{d}\boldsymbol{u}_s = [\boldsymbol{f}(\boldsymbol{u}_s, s) - \frac{1}{2}g^2(s)\nabla_{\boldsymbol{u}_s}\log p_s(\boldsymbol{u}_s)]\mathrm{d}s, \quad \boldsymbol{u}_L \sim p_L(\cdot),$$

called probability flow (PF) ODE.

In practice, Song et al. (2020b) and Kingma et al. (2021) designed several examples of the forward process such that it diffuses the data distribution $p_0(\cdot)$ to a fixed unstructured distribution $p_L(\cdot)$. Here we mainly consider the Variance Preserving SDE (VP-SDE) used in DDPM (Ho et al., 2020; Song et al., 2020b). Let the drift coefficient $\boldsymbol{f}(\boldsymbol{u}_s, s) = \frac{\mathrm{d}\log\alpha(s)}{2\mathrm{d}s}\boldsymbol{u}_s$ and the diffusion coefficient $g^2(s) = -\frac{\mathrm{d}\log\alpha(s)}{\mathrm{d}s}$, where $\alpha(s) \in \mathbb{R}^+$ is a decreasing smooth function with $\alpha(0) = 1, \alpha(L) \approx 0$. Then the distribution of $\boldsymbol{u}_s$ conditioned on $\boldsymbol{u}_0$ is explicit as

$$\boldsymbol{u}_s|\boldsymbol{u}_0 \sim \mathcal{N}\left(\sqrt{\alpha(s)}\bar{\boldsymbol{x}}_0, (1-\alpha(s))\mathbf{I}\right), \text{ i.e. } \boldsymbol{u}_s = \sqrt{\alpha(s)}\boldsymbol{u}_0 + \sqrt{1-\alpha(s)}\boldsymbol{\epsilon}, \tag{12}$$

where $\boldsymbol{\epsilon}$ is a standard Gaussian noise. In practice, diffusion models use a neural network $\boldsymbol{S}_\theta(\boldsymbol{u}_s, s)$ to approximate the score function $\boldsymbol{S}_\theta(\boldsymbol{u}_s, s)$ by optimizing the denoising score matching objective (Vincent, 2011)

$$\mathcal{L}_{\text{dsm}}(\theta, \omega(s)) := \frac{1}{2}\int_0^L \omega(s)\mathbb{E}_{\boldsymbol{u}_0\sim p_0(\boldsymbol{u}_0), \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\left\|\boldsymbol{S}_\theta(\boldsymbol{u}_s, s) + \boldsymbol{\epsilon}/\sqrt{1-\alpha(s)}\right\|_2^2 \mathrm{d}s, \tag{13}$$

where $\omega(s)$ is a positive weighting function. Instead of modeling the score function, Ho et al. (2020) proposed to predict the conditional noise $\boldsymbol{\epsilon}$ based on $\boldsymbol{u}_t$. This leads to the following DDPM loss

$$\mathcal{L}_{\text{ddpm}}(\theta, \bar{\omega}(s)) := \frac{1}{2}\int_0^L \bar{\omega}(s)\mathbb{E}_{\boldsymbol{u}_0\sim p_0(\boldsymbol{u}_0), \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I})}\|\boldsymbol{\epsilon}_\theta(\boldsymbol{u}_s, s) - \boldsymbol{\epsilon}\|_2^2 \mathrm{d}s, \tag{14}$$

where $\bar{\omega}(s)$ is a positive weighting function. In fact, we have the relationship

$$\boldsymbol{S}_\theta(\boldsymbol{u}_s, s) = -\boldsymbol{\epsilon}_\theta(\boldsymbol{u}_s, s)/\sqrt{1-\alpha(s)}. \tag{15}$$

We call $\mathcal{L}_{\text{dsm}}$ "score-prediction" training and $\mathcal{L}_{\text{ddpm}}$ "ϵ-prediction" training.

With the pre-trained score model $\boldsymbol{S}_\theta(\boldsymbol{u}_s, s)$ or noise model $\boldsymbol{\epsilon}_\theta(\boldsymbol{u}_s, s)$, Song et al. (2020b) shows that the samples of $p_0(\cdot)$ can be generated by simulating the backward SDE, e.g. the sampling scheme of DDPM (Ho et al., 2020). Moreover, Bao et al. (2022) proposed Analytic-DPM, the optimal discretization form responding to the KL divergence of the joint distribution on the discrete time steps. Also, several high-order ODE solvers (Song et al., 2020a; Zhang & Chen, 2022; Lu et al., 2022) were proposed to achieve faster sampling.

B More details of UIVI

Unlike optimizing the surrogate ELBO, Titsias & Ruiz (2019) proposed unbiased implicit variational inference (UIVI) which is based on an unbiased gradient estimator of the exact ELBO. More

specifically, consider a reparametrizable conditional $q_\phi(\boldsymbol{x}|\boldsymbol{z})$ such that $\boldsymbol{x} = T_\phi(\boldsymbol{z}, \boldsymbol{\epsilon}), \boldsymbol{\epsilon} \sim q_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}) \Leftrightarrow \boldsymbol{x} \sim q_\phi(\boldsymbol{x}|\boldsymbol{z})$, then

$$\begin{aligned}\nabla_\phi \text{ELBO} &= \nabla_\phi \mathbb{E}_{\boldsymbol{\epsilon}\sim q_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}), \boldsymbol{z}\sim q(\boldsymbol{z})} \left[\log p(D, \boldsymbol{x}) - \log q_\phi(\boldsymbol{x})|_{\boldsymbol{x}=T_\phi(\boldsymbol{z},\boldsymbol{\epsilon})}\right] \\ &= \mathbb{E}_{\boldsymbol{\epsilon}\sim q_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon}), \boldsymbol{z}\sim q(\boldsymbol{z})} \left[g_\phi^{\text{mod}}(\boldsymbol{z}, \boldsymbol{\epsilon}) + g_\phi^{\text{ent}}(\boldsymbol{z}, \boldsymbol{\epsilon})\right],\end{aligned}$$

where

$$\begin{aligned}g_\phi^{\text{mod}}(\boldsymbol{z}, \boldsymbol{\epsilon}) &:= \nabla_{\boldsymbol{x}} \log p(D, \boldsymbol{x})|_{\boldsymbol{x}=T_\phi(\boldsymbol{z},\boldsymbol{\epsilon})} \nabla_\phi T_\phi(\boldsymbol{z}, \boldsymbol{\epsilon}), \\ g_\phi^{\text{ent}}(\boldsymbol{z}, \boldsymbol{\epsilon}) &:= -\mathbb{E}_{q_\phi(\boldsymbol{z}'|\boldsymbol{x})} \nabla_{\boldsymbol{x}} \log q_\phi(\boldsymbol{x}|\boldsymbol{z}')\big|_{\boldsymbol{x}=T_\phi(\boldsymbol{z},\boldsymbol{\epsilon})} \nabla_\phi T_\phi(\boldsymbol{z}, \boldsymbol{\epsilon}).\end{aligned} \tag{16}$$

The gradient term in equation (16) involves an expectation w.r.t. the reverse conditional $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ which is be estimated using an MCMC sampler (e.g., Hamiltonian Monte Carlo (Neal, 2011)) in UIVI. However, the inner-loop MCMC runs may require long iterations for convergence.

C More details of HSIVI

C.1 Score-based training of HSIVI-LB

In the sequential training of HSIVI-LB, although the objective $\mathcal{L}_{\text{SIVI-LB}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ is calculated based on $p_t(\boldsymbol{x})$, the gradient of it w.r.t. ϕ_t has a closed form containing only the score function $\boldsymbol{S}_t(\boldsymbol{x})$ without knowing the corresponding pdfs. This derivation is important in the tasks where score functions of the auxiliary distributions are tractable while pdfs (up to a constant) of them are unavailable (for example, the diffusion bridge in Example 2). Concretely, assume the t-th conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ is induced by a parametrized transform $\boldsymbol{x}_t = \boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)$ where $\boldsymbol{\epsilon} \sim p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})$ is a random noise, since $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)$ is reparametrizable according to Definition 1. The only term in $\mathcal{L}_{\text{SIVI-LB}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ containing $p_t(\boldsymbol{x}_t)$ is $\mathbb{E}_{q_t(\boldsymbol{x}_t;\phi_{\geq t})} \log p_t(\boldsymbol{x}_t)$ (see equation (2)) whose gradient takes the form

$$\begin{aligned}\nabla_{\phi_t} \mathbb{E}_{q_t(\boldsymbol{x}_t;\phi_{\geq t})} \log p_t(\boldsymbol{x}_t) &= \nabla_{\phi_t} \mathbb{E}_{q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\geq t+1})p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})} \log p_t(\boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)) \\ &= \mathbb{E}_{q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\geq t+1})p_{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon})} \boldsymbol{S}_t\left(\boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)\right) \nabla_{\phi_t} \boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)\end{aligned}$$

by the chain rule, where $\nabla_{\phi_t} \boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)$ is the jacobian matrix of $\boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t)$.

In our implementation of HSIVI (in both sequential training and joint training), we generally assume the conditional layer $q_t(\cdot|\boldsymbol{x}_{t+1};\phi_t)$ is induced by

$$\boldsymbol{h}_t(\boldsymbol{x}_{t+1}, \boldsymbol{\epsilon}; \phi_t) = \boldsymbol{\mu}_t(\boldsymbol{x}_{t+1};\phi_t) + \boldsymbol{\Sigma}_t^{1/2}(\boldsymbol{x}_{t+1};\phi_t)\boldsymbol{\epsilon} \tag{17}$$

where $\boldsymbol{\Sigma}_t(\boldsymbol{x}_{t+1};\phi_t)$ is a positive definite covariance matrix and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a standard multivariate gaussian variable. In equation (17), ϕ_t should be replaced by ϕ in the joint training case.

C.2 Proof of Proposition 1

Proposition 1. *Let $q_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1};\phi_{\geq t}) = q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)q_{t+1}(\boldsymbol{x}_{t+1};\phi_{\geq t+1})$. The minimax optimization of $\mathcal{L}_{\text{SIVI-SM}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ is equivalent to*

$$\begin{aligned}&\min_{\phi_t} \quad \mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\left[\boldsymbol{S}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)\right]^T \left[\boldsymbol{S}_t(\boldsymbol{x}_t) + \boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t) - 2\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\right], \\ &\min_{\psi_t} \quad \mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\|\boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\|_2^2.\end{aligned}$$

Proof of Propsition 1 The minimax optimization problem of $\mathcal{L}_{\text{SIVI-SM}}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi_{\geq t})\right)$ is

$$\min_{\phi_t}\max_{\psi_t} \mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_t;\phi_{\geq t})}\left[2\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t)^T[\boldsymbol{S}_t(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)] - \|\boldsymbol{f}_t(\boldsymbol{x};\psi_t)\|_2^2\right]$$

according to equation (4). For minimization w.r.t. ϕ_t, this target is equivalent to

$$\begin{aligned}&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\left[2\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t)^T[\boldsymbol{S}_t(\boldsymbol{x}_t) - \nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)] - \|\boldsymbol{f}_t(\boldsymbol{x};\psi_t)\|_2^2\right] \\ =&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t)^T[2\boldsymbol{S}_t(\boldsymbol{x}_t) - \boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t) - 2\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)] \\ =&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}[\boldsymbol{S}_t(\boldsymbol{x}_t) - \boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)]^T[\boldsymbol{S}_t(\boldsymbol{x}_t) + \boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t) - 2\nabla_{\boldsymbol{x}_t} \log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)].\end{aligned}$$

For maximization w.r.t. ψ_t, this target is equivalent to

$$\begin{aligned}
&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\left[2\boldsymbol{f}_t(\boldsymbol{x}_t;\psi_t)^T[\boldsymbol{S}_t(\boldsymbol{x}_t)-\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)]-\|\boldsymbol{f}_t(\boldsymbol{x};\psi_t)\|_2^2\right]\\
=&-\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\|\boldsymbol{f}_t(\boldsymbol{x};\psi_t)-\boldsymbol{S}_t(\boldsymbol{x}_t)+\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\|_2^2+C\\
=&-\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\|\boldsymbol{g}_t(\boldsymbol{x};\psi_t)-\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\|_2^2+C,
\end{aligned}$$

where C is a term that does not contain ψ_t. Therefore, the minimax optimization problem is equivalent to

$$\begin{aligned}
\min_{\phi_t}\quad&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\left[\boldsymbol{S}_t(\boldsymbol{x}_t)-\boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)\right]^T\left[\boldsymbol{S}_t(\boldsymbol{x}_t)+\boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)-2\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\right],\\
\min_{\psi_t}\quad&\mathbb{E}_{q_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi_{\geq t})}\|\boldsymbol{g}_t(\boldsymbol{x}_t;\psi_t)-\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi_t)\|_2^2.
\end{aligned}$$

C.3 Joint training of HSIVI

As mentioned in Section 4.2, when parameter sharing scheme is used in the conditional layers for application to diffusion model acceleration, sequential training from $t=T-1$ to $t=0$ is not feasible. Therefore, we consider the following training objective

$$\mathcal{L}_{\text{HSIVI-}f}(\phi)=\sum_{t=0}^{T-1}\beta(t)\mathcal{L}_{\text{SIVI-}f}\left(p_t(\boldsymbol{x}_t)\|q_t(\boldsymbol{x}_t;\phi)\right).$$

An intuitive method is to randomly sample a batch of time steps $\{t_k\}_{k=1}^K$ and for each t_k train $\mathcal{L}_{\text{SIVI-}f}\left(p_{t_k}(\boldsymbol{x}_{t_k})\|q_{t_k}(\boldsymbol{x}_{t_k};\phi)\right)$ directly. However, sequentially sampling $\boldsymbol{x}_{t_k}$ through $q(\boldsymbol{x}_i|\boldsymbol{x}_{i+1};\phi)$ from $i=T-1$ to $i=t_k$ is still necessary in this case, making it memory-consuming to preserve the computation graphs of the entire sampling process.

In order to reduce the cost of accumulating computation graphs, for each t, we treat $q_{t+1}(\boldsymbol{x}_{t+1};\phi)$ as a fixed mixing layer denoted by $\tilde{q}_{t+1}(\boldsymbol{x}_{t+1})$ and only fit the conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)$. More specifically, for HSIVI-SM, we consider the following optimization problem

$$\min_{\phi}\sum_{t=0}^{T-1}\beta(t)\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi)}\left[\boldsymbol{S}_t(\boldsymbol{x}_t)-\boldsymbol{g}_t(\boldsymbol{x}_t;\psi)\right]^T\left[\boldsymbol{S}_t(\boldsymbol{x}_t)+\boldsymbol{g}_t(\boldsymbol{x}_t;\psi)-2\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\right], \quad (18)$$

$$\min_{\psi}\sum_{t=0}^{T-1}\beta(t)\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi)}\|\boldsymbol{g}_t(\boldsymbol{x}_t;\psi)-\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\|_2^2, \quad (19)$$

where $\tilde{q}_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi)=q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\tilde{q}_{t+1}(\boldsymbol{x}_{t+1})$. In what follows, we demonstrate that the above problem also ensures an accurate approximation of the target score function.

For equation (19), by the denoising score matching trick (Hyvärinen, 2005), the optimal point of ψ, denoted by $\psi^*(\phi)$, satisfies

$$\boldsymbol{g}_t(\boldsymbol{x}_t;\psi^*(\phi))=\nabla_{\boldsymbol{x}_t}\log\tilde{q}_t(\boldsymbol{x}_t;\phi),$$

where $\tilde{q}_t(\boldsymbol{x}_t;\phi)=\int q(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\tilde{q}(\boldsymbol{x}_{t+1})\mathrm{d}\boldsymbol{x}_{t+1}$. By plugging in the optimal point $\psi^*(\phi)$, each term in equation (18) is equivalent to

$$\begin{aligned}
&\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t,\boldsymbol{x}_{t+1};\phi)}\left[\boldsymbol{S}_t(\boldsymbol{x}_t)-\boldsymbol{g}_t(\boldsymbol{x}_t;\psi^*(\phi))\right]^T\left[\boldsymbol{S}_t(\boldsymbol{x}_t)+\boldsymbol{g}_t(\boldsymbol{x}_t;\psi^*(\phi))-2\nabla_{\boldsymbol{x}_t}\log q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\right]\\
=&\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t;\phi)}\left[\boldsymbol{S}_t^2(\boldsymbol{x}_t)-\boldsymbol{g}_t^2(\boldsymbol{x}_t;\psi^*(\phi))\right]-2\iint\tilde{q}(\boldsymbol{x}_{t+1})\left[\boldsymbol{S}_t(\boldsymbol{x}_t)-\boldsymbol{g}_t(\boldsymbol{x}_t;\psi^*(\phi))\right]^T\nabla_{\boldsymbol{x}_t}q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1};\phi)\mathrm{d}\boldsymbol{x}_{t+1}\mathrm{d}\boldsymbol{x}_t\\
=&\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t;\phi)}[\boldsymbol{S}_t^2(\boldsymbol{x}_t)-\boldsymbol{g}_t^2(\boldsymbol{x}_t;\psi^*(\phi))]-2\int\left[\boldsymbol{S}_t(\boldsymbol{x}_t)-\boldsymbol{g}_t(\boldsymbol{x}_t;\psi^*(\phi))\right]^T\nabla_{\boldsymbol{x}_t}\tilde{q}_t(\boldsymbol{x}_t;\phi)\mathrm{d}\boldsymbol{x}_t\\
=&\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t;\phi)}\left[\boldsymbol{S}_t^2(\boldsymbol{x}_t)-2\boldsymbol{S}_t(\boldsymbol{x}_t)^T\nabla_{\boldsymbol{x}_t}\log\tilde{q}_t(\boldsymbol{x}_t;\phi)+\left(\nabla_{\boldsymbol{x}_t}\log\tilde{q}_t(\boldsymbol{x}_t;\phi)\right)^2\right]\\
=&\mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t;\phi)}\|\boldsymbol{S}_t(\boldsymbol{x}_t)-\nabla_{\boldsymbol{x}_t}\log\tilde{q}_t(\boldsymbol{x}_t;\phi)\|^2.
\end{aligned}$$

Therefore, the global optimal point ϕ^* also ensures that the score of the variational distribution fits the target score function.

Based on the training objectives (18) (19) mentioned above, we propose Algorithm 2 for joint training, which does not need to store the computation graphs of the sample sequences. Moreover, by assuming an increasing weighting function $\beta(t)$, we assign larger weights $\beta(t)$ for those t close to $T-1$, which tends to train the conditional layers that are close to $T-1$ first during the training, resembling the sequential training.

Algorithm 2 Hierarchical semi-implicit variational inference (joint training)

Input: Auxiliary bridge $\{p_t(\boldsymbol{x})\}_{t=0}^{T-1}$; a weighting function $\beta(t)$; initial value of parameters $\phi^{(0)}$.
Output: The optimal parameters ϕ^*.
Initialization: $\phi \leftarrow \phi^{(0)}$.
while not converge **do**
  Uniformly sample K time steps $\{t_k\}_{k=0}^K$ with replacement from $\{0, \dots, T-1\}$.
  Sample a minibatch $\{\boldsymbol{x}_T^{(k)}\}_{k=1}^K$ from the base distribution $q_T(x)$.
  for $k = 1, \dots, K$ and $t_k < T-1$ **do**
    Sequentially sample $\boldsymbol{x}_{t_k+1}^{(k)}$ through $q(\boldsymbol{x}_i|\boldsymbol{x}_{i+1};\phi_i)$ from $i = T-1$ to $i = t_k+1$.
    Detach the computation graphs from $\{x_{t_k+1}^{(k)}\}_{k=1}^K$.
  end for
  Update ϕ by optimizing the objective $\sum_{k=1}^K \beta(t_k)\mathcal{L}_{\text{SIVI-}f}\left(p_{t_k}(\boldsymbol{x}_{t_k})\|q_{t_k}(\boldsymbol{x}_{t_k};\phi)\right)$, where the k-th term is computed based on a single sample $\boldsymbol{x}_{t_k+1}^{(k)}$.
end while
$\phi^* \leftarrow \phi$.

C.4 ϵ-training of HSIVI-SM

Another popular formulation of diffusion models is modeling the conditional noise $\boldsymbol{\epsilon}_\theta(\boldsymbol{u}_s, s)$ by optimizing the DDPM loss in equation (14) where $\boldsymbol{u}_s = \sqrt{\alpha(s)}\boldsymbol{u}_0 + \sqrt{1-\alpha(s)}\boldsymbol{\epsilon}$, introduced as "$\epsilon$-prediction" in Appendix A. Now, let us assume the diffusion bridge is constructed with VP-SDE and we have a pre-trained model of conditional noise $\boldsymbol{\epsilon}^*(\boldsymbol{u}, s)$. Similarly, we construct a sequence of noise models $\{\boldsymbol{\epsilon}_t^*(\boldsymbol{x}_t)\}_{t=0}^{T-1}$ by letting $\boldsymbol{x}_t = \boldsymbol{u}_{s_t}$ and $\boldsymbol{\epsilon}_t^*(\boldsymbol{x}) = \boldsymbol{\epsilon}^*(\boldsymbol{x}, s_t)$ which forms a (generalized) T-layer diffusion bridge. We only discuss how ϵ-training can be applied to joint training and the derivation for sequential training is similar. In what follows, we consider the transformation of the joint training objective $\mathcal{L}_{\text{HSIVI-SM}}$ for diffusion model acceleration.

By letting the weighting function $\beta(t) = 1 - \alpha(s_t)$ and considering the reparametrization form (17) where ϕ_t is replaced by ϕ, the objective of HSIVI-SM takes the form

$$\mathcal{L}_{\text{HSIVI-SM}}(\phi, \psi) = \sum_{t=0}^{T-1} \mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1};\phi)} \Big[2\sqrt{\beta(t)}\boldsymbol{f}_t(\boldsymbol{x}_t;\psi)^T[\sqrt{\beta(t)}\boldsymbol{S}_t^*(\boldsymbol{x}_t) + \sqrt{\beta(t)}\boldsymbol{\Sigma}_t^{-1/2}(\boldsymbol{x}_{t+1};\phi)\boldsymbol{\epsilon})] - \|\sqrt{\beta(t)}\boldsymbol{f}_t(\boldsymbol{x}_t;\psi)\|_2^2\Big]. \tag{20}$$

where $\boldsymbol{S}_t^*(\boldsymbol{x}_t)$ is a pre-trained score model. Note that we have $\sqrt{\beta(t)}\boldsymbol{S}_t^*(\boldsymbol{x}_t) = -\boldsymbol{\epsilon}_t^*(\boldsymbol{x}_t)$ by equation (15). Define

$$\tilde{\boldsymbol{f}}_t(\boldsymbol{x}_t;\psi) = \sqrt{\beta(t)}\boldsymbol{f}_t(\boldsymbol{x}_t;\psi),$$
$$\tilde{\boldsymbol{\Sigma}}_t(\boldsymbol{x}_{t+1};\phi) = \boldsymbol{\Sigma}_t(\boldsymbol{x}_{t+1};\phi)/\beta(t).$$

The HSIVI-SM objective (20) then takes the form

$$\tilde{\mathcal{L}}_{\text{HSIVI-SM}}(\phi, \psi) = \sum_{t=0}^{T-1} \mathbb{E}_{\tilde{q}_t(\boldsymbol{x}_t, \boldsymbol{x}_{t+1};\phi)} \Big[2\tilde{\boldsymbol{f}}_t(\boldsymbol{x}_t;\psi)^T[-\boldsymbol{\epsilon}_t^*(\boldsymbol{x}_t) + \tilde{\boldsymbol{\Sigma}}_t^{-1/2}(\boldsymbol{x}_{t+1};\phi)\boldsymbol{\epsilon})] - \|\tilde{\boldsymbol{f}}_t(\boldsymbol{x}_t;\psi)\|_2^2\Big] \tag{21}$$

and we call it the objective for ϵ-training. In our implementation of ϵ-training, we directly parametrize $\tilde{\boldsymbol{f}}_t(\boldsymbol{x}_t;\psi)$ and $\tilde{\boldsymbol{\Sigma}}_t(\boldsymbol{x}_{t+1};\phi)$ instead of $\boldsymbol{f}_t(\boldsymbol{x}_t;\psi)$ and $\boldsymbol{\Sigma}_t(\boldsymbol{x}_{t+1};\phi)$. The objective (21) is more numerically stable since the magnitude of $\tilde{\boldsymbol{\Sigma}}_t(\boldsymbol{x}_{t+1};\phi)$ is generally larger than $\boldsymbol{\Sigma}_t(\boldsymbol{x}_{t+1};\phi)$.

D Additional results of experiments

D.1 Gaussian mixture model

For HSIVI on the Gaussian mixture model, the auxiliary distributions can also be constructed with diffusion bridge in Example 2. Concretely, the diffusion bridge is constructed by

$$\boldsymbol{x}_t|\boldsymbol{x}_0 \sim \mathcal{N}(\sqrt{\alpha_t}\boldsymbol{x}_0, (1-\alpha_t)\mathbf{I}), \quad \boldsymbol{x}_0 \sim p_0(\boldsymbol{x}_0).$$

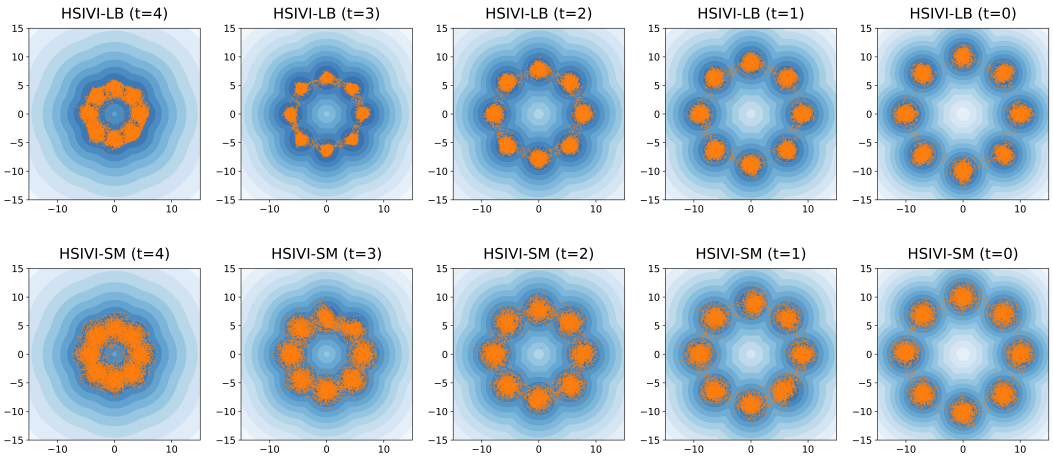


Figure 7: **Upper row**: Sample trajectories progressively generated by 5-layer HSIVI-LB guided by diffusion bridge. **Bottom row**: Sample trajectories progressively generated by 5-layer HSIVI-SM guided by diffusion bridge.

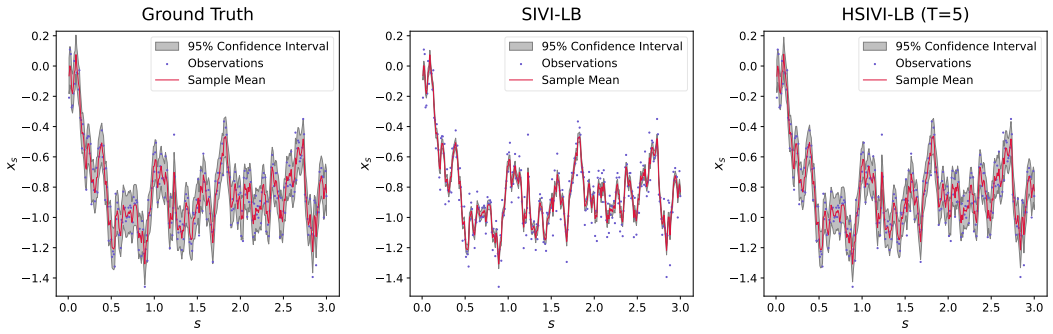


Figure 8: The posterior estimates for conditioned diffusion obtained by SIVI-LB and 5-layer HSIVI-LB. For each method, we collect 100,000 samples to calculate the sample mean and confidence interval.

where $\alpha_t = \alpha(s_t)$ with $\alpha(s)$ defined in equation (12). In this example, the score function $\boldsymbol{S}_t(\boldsymbol{x}_t) = \nabla_{\boldsymbol{x}_t} \log p_t(\boldsymbol{x}_t)$ has an analytical form

$$\boldsymbol{S}_t(\boldsymbol{x}_t) = \boldsymbol{S}_0\left(\boldsymbol{x}_t; \sqrt{\alpha_t}\boldsymbol{\mu}, (\alpha_t\sigma^2 + 1 - \alpha_t)\mathbf{I}\right), \quad 0 \le t \le T-1.$$

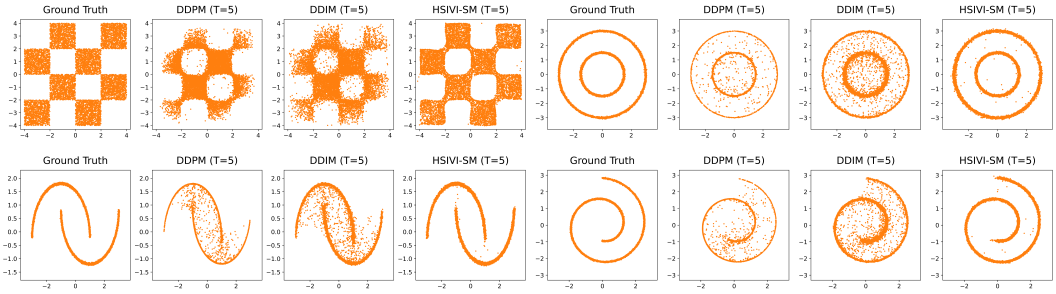
where $\boldsymbol{S}_0(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\mathbf{I})$ is the score function of the Gaussian mixture model $p(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^2\mathbf{I}) = \sum_{i=1}^{8} 1/8 \cdot \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_i, \sigma^2\mathbf{I})$. We set the number of layers $T = 5$ and $\alpha_t = 1 - t/5$ for $t = 0, \ldots, 4$. Figure 7 shows the sample trajectories generated by HSIVI. We see clearly that semi-implicit distributions are guided toward the target distribution following the diffusion bridge.

D.2 High-dimensional conditioned diffusion

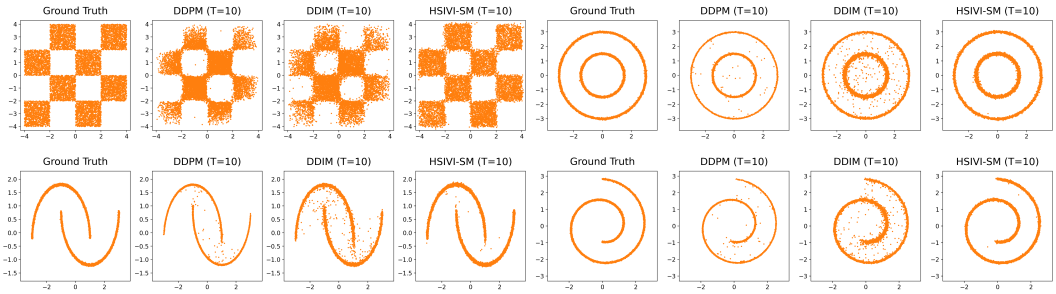
We also test SIVI-LB and HSIVI-LB for fitting the posterior in high-dimensional conditioned diffusion. The auxiliary bridge is formed using the same geometric interpolation as for HSIVI-SM, i.e.

$$p_{\text{base}} = \mathcal{N}(\boldsymbol{x}; \boldsymbol{y}, \sigma^2\mathbf{I}), \quad \lambda_t = 1 - \frac{t}{T-1} \text{ for } 0 \le t \le T-1.$$

From Figure 8, we see that SIVI-LB also underestimates the posterior variance and 5-layer HSIVI-LB fits the variance better. This phenomenon is also observed in the performances of SIVI-SM and HSIVI-SM in Figure 3. The quantitative comparison between different numbers of layers is reported in Table 3, where we see that for both HSIVI-SM and HSIVI-LB, the variational approximation gets more accurate with more layers. We also find that HSIVI-SM fits better than HSIVI-LB consistently.



(a) 5 steps



(b) 10 steps

Figure 9: Comparison of 10,000 samples generated by DDPM, DDIM, and HSIVI-SM.

Table 3: Frobenius distances between the estimated covariance matrices and that of the ground truth. For each method, we collect 100,000 samples to estimate the covariance matrix.

	$T = 1$	$T = 2$	$T = 3$	$T = 5$
HSIVI-SM	0.0886	0.0813	0.0431	**0.0333**
HSIVI-LB	0.0883	0.0825	0.0722	**0.0433**

D.3 Toy examples of diffusion model acceleration

We compare the samples from DDPM, DDIM, and our proposed HSIVI-SM with 5 and 10 steps in Figure 9. We find that DDIM and DDPM fail to converge to the target distribution with a small number of steps, while HSIVI-SM can provide noticeably better samples. Moreover, DDPM tends to underestimate the variance as evidenced by the narrower region occupied by the samples.

D.4 MNIST

Figure 10 shows the samples from DDPM, DDIM, and HSIVI-SM with $T = 10$ steps. We see that the samples produced by HSIVI-SM is much cleaner and more recognizable than those produced by DDPM and DDIM.

D.5 CIFAR-10 & CelebA

Figure 11 shows the uncurated samples from our proposed HSIVI-SM method with different numbers of layers on CIFAR-10 and CelebA. We also compare the sampling time of different methods when NFE $= 5$ in Figure 12. One can observe that HSIVI-SM has almost the same running time as the simplest DDIM algorithm and is faster than other samplers on CelebA. Finally, we report the number of parameters in the score model (or noise model) used by different methods in Table 4, which corresponds to Table 2 and Figure 12. In our implementations of HSIVI-SM, the number of parameters in the noise model equals that in the conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi)$. We find that our model with the same or less number of parameters still reaches comparable results in Table 2.

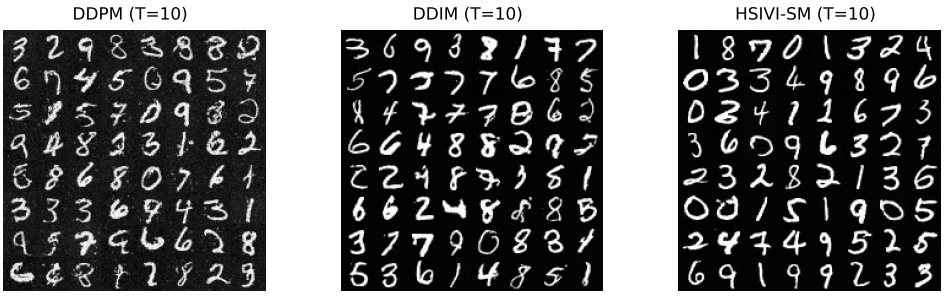
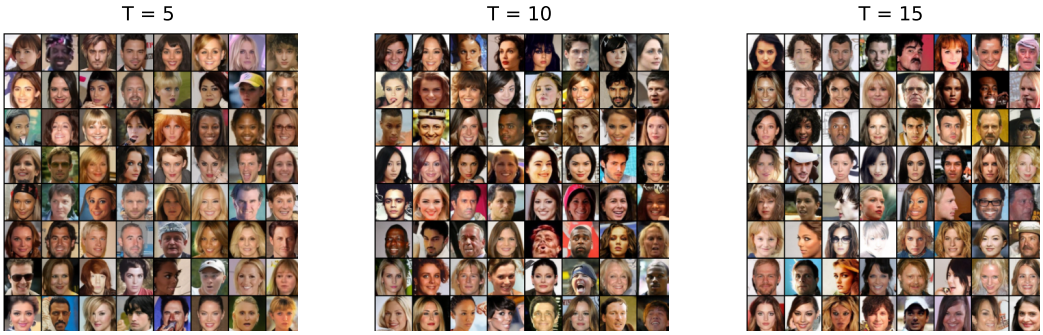


Figure 10: Comparison of the quality of uncurated samples generated by DDPM, DDIM, and HSIVI-SM with 10 discrete time steps on MNIST.

(a) CIFAR-10 (32×32)



(b) CelebA (64×64)

Figure 11: Uncurated samples generated by HSIVI-SM with different numbers of layers on CIFAR-10 and CelebA.

E Experimental details

E.1 Target distribution approximation

In this part, we set the conditional layer to be $q_\phi(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}(\boldsymbol{z}; \phi^\mu), \text{diag}\{\exp(\phi^\sigma)\})$ and the mixing layer to be $\mathcal{N}(\mathbf{0}, \mathbf{I})$ for SIVI. Here, $\{\phi^\mu, \phi^\sigma\} = \phi$ are the variational parameters. For T-layer hierarchical semi-implicit variational distribution with $T \geq 2$, the variational prior $q_T(\boldsymbol{x}_T)$ is set to be $\mathcal{N}(\mathbf{0}, \mathbf{I})$. Each conditional layer $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi_t)$ for $t = 0, \ldots, T-1$ is a conditional Gaussian distribution

$$q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi_t) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu), \text{diag}\{\exp(\phi_t^\sigma)\}).$$

Note that the ϕ^σ and $\{\phi_t^\sigma\}_{t=0}^{T-1}$ above are all vectors with the same dimension as $\boldsymbol{x}$. We use sequential training for HSIVI in the two experiments in this part. The parameters $\{\phi_t\}_{t=0}^{T-1}$ are independent

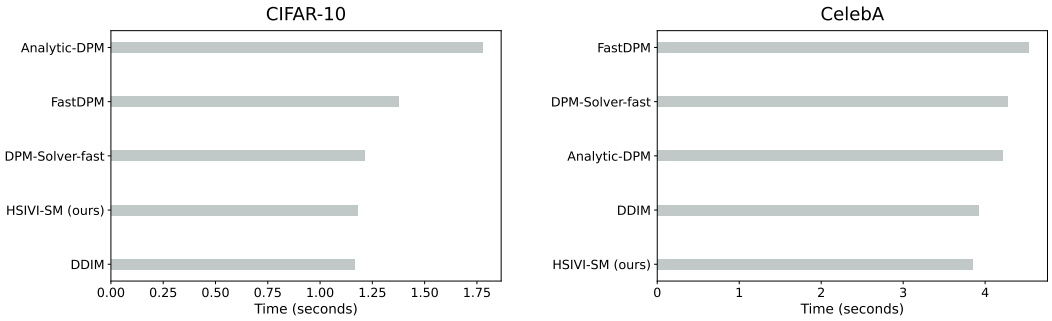


Figure 12: Sampling time (↓) of different methods when NFE $= 5$ on CIFAR-10 and CelebA. Results are averaged by 100 independent runs with a batch size of 128 on a single Nvidia 2080Ti GPU.

Table 4: Number of parameters in the score model (or noise model) used by different methods in Table 2. 'M' refers to million.

	CIFAR-10	CelebA
other methods	38.72M	78.66M
HSIVI-SM (ours)	38.72M	38.72M

across different t. If not otherwise specified, we use the Adam optimizer (Kingma & Ba, 2015) with $\beta = (0.9, 0.99)$ for training.

E.1.1 Gaussian mixture model

For the experiment on the Gaussian mixture model, we construct 5-layer hierarchical semi-implicit variational distributions. The mean of each conditional layer $\boldsymbol{\mu}(\boldsymbol{z}; \phi^\mu)$ in SIVI or $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu)$ in HSIVI has a residual form, i.e. $\boldsymbol{\mu}(\boldsymbol{z}; \phi^\mu) = \boldsymbol{z} + \bar{\boldsymbol{\mu}}(\boldsymbol{z}; \phi^\mu)$ and $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu) = \boldsymbol{x}_{t+1} + \bar{\boldsymbol{\mu}}(\boldsymbol{x}_{t+1}; \phi_t^\mu)$, for $t = 0, \ldots, T-1$. $\bar{\boldsymbol{\mu}}(\boldsymbol{z}; \phi^\mu)$ in SIVI and $\{\bar{\boldsymbol{\mu}}(\boldsymbol{x}_{t+1}; \phi_t^\mu)\}_{t=0}^4$ in HSIVI all have the same structures of multi-layer perceptrons (MLPs) with layer widths $[2, 50, 50, 2]$ and ReLU activation functions. For each t, $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi_t)$ in HSIVI-SM and $\boldsymbol{f}(\boldsymbol{x}; \psi)$ in SIVI-SM are parameterized by MLPs with layer widths $[2, 128, 128, 2]$ and ReLU activation functions.

The noise levels in the diffusion bridge are $1 - \alpha(s_t) = 1 - t/5$ for $t \in \{0, 1, \cdots, 4\}$. We set the learning rate of variational parameters ϕ_t (or ϕ) to 0.001 and the learning rate of ψ_t (or ψ) to 0.002 in both SIVI and HSIVI. For HSIVI-LB and HSIVI-SM, we run 80000 variational parameter updates for every conditional layer; for SIVI-LB and SIVI-SM, we run $5\times$80000 variational parameter updates. For HSIVI-SM and SIVI-SM, in each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi_t)$ (or $\boldsymbol{f}(\boldsymbol{x}; \psi)$), we update ψ_t (or ψ) one time after each update of ϕ_t (or ϕ). All the algorithms are trained with a batch size of 64.

E.1.2 High-dimensional conditioned diffusion

For the experiment on high-dimensional conditioned diffusion, we examine the performances of SIVI and 5-layer HSIVI. The ground truth is formed by running 100,000 independent stochastic gradient Langevin dynamics (SGLD) chains with a step size of 0.0001 and collecting the results after 10,000 iterations. For $t = 0, \ldots, T-2$, the mean of each conditional layer $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu)$ in HSIVI has a residual form, i.e. $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu) = \boldsymbol{x}_{t+1} + \bar{\boldsymbol{\mu}}(\boldsymbol{x}_{t+1}; \phi_t^\mu)$. For SIVI and $t = T-1$ in HSIVI, we assume $\boldsymbol{\mu}(\boldsymbol{z}; \phi^\mu) = \bar{\boldsymbol{\mu}}(\boldsymbol{z}; \phi^\mu)$ and $\boldsymbol{\mu}(\boldsymbol{x}_{t+1}; \phi_t^\mu) = \bar{\boldsymbol{\mu}}(\boldsymbol{x}_{t+1}; \phi_t^\mu)$. For each t, $\bar{\boldsymbol{\mu}}_t(\boldsymbol{x}; \phi_t^\mu)$ in HSIVI and $\bar{\boldsymbol{\mu}}(\boldsymbol{z}; \phi^\mu)$ in SIVI are MLPs with layer widths $[300, 512, 512, 300]$ and ReLU activation functions. For each t, $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi_t)$ in HSIVI-SM and $\boldsymbol{f}(\boldsymbol{x}; \psi)$ in SIVI-SM are MLPs with layer widths $[300, 512, 512, 300]$ and ReLU activation functions. For both SIVI and HSIVI, we train each conditional layer for 100,000 iterations with a batch size of 128. For HSIVI-SM and SIVI-SM, in each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi_t)$ (or $\boldsymbol{f}(\boldsymbol{x}; \psi)$), we update ψ_t (or ψ) one time after each update of ϕ_t (or ϕ). We set the learning rate to be 0.0001 for ϕ_t (or ϕ) and 0.0005 for ψ_t (or ψ).

E.2 Diffusion model acceleration

In this part, we use the diffusion bridge to construct the auxiliary distributions and joint training as mentioned in Section 4.2. With a pre-trained score model or noise model, we consider the generative tasks as score-based variational inference problems. Therefore, we do not use any training data to train HSIVI-SM.

For HSIVI-SM, the variational prior $q_T(\boldsymbol{x}_T)$ is set to be $\mathcal{N}(\mathbf{0}, \mathbf{I})$. To avoid the large memory consumption, we use the joint training method where the parameters of the conditional layers $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi)$ and $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ are the same across different t. The t-th conditional layer is a conditional Gaussian distribution

$$q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi) = \mathcal{N}\left(\boldsymbol{x}_t; \boldsymbol{\mu}_t(\boldsymbol{x}_{t+1}; \phi^\mu), \mathrm{diag}\left(\sigma_t^2 \exp(\phi^\sigma)\right)\right),$$

where $\{\phi^\mu, \phi^\sigma\} = \phi$ are the variational parameters, ϕ^σ is a vector with the same dimension as $\boldsymbol{x}$, and σ_t is a fixed scalar value. We use the generalized inference process in DDIM (Song et al., 2020a) with the noise level $\eta > 0$ to initialize $\boldsymbol{\mu}_t(\boldsymbol{x}_{t+1}; \phi^\mu)$ and determine the value of σ_t for each t. If not otherwise specified, we use the Adam optimizer (Kingma & Ba, 2015) with $\beta = (0.9, 0.99)$ for training.

E.2.1 Toy examples of diffusion model acceleration

For pre-training the score model $\boldsymbol{S}^*(\boldsymbol{x}, s)$, we consider quadratic noise levels $1 - \alpha(s) = s^2$ for $s \in [0, 1]$. We then train $\boldsymbol{S}^*(\boldsymbol{x}, s)$ on 1000 fixed noise levels $\{1 - \alpha(i/1000)\}_{i=1}^{1000}$ by optimizing the DDPM loss in equation (13) for 200,000 iterations with a learning rate of 0.0003 and a batch size of 100. For constructing the diffusion bridge, we choose T discrete time steps $\{s_t\}_{t=0}^{T-1}$ so that $1 - \alpha(s_t) = [0.01 + (\sqrt{0.8} - 0.1)t/T]^2$ for $t = 0, 1, \ldots, T-1$.

Model architecture The model architecture of $\boldsymbol{S}^*(\boldsymbol{x}, s)$ is

$$\boldsymbol{S}^*(\boldsymbol{x}, s) = \mathrm{MLP}^{\mathrm{dec}}\left(\mathrm{MLP}^{\mathrm{embx}}(\boldsymbol{x}) + \mathrm{MLP}^{\mathrm{embt}}(1 - \alpha(s))\right),$$

where $\mathrm{MLP}^{\mathrm{dec}}$ is a decoder implemented as MLPs with layer widths $[128, 128, 128, 2]$, $\mathrm{MLP}^{\mathrm{embx}}$ is data embedding block implemented as MLPs with layer widths $[2, 128]$, and $\mathrm{MLP}^{\mathrm{embt}}$ is a time embedding block implemented as MLPs with layer widths $[256, 128, 128, 2]$. We use the sinusoidal positional embedding (Vaswani et al., 2017) of $1 - \alpha(s)$ as the input of $\mathrm{MLP}^{\mathrm{embt}}$. All these three MLPs use GELU as activation functions. We use the generalized inference process with noise level $\eta = 1.0$ to initialize the conditional layers. The architecture of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ is the same as that of $\boldsymbol{S}^*(\boldsymbol{x}, s)$. We initialize $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ with $\boldsymbol{S}_t^*(\boldsymbol{x}_t) := \boldsymbol{S}^*(\boldsymbol{x}_t, s_t)$.

Training setting The learning rate is set to be 0.0002 for $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi)$ and 0.0005 for $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ on Swissroll, Circles, and Moons for both $T = 5, 10$. On Checkerboard, the learning rate is set to be 0.00001 (0.00002) for $q_t(\cdot|\boldsymbol{x}_{t+1}; \phi)$ and 0.00005 (0.0001) for $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ when $T = 5$ ($T = 10$). We train HSIVI-SM for 25,000 iterations with a batch size of 64 in all cases. In each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$, we update ψ 3 times after each update of ϕ.

E.2.2 MNIST

For the experiment on MNIST, we use the pre-trained noise model $\epsilon^*(\boldsymbol{x}, s)$ and train HSIVI-SM with ϵ-training introduced in Section C.4. The following construction of noise schedule comes from Song et al. (2020a). Let $\beta_j = \beta_{\min} + \frac{\beta_{\max} - \beta_{\min}}{999} j$ for $j = 0, \ldots, 999$, where $\beta_{\min} = 0.0001, \beta_{\max} = 0.02$. We pre-train the noise model on the 1000 fixed noise levels $1 - \alpha(s) := \prod_{j=0}^{s} \beta_j$ for $s = 0, \ldots, 999$ by equation (14). The noise model is trained for 100,000 iterations with a learning rate of 0.0001 and a batch size of 64. We then choose T discrete time steps $s_t = \lfloor 800 \cdot \frac{t^2}{T^2} \rfloor$ for $t = 0, \ldots, T-1$ to construct the T-layer diffusion bridge.

Model architecture The pre-trained noise model $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ follows the UNet structure employed by Ho et al. (2020) where the number of input channels and output channels is reduced to one. Additionally, we pad the image size to 32×32 to fit $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$. We use the generalized inference process with noise level $\eta = 0.2$ to initialize the conditional layers. The architecture of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ is the same as that of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$. We initialize $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ with $-\boldsymbol{\epsilon}^*(\boldsymbol{x}_t, s_t)/\sqrt{1 - \alpha(s_t)}$.

Training setting For both $T = 5, 10$, the learning rate is set to be 1.6×10^{-5} for ϕ and 6.4×10^{-5} for ψ. We train HSIVI-SM for 10,000 iterations with a batch size of 64 in all cases. In each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$, we update ψ 20 times after each update of ϕ.

E.2.3 CIFAR-10 & CelebA

For experiments on CIFAR-10 and CelebA, we use the pre-trained noise model $\epsilon^*(\boldsymbol{x}, s)$ and train HSIVI-SM with ϵ-training introduced in Section C.4. We use the same noise schedule as in the experiment on MNIST. Let $\beta_j = \beta_{\min} + \frac{\beta_{\max} - \beta_{\min}}{999} j$ for $j = 0, \ldots, 999$, where $\beta_{\min} = 0.0001, \beta_{\max} = 0.02$. We pre-train the noise model on the 1000 fixed noise levels $1 - \alpha(s) := \prod_{j=0}^{s} \beta_j$ for $s = 0, \ldots, 999$ by optimizing equation (14). On CIFAR-10, the noise model is trained for 2160 epochs, with a learning rate of 0.0002 and batch size of 128; on CelebA, the noise model is trained for 600 epochs, with a learning rate of 0.00002 and batch size of 128. We then choose T discrete time steps $s_t = \lfloor 800 \cdot \frac{t^2}{T^2} \rfloor$ for $t = 0, \ldots, T-1$ to construct the T-layer diffusion bridge.

Model architecture On CIFAR-10, the structure of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ is exactly the UNet[2] employed in Ho et al. (2020) without modification; on CelebA, the structure of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ follows the UNet in Ho et al. (2020) but is reduced by one downsampling block and one upsampling block. Therefore, the structures of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$ are the same on CIFAR-10 and CelebA. We use the generalized inference process with noise level $\eta = 0.2$ to initialize the conditional layers. The architecture of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ is the same as that of $\boldsymbol{\epsilon}^*(\boldsymbol{x}, s)$. We initialize $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ with $-\boldsymbol{\epsilon}^*(\boldsymbol{x}_t, s_t)/\sqrt{1 - \alpha(s_t)}$.

Training setting The number of layers, which is also the number of function evaluations (NFE), is set to be $T = 5, 10, 15$ in our test cases. On CIFAR-10, the learning rate is set to be 1.6×10^{-5} for $q_t(\cdot|\boldsymbol{x}_{t+1}; \phi)$ and 8×10^{-5} for $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$; on CelebA, the learning rate is set to be 1.2×10^{-6} for $q_t(\cdot|\boldsymbol{x}_{t+1}; \phi)$ and 6×10^{-6} for $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$. We trained HSIVI-SM for 10,000 iterations with a batch size of 128. During each nested training loop of $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$, we update ψ 20 times after each update of ϕ, since we find $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$ needs more training empirically to provide reliable guidance. For $T = 10, 15$, we use the above training settings; for $T = 5$, we find that further fine-tuning on the well-trained 15-layer HSIVI-SM for 1,000 iterations yields better results, and we utilize this strategy to optimize the 5-layer HSIVI-SM with a $0.1\times$ smaller learning rate. Experiments need about 2.5 days on CIFAR-10 and need about 5 days on CelebA using 8 Nvidia 2080 Ti GPUs. During the training, we find that HSIVI-SM converges in the first 30% iterations on CIFAR-10 and converges in the first 50% iterations on CelebA.

F Limitations

For the application of accelerating the sampling process of diffusion models, our HSIVI-SM training involves three models: the score model (or noise model), the conditional layers $q_t(\boldsymbol{x}_t|\boldsymbol{x}_{t+1}; \phi)$, and $\boldsymbol{f}_t(\boldsymbol{x}_t; \psi)$. As a result, HSIVI-SM requires higher memory consumption due to the involvement of multiple models. Additionally, since our HSIVI algorithm approximates the target distribution using the score function, it necessitates a pre-trained score model (or noise model) with high accuracy and additional training steps. Finally, we recognize that the alternative method HSIVI-LB remains unexplored for accelerating the diffusion model, and we defer this aspect to future research.

[2]We use the Pytorch implementation of UNet structure in https://github.com/tqch/ddpm-torch.