WHERE TO DIFFUSE, HOW TO DIFFUSE AND HOW TO GET BACK: LEARNING IN MULTIVARIATE DIFFUSIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based generative models (DBGMs) perturb data to a target noise distribution and reverse this inference process to generate samples. The choice of inference diffusion affects both likelihoods and sample quality as it is tied to the generative model. Recent work in DBGMs has applied the principle of improving generative models with the use of auxiliary variables, leading to improved sample quality. While there are many such multivariate diffusions to explore, each new one requires significant model-specific analysis, hindering rapid prototyping and evaluation. In this work, we study Multivariate Diffusion Models (MDMs). First, for any number of auxiliary variables, we provide a recipe for maximizing a lower-bound on the MDMs likelihood, without requiring any model-specific analysis. Next, we demonstrate how to parameterize the diffusion for a specified target noise distribution; these two points together enable optimizing the inference diffusion process. Optimizing the diffusion expands easy experimentation from just a few well-known processes to an automatic search over the set of linear diffusions. To demonstrate these ideas, we introduce two new specific diffusions as well as learn a diffusion process on the MNIST and CIFAR10 datasets. We achieve improved bits-per-dim bounds using the new diffusion, compared to the existing likelihood-trained VPSDE. We additionally connect the existing cld objective to the likelihood lower bound.

1 INTRODUCTION

Diffusion-based generative models (DBGMs) perturb data to a target noise distribution and reverse this inference process to generate samples. They have achieved impressive performance in image generation, editing, translation (Dhariwal & Nichol, 2021; Nichol & Dhariwal, 2021; Sasaki et al., 2021; Ho et al., 2022), conditional text-to-image tasks (Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022) and music and audio generation. (Chen et al., 2020; Kong et al., 2020; Mittal et al., 2021). They are trained for likelihood using a lower-bound on their likelihood featuring a variational process, often interpreted as gradually "noising" the data (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020b).

The choice of inference process affects both likelihoods and sample quality as it is tied to the generative model; on different datasets and models different process work better; there is no universal best choice of inference, and the choice matters (Song et al., 2020b). A given combination of model, inference, and choice of loss can face challenges such as unbounded gradients (Kim et al., 2021), mismatch of inference with the model prior, and large number of steps required for sampling (Zheng et al., 2022).

So how can diffusions be improved? A general principle in probabilistic modeling and inference is using auxiliary variables: HMC improves MCMC with Hamiltonian dynamics (Neal et al., 2011); Hierarchical variational inference (Ranganath et al., 2016) and hierarchical variational auto-encoders (Sønderby et al., 2016; Roberts et al., 2017; Hsu et al., 2018; Maaløe et al., 2019; Klushyn et al., 2019; Vahdat & Kautz, 2020; Child, 2020) marginalize auxiliary to build expressive inference distributions.

Dupont et al. (2019) shows that operating on an augmented state space in neural ordinary differential equations (ODEs) allows a richer set of functions to modeled. (Huang et al., 2020) use this principle

for learning normalizing flows, where the authors maximize a joint likelihood $p_\theta(\mathbf{x}, \mathbf{e})$ where $\mathbf{x} \sim q_{\text{data}}$ and $\mathbf{e} \sim \mathcal{N}(0, I_d)$ to model richer distributions over $\mathbf{x}$.

Recent work in DBGMs has applied this principle to diffusions, leading to both stable training and improved sample quality. Based on connections with statistical mechanics, Dockhorn et al. (2021) introduce the critically-damped langevin diffusion (CLD) for diffusion-based generative modeling. CLD diffuses an extra "velocity" variable together with each data dimension in an extended space. Empirically, CLD-based generative models have smoother scores leading to faster sampling, improved sampling quality and improved likelihood bounds. The extended space also allows them to choose dynamics that circumvent certain issues with unbounded gradients in score matching.

While there are many such auxiliary variable models to explore, models like CLD are evidence that each new process requires significant model-specific analysis. Deriving the likelihood lower-bound, also called the evidence lower bound (ELBO), for diffusions is challenging enough because models and inference are both specified in continuous time (Huang et al., 2021; Kingma et al., 2021; Durkan & Song, 2021). The derivations are typically carried out in a case-by-case manner. Auxiliary variables complicate this process further, for example computing the time-varying distribution of noise given data necessitates solving matrix Lyupanov equations. Deriving the stationary distribution of the inference process, which helps the model and inference match can be intractable. These challenges limit rapid prototyping and evaluation of new auxiliary variable diffusions.

Concretely, to train a diffusion model, possibly with auxiliary variables, one needs to:

(A1) Select a compatible inference-model process pair

(A2) Derive the ELBO for this pair

(A3) Sample the time-varying noise distribution and approximate the ELBO's gradients

In this work, we study Multivariate Diffusion Models (MDMs). MDMs extend the inference process for training diffusions from specific instantiations–like the variance-preserving stochastic differential equation (VPSDE) and CLD—to the set of linear diffusions with auxiliary variables.

First, for any number of auxiliary variables, we derive the likelihood lower bound (A2). Second, without requiring any model-specific analysis, we show that the transition kernels of linear MDMs can be computed automatically and generically, and scaled to higher-order auxiliary systems, without incurring significant cost (A3).

We then provide a parameterization of inference processes that can converge to a chosen model prior (A1); this tightens the likelihood lower-bound. We develop the parameterization by making a connection between diffusions models and the design of gradient-based MCMC samplers (Ma et al., 2015). The parameterization spans the set of all diffusions with a given stationary distribution.

With an ELBO, tractable kernels, and inference process parameterization, we can explore any number of linear MDMs. However as mentioned in Song et al. (2020b), the choice of the inference process is crucial and can vary based on the choice of the dataset and model.

The automatic transition kernels and fixed stationary distributions facilitate directly learning the inference process to maximize the MDM ELBO. This turns diffusion model training into a search not only over score models but also into an automatic search over the entire set of linear diffusions. This is at no extra mathematical or algorithmic cost, aside what it handled by auto-differentiation for some additional parameters. This extends the learning in Kingma et al. (2021) beyond VPSDE to diffusions with auxiliary variables and extends the successes of CLD to a larger class of models.

To demonstrate these ideas, we introduce two new specific linear diffusions, accelerated Langevin diffusion (ALDA) and modified accelerated Langevin diffusion (MALDA), as well learn a multivariate diffusion process, on the MNIST and CIFAR10 datasets. We study the effect of these inference processes on DBGM training without needing to derive their sampling algorithms and objectives. Using MALDA and the learned SDE, we achieve impressive bits-per-dim bounds, compared to the existing likelihood-trained VPSDE and CLD on the CIFAR-10 and MNIST datasets.

Most importantly, the MDM framework makes it simple for practitioners and researchers to extend beyond the diffusions we explore in this work and those known in the literature.

2 SETUP

We describe diffusions similarly to Sohl-Dickstein et al. (2015); Huang et al. (2021); Kingma et al. (2021); rather than starting with a noising (inference) process, and using time reversal to specify a model, we start with a model and then pick inference for which an ELBO can be derived. We use $\mathbf{z}_t$ to describe the process under the generative model p_θ using time variable t and $\mathbf{y}_s$ for the process under inference distribution q_ϕ using time variable s.

Model. Let $\mathbf{B}_t$ be a Brownian motion and let $\mathbf{z}_t \in \mathbb{R}^d$ be a continuous-time stochastic process evolving according to:

$$d\mathbf{z} = h_\theta(\mathbf{z}, t)dt + \beta_\theta(\mathbf{z}, t)d\mathbf{B}_t. \tag{1}$$

Diffusions sample a model prior $\mathbf{z}_0 \sim \pi_\theta$ and then integrate eq. (1) to approximate the data $\mathbf{x} \sim q_{\text{data}}$ with $\mathbf{z}_T \sim p_\theta$[1].

Inference. MLE training of diffusion models is intractable because computing their density $p_\theta(\mathbf{z}_T)$ requires integrating over all possible Brownian paths (Huang et al., 2021; Durkan & Song, 2021; Kingma et al., 2021). Instead, one can introduce an inference process $q_\phi(\mathbf{y}_s|\mathbf{x} = x)$ for each data value x and time s so that p_θ can be trained using a variational lower bound on the log likelihood. The inference is chosen so that $\lim_{t\to 0} q_\phi(\mathbf{y}_t = x|x) = 1$. Like the model, the inference process is represented as a diffusion:

$$d\mathbf{y} = f_\phi(\mathbf{y}, s)ds + g_\phi(\mathbf{y}, s)d\widehat{\mathbf{B}}_s, \tag{2}$$

where $\widehat{\mathbf{B}}_s$ is another Brownian motion independent of $\mathbf{B}_t$. The inference process is usually taken to be specified rather than learned, and chosen to be i.i.d. for each $\mathbf{z}_{tj}$ conditional on each $\mathbf{x}_j$. This leads to the interpretation of the $\mathbf{z}_{tj}$ as noisy versions of features $\mathbf{x}_j$ (Ho et al., 2020).

Likelihood Lower Bound. How does one derive an elbo for models and inferences based on diffusion processes? As shown in Huang et al. (2021) and in appendix E, this relies on a particular likelihood ratio of the stochastic processes. The ratio can be computed by setting the generative processes' parameters as

$$h_\theta(\cdot, t) = g_\phi^2(T-t)s_\theta(\cdot, T-t) - f_\phi(\cdot, T-t), \qquad \beta_\theta(t) = g_\phi(T-t), \tag{3}$$

for underlying *score model* s_θ[2]. Then, by Huang et al. (2021); Durkan & Song (2021), the ELBO can be written as $\log p_\theta(x) \geq \mathcal{L}^{\text{ism}}$ with:

$$\mathcal{L}^{\text{ism}}(x) = \mathbb{E}_{q_\phi(\mathbf{y}|x)}\left[\log \pi_\theta(\mathbf{y}_T) + \int_0^T -\frac{1}{2}||s_\theta||^2_{g_\phi^2} - \nabla\cdot(g_\phi^2 s_\theta - f_\phi)ds\right], \tag{4}$$

where f_ϕ, g_ϕ, s_θ are evaluated at $(\mathbf{y}_s, s)$, and $||x||^2_A = x^\top Ax$. This features the Implicit Score Matching (ISM) loss plus terms that vary with π_θ and q_ϕ (Song et al., 2020a). The ISM ELBO can also be written as an alternate ELBO based on the Denoising Score Matching (DSM) loss (Vincent, 2011; Song et al., 2020b). We show a multivariate version of this in eq. (13).

3 MULTIVARIATE DIFFUSION MODELS

Auxiliary variables have been used to improve generative models in MCMC (Neal et al., 2011), Hierarchical Variational Inference (Ranganath et al., 2016), ladder variational autoencoder (Sønderby et al., 2016), among others. CLD (Dockhorn et al., 2021) explore this for a Langevin diffusion with one auxiliary "velocity" variable and show CLD requires fewer function evaluation to sample from the generative process ,and the CLD score function is smoother compared to VPSDE's. Beyond CLD,

[1]Following Huang et al. (2021); Dockhorn et al. (2021) we integrate all processes in forward time 0 to T.

[2]Though this looks like it ties the model and inference, and corresponds to the time-reversal results (Anderson, 1982), this model class is equally expressive to that where the drift h_θ does not use f_ϕ, g_ϕ because s_θ can be any function. However, the model and inference must share a diffusion coefficient (appendix E).

augmenting diffusions is not common because of apparent challenges: it is not immediately clear how to derive the necessary components of training DBGMs for multivariate diffusions.

In this work, we explore this question and extend the set of diffusions that can be used in diffusion objectives from CLD to any linear multivariate diffusion with auxiliary variables. This set also includes common scalar diffusions such as the Variance Preserving (VP) process (Song et al., 2020b).

At high level, MDMs pair $\mathbf{z} \in \mathbb{R}^d$ with auxiliary variables $\mathbf{v} \in \mathbb{R}^{d(K-1)}$ and diffuse in the extended space:

$$d\mathbf{u}_t = \begin{pmatrix} d\mathbf{z}_t \\ d\mathbf{v}_t \end{pmatrix} = h_\theta(\mathbf{z}_t, \mathbf{v}_t, t)dt + \beta_\theta(\mathbf{z}_t, \mathbf{v}_t, t)d\begin{pmatrix} \mathbf{B}_t^z \\ \mathbf{B}_t^v \end{pmatrix}.$$

We proceed by deriving the MDM ELBO and providing a set of tools to work with it in practice.

3.1 Multivariate Model and Inference

The generative process p_θ and inference process q_ϕ are now over

$$\mathbf{u}_t = [\mathbf{u}_t^z, \mathbf{u}_t^v] = [\mathbf{z}_t, \mathbf{v}_t^1, \dots, \mathbf{v}_t^{K-1}], \qquad \mathbf{y}_s = [\mathbf{y}_s^z, \mathbf{y}_s^v] = [\mathbf{y}_s^z, \mathbf{y}_s^{v_1}, \dots, \mathbf{y}_s^{v_{K-1}}]. \tag{5}$$

For each data feature $\mathbf{x}_j$, each $\mathbf{u}_{tj} \in \mathbb{R}^K$ and $\mathbf{y}_{sj} \in \mathbb{R}^K$ now consists of a primary "data" dimension (superscripts z) and $K-1$ auxiliary variables (supserscripts v). There is a set of these variables for each data dimension j, so $\mathbf{u}, \mathbf{y} \in \mathbb{R}^{dK}$. To specify MDMs, we extend the drift coefficient from a function in $\mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ to the extended space $\mathbb{R}^{dK} \times \mathbb{R}_+ \to \mathbb{R}^{dK}$. This function lets the z and v coordinates of $\mathbf{u}$ interact. We likewise extend the diffusion coefficient to a matrix β_θ acting on brownian motion $\mathbf{B}_t \in \mathbb{R}^{dK}$. We consider MDMs specified by:

$$\mathbf{u}_0 \sim \pi_\theta, \qquad d\mathbf{u} = h_\theta(\mathbf{u}, t)dt + \beta_\theta(t)d\mathbf{B}_t. \tag{6}$$

MDMs model the data $\mathbf{x}$ with $\mathbf{z}_T$, a coordinate in $\mathbf{u}_T \sim p_\theta$.

Because the MDM model is over the extended space, the inference distribution must be too. We then add an additional initial distribution over the auxiliary variable $\mathbf{y}_0^v$ at time 0, which has an important role in section 3.2. We set $q(\mathbf{y}_0^v|x)$ to be any chosen initial distribution, e.g. $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and discuss this choice in section 4. Then $\mathbf{y}_s$ evolves according to

$$d\mathbf{y} = f_\phi(\mathbf{y}, s)ds + g_\phi(s)d\widehat{\mathbf{B}}_s, \tag{7}$$

where the inference drift and diffusion coefficients f_ϕ, g_ϕ are now over the extended space.

3.2 ELBO for MDMs

We now show how to train MDMs. We derive two lower bounds on the MDM data marginal log likelihood. We sketch a roadmap and show the result here, and include details in appendix E.

We start by reparameterizing the model drift $h_\theta(\mathbf{u}, t)$ and diffusion $\beta_\theta(t)$ as in the scalar case in eq. (3). Then, the multivariate ELBO can be derived straightforwardly by starting with the bound in eq. (4) for $\log p_\theta(\mathbf{x})$, applying it to the joint space $p_\theta([\mathbf{u}_T^z, \mathbf{u}_T^v])$, and then marginalizing $\mathbf{u}_T^v$ under the time-zero inference distribution $q_\phi(\mathbf{y}_0^v|x)$.

First, by applying diffusion models to $\mathbf{u}_T$ instead of $\mathbf{z}_t$, the ISM bound is:

$$\mathcal{L}^{ism}(\mathbf{u}_T = [x, v]) = \mathbb{E}_{q_\phi(\mathbf{y}|\mathbf{y}_0=[x,v])}\left[\underbrace{\log \pi_\theta(\mathbf{y}_T)}_{\ell_T} + \int_0^T \underbrace{-\frac{1}{2}||s_\theta||^2_{g_\phi^2} - \nabla \cdot (g_\phi^2 s_\theta - f_\phi)}_{\ell^{\text{ism}}} ds\right], \tag{8}$$

where $g^2(s)$ denotes $g(s)g(s)^\top$. By the diffusion ELBO derivation, the bound satisfies:

$$\log p_\theta(\mathbf{u}_T = [x, v]) \geq \mathcal{L}^{\text{ism}}(\mathbf{u}_T = [x, v]). \tag{9}$$

Next, for any joint, for any q_ϕ such that p_θ/q is finite, we can appeal to Jensen's inequality and change of measure to lower-bound the log marginal:

$$\log p_\theta(\mathbf{u}_T^z = x) \geq \mathbb{E}_{q_\phi(\mathbf{y}_0^v = v|x)}\Big[\log p_\theta(\mathbf{u}_T^z = x, \mathbf{u}_T^v = v) - \log q_\phi(\mathbf{y}_0^v = v|x)\Big]. \tag{10}$$

We can therefore use $q(\mathbf{y}_0^v|x)$ together with $\mathcal{L}^{\text{ism}}(\mathbf{u}_T)$ to lower-bound the data log marginal likelihood:

$$\log p_\theta(\mathbf{u}_T^z = x) \geq \mathbb{E}_{q_\phi(\mathbf{y}_0^v = v|x)}\Big[\mathcal{L}^{\text{ism}}(\mathbf{u}_T = [x, v]) \underbrace{- \log q_\phi(\mathbf{y}_0^v = v)}_{\ell_q}\Big]. \tag{11}$$

The ISM ELBO for MDMs, (derived in appendix E) can then be written as $\log p_\theta(x) \geq \mathcal{L}^{\text{mdm-ism}}$, with

$$\mathcal{L}^{\text{mism}} = \mathbb{E}_q\left[\mathbb{E}_{s\sim\text{Unif}(0,T)}\left[\ell_s^{(ism)} \cdot T\right] + \ell_T + \ell_q\right]. \tag{12}$$

Like in the scalar case, the MDM ISM ELBO can be transformed into an equal MDM DSM ELBO $\mathcal{L}^{\text{mdsm}} = \mathcal{L}^{\text{mism}}$. Let $\nabla \log q_{s|0}$ denote either $\nabla_{\mathbf{y}_s} \log q(\mathbf{y}_s|\mathbf{y}_0)$ or $\nabla_{\mathbf{y}_s} \log q(\mathbf{y}_s|x)$, and define:

$$\ell_s^{(dsm)} = \frac{1}{2}\big|\big|\nabla \log q_{s|0}\big|\big|_{g_\phi}^2 - \frac{1}{2}\big|\big|s_\theta - \nabla \log q_{s|0}\big|\big|_{g_\phi}^2 + \nabla_{\mathbf{y}_s} \cdot f_\phi. \tag{13}$$

The MDM DSM bound $\mathcal{L}^{\text{mdsm}}$ takes the same form as $\mathcal{L}^{\text{mism}}$ but with $\ell_s^{(dsm)}$ in place of $\ell_s^{(ism)}$ (appendix F). We now provide 3 ingredients for tightening and optimizing these bounds in a generic fashion.

3.3 INGREDIENT 1: COMPUTING THE TRANSITION $q_\phi(\mathbf{y}_s|\mathbf{y}_0)$

To compute the ELBOs, we need two things from the transition kernel, (a) samples $\mathbf{y}_s|\mathbf{y}_0$ and computing the transition kernel terms in eqs. (12) and (13). For linear MDMs, we can sample and compute transitions in closed form without requiring any diffusion specific analysis.

For linear diffusions, the transition kernel $q(\mathbf{u}_s|\mathbf{u}_0)$ is always Gaussian (Särkkä & Solin, 2019). Therefore, we just find the mean $\mathbf{m}_{s|0}$ and covariance $\mathbf{\Sigma}_{s|0}$ of $q(\mathbf{y}_s|\mathbf{y}_0)$. The mean and covariance are solutions to ODEs discussed in appendix C. Let $f(\mathbf{y}, s) = \mathbf{A}(s)\mathbf{y}$. The mean is:

$$\mathbf{m}_{s|0} = \exp\left[\int_0^s \mathbf{A}(\nu)d\nu\right]\mathbf{y}_0 \underbrace{= \exp(s\mathbf{A})\mathbf{y}_0}_{\text{no integration if } \mathbf{A}(\nu) = \mathbf{A}}, \tag{14}$$

where exp denotes matrix exponential. We will express the covariance of $q(\mathbf{y}_s|\mathbf{y}_0)$ as $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s{\mathbf{H}_s}^{-1}$ where, letting $I_s[\mathbf{A}] = \int_0^s \mathbf{A}(\nu)d\nu$,

$$\begin{pmatrix}\mathbf{C}_s\\ \mathbf{H}_s\end{pmatrix} = \exp\left[\begin{pmatrix} I_s[\mathbf{A}] & I_s[g^2]\\ \mathbf{0} & -I_s[\mathbf{A}^\top]\end{pmatrix}\right]\begin{pmatrix}\mathbf{\Sigma}_0\\ \mathbf{I}\end{pmatrix} \underbrace{= \exp\left[s\begin{pmatrix}\mathbf{A} & g^2\\ \mathbf{0} & -\mathbf{A}^\top\end{pmatrix}\right]\begin{pmatrix}\mathbf{\Sigma}_0\\ \mathbf{I}\end{pmatrix}}_{\text{no integration if } \mathbf{A}(\nu) = \mathbf{A},\, g(\nu) = g}. \tag{15}$$

Conditioning on $\mathbf{u}_0 = (x, v)$ amounts to setting $\mathbf{\Sigma}_0 = \mathbf{0}$. Finally, $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s{\mathbf{H}_s}^{-1}$.

Hybrid Score Matching. Instead of computing $q(\mathbf{y}_s|\mathbf{y}_0)$, we can apply the hybrid score matching principle (Dockhorn et al., 2021) to reduce variance by compute objectives using $q(\mathbf{y}_s|x)$ instead of $q(\mathbf{y}_s|\mathbf{y}_0)$, which amounts to integrate out $\mathbf{v}_0$. Formally, we show the equality of the ELBO with $q(\mathbf{y}_s|\mathbf{y}_0)$ swapped-out for $q(\mathbf{y}_s|x)$ in appendix F. To accomplish this, we simpy replace $\mathbf{y}_0$ with $[x, \mathbb{E}[\mathbf{v}_0]]$ in the expression for $\mathbf{m}_{s|0}$, and for $\mathbf{\Sigma}_{s|0}$, we replace the choice of $\mathbf{0}$ for $\mathbf{\Sigma}_0$ instead with a block matrix featuring $\mathbf{\Sigma}_{0,zz} = 0$, and $\mathbf{\Sigma}_{0,zv}, \mathbf{\Sigma}_{0,vv}$ set to the chosen distribution for $q_\phi(\mathbf{y}_0^v|\mathbf{y}_0^z)$.

A fast and simple algorithm. Instead of solving the mean and covariance ODEs analytically like CLD (see for instance pages 50-54 of Dockhorn et al. (2021)), we provide a simple algorithm to compute the mean and covariance which only require knowing the functions f, g, see algorithm 1 in appendix H. This makes the ELBO tractable and easy, i.e. skips both analytic derivations and numerical integration during training.

Table 1: Runtime Comparison: we compare the run time of sampling from the CLD diffusion analytically versus using our automated algorithm.

Method	CIFAR-10	MNIST
Analytical	0.027	0.0062
Automated	0.029	0.007

For scalar diffusions, the drift A and diffusion g coefficients are scalars. For MDMs, $\mathbf{A}, g$ are $K \times K$. Similar to scalar diffusions, these parameters are shared across coordinates, where each matrix acts on a K-dimensional vector $[(\mathbf{y}^z)_j, (\mathbf{y}^{v_1})_j, \dots (\mathbf{y}^{v_{K-1}})_j]$ for $j \in \{1, \dots, d\}$. This means that the matrix operations requiring matrix exponentials and inverses are done on $K \times K$ matrices. In table 1, we compare the time to sample a batch of size 256 from the transition kernel for the CIFAR10 and MNIST datasets. The table shows the extra computational cost of the automated algorithm is negligible.

We have accomplished tractable transition computation by restricting to linear processes, which include most diffusions used in the mainstream literature (VP, sub-VP, Variance Exploding (VE), CLD) and even extends to the general multivariate case for any number of auxiliary variables.

3.4 Ingredient 2: MDM Parameterization

The MDM ELBO eq. (8) is higher when the inference distribution tends toward the model's prior. Thus in ingredient 2, we construct inference processes with the model prior π_θ as their stationary distribution q_∞. Ma et al. (2015) provide a complete recipe for constructing gradient-based MCMC samplers; they span all Markov diffusion processes with a given stationary distribution for the general non-linear case, which we extend to the time-varying processes.

For the linear case, the parameterization simplifies to

$$d\mathbf{y} = \underbrace{-[\mathbf{Q}(s) + \mathbf{D}(s)]\nabla \mathbf{H}(\mathbf{y})}_{f} ds + \underbrace{\sqrt{2\mathbf{D}(s)}}_{g} d\widehat{\mathbf{B}}_s, \tag{16}$$

where $\mathbf{Q}$ is skew-symmetric ($-\mathbf{Q} = \mathbf{Q}^\top$), $\mathbf{D}$ is positive semi-definite and $\mathbf{H}(\mathbf{y}) = -\log q_\infty(\mathbf{y})$. In this work, we choose linear processes meaning $\nabla \mathbf{H}$ must be linear and therefore q_∞ must be Gaussian. Therefore, eq. (16) spans all linear processes with a stationary distribution, and we can compute automatic transition kernels for these processes.

In appendix C, we show the transition kernel computation in the $\mathbf{Q}/\mathbf{D}$ parameterization, with an algorithm in Appendix H. We show the ELBO in terms of $\mathbf{Q}/\mathbf{D}$ in appendix G.

3.5 Ingredient 3: Learning the inference process

Song et al. (2020b) show that the choice of the optimal inference process varies for different datasets and score models. However, training DBGMs requires computing the transition kernel and being able to parameterize diffusions processes. This prevented training DBGMs in the multivariate setting, let alone the inference process.

We observe that the ELBOs in eq. (12) and eq. (13) have no requirement for a fixed inference, and propose to learn the inference process jointly with the model in multivariate MDMs. Under the linear transitions (ingredient 1) and stationary parameterization (ingredient 2), learning the inference comes at no cost; we can do so without the stationary distribution going awry, and because transitions are computed generically, no algorithmic details change when learning. Learning the infernece alongside the model is well-motivated in previous literature as in variational auto-encoders, where the encoder (inference) and decoder (model) are learned jointly (Kingma & Welling, 2013).

We can analyze how learning the inference process is expected to help diffusion ELBOs specifically. First, $q_{\phi,\infty}$ may be set to equal π_θ, but it is $q_{\phi,T}$ for the chosen T that is featured in the ELBO. Learning the inference will reduce the cross-entropy:

$$-\mathbb{E}_{q_\phi(\mathbf{y}_T|x)}[\log \pi_\theta(\mathbf{y}_T)]. \tag{17}$$

Minimizing this term will tighten the ELBO for any score model. Next, q_ϕ is featured in the remaining DSM and ISM terms; optimizing for q_ϕ in those terms will tighten and improve the ELBO alongside the score model. Finally, q_ϕ is featured in the expectations and the $-\log q_\phi$ term:

$$\log p_\theta(\mathbf{u}_T^z = x) \geq \underbrace{\mathbb{E}_{q_\phi(\mathbf{y}_0^v = v|x)}}\left[\mathcal{L}^{\text{ism}}(\mathbf{u}_T = [x, v]) \underbrace{- \log q_\phi(\mathbf{y}_0^v = v)}\right] \tag{18}$$

These terms impose an optimality condition that $p_\theta(\mathbf{u}_T^v|\mathbf{u}_T^z) = q_\phi(\mathbf{y}_0^v|\mathbf{y}_0^z)$ (appendix D), similar to the role of $-\log q$ in the traditional ELBO (Blei et al., 2017). When it is satisfied, no looseness in the ELBO for *the marginal likelihood of just the data* x is due to the initial time zero auxiliary variables.

The only change is that $\mathbf{Q}, \mathbf{D}$ need to be specified with parameters that enable gradients, and we have to use re-parameterized samples $\mathbf{y}_s|\mathbf{y}_0$. For parameterization,

- Any skew-symmetric matrix $\mathbf{Q}$ can be written as $\mathbf{Q} = \mathbf{G} - \mathbf{G}^\top$ for some $\mathbf{G}_\phi$; therefore we can set $\mathbf{Q}_\phi(s) = \beta(s) \cdot \left[\mathbf{G}_\phi - \mathbf{G}_\phi^\top\right]$ for an unconstrained parameter $\mathbf{G}_\phi$
- For any positive semi-definite matrix $\mathbf{D}$, there exists a matrix $\mathbf{B}$ such that $\mathbf{D} = \mathbf{B}\mathbf{B}^\top$; therefore we can set $\mathbf{D}_\phi(s) = \beta(s) \cdot \left[\mathbf{B}_\phi\mathbf{B}_\phi^\top\right]$ for unconstrained $\mathbf{B}_\phi$

We provide more details on parameterization in appendix B.

4 Insights into Multivariate Diffusions

Scalar versus Multivariate Processes. The parameterization in eq. (16) clarifies what can change and what cannot to preserve the stationary distribution. Recall that $\mathbf{Q}$ and $\mathbf{D}$ are each $K \times K$ where there are $K - 1$ auxiliary variables. Because 0 is the only 1×1 skew-symmetric matrix, this means that scalar processes must set $\mathbf{Q} = 0$. With a standard normal stationary distribution, the resulting process is:

$$d\mathbf{y} = -\mathbf{D}(s)\mathbf{y}ds + \sqrt{2\mathbf{D}(s)}d\widehat{\mathbf{B}}_s. \tag{19}$$

What is left is precisely the VP process used widely in diffusion models where $\mathbf{D}(s) = \frac{1}{2}\beta(s)$ (Song et al., 2020b). This reveals that the VP process is the only[3] scalar diffusion with a stationary distribution. This also clarifies the role of $\mathbf{Q}$: it accounts for mixing between dimensions in multivariate processes, as do non-diagonal entries in $\mathbf{D}$.

CLD optimizes a lower bound on the data log-likelihood. The initial distribution $q(\mathbf{y}_0^v|x)$ appears in several places in MDM elbos. Looking at eqs. (12) and (13), we establish that the CLD objective is an ELBO without the $-\log q$. Because their q_ϕ is constant, their objective is formally maximizing a lower bound on the log marginal likelihood of the data, not just the extended space.

Combining MDMs and hybrid score matching. The choice in CLD to set $q(\mathbf{y}_0^v|x) = q(\mathbf{y}_0^v)$ has certain good properties when combined with CLD's $\mathbf{D}$ matrix and hybrid score matching. The matrix has a 0 in the top-left coordinate which means that DSM does not match scores of the main coordinates. Becaue hybrid score matching samples $q(\mathbf{y}_s|x)$ rather than $q(\mathbf{y}_s|\mathbf{y}_0)$, the score-matching gradients at time zero are bounded. The new SDEs that we explore in the experiments share these properties, and the principle is useful for designing new and learned linear MDMs in general.

Does my model use auxiliary variables? In section 3 we gave the example choice of $q(\mathbf{y}_0^v|x) = q(\mathbf{y}_0^v) = \mathcal{N}(0, 1)$ coordinate-wise. It is also a common choice to set π_θ as independent standard Gaussians for all variables. Because the optimum in diffusion models is $p_\theta = q$ for all t, we see a peculiar phenomenon under this choice: the model has main and auxiliary dimensions independent at both endpoints 0 and T. Does this mean that the model does not use auxiliary variables? In appendix A, we show that even when $q_\phi(\mathbf{y}_0)$ and π_θ have main and auxiliary variables independent at the end points, the model uses the auxiliary variables in the intermediate time. A sufficient condition is $\mathbf{Q}+\mathbf{D}$ is non-diagonal. On the other hand, when $\mathbf{Q} = \mathbf{0}$ and $\mathbf{D}$ is diagonal, under the independent endpoints, the model will not use the auxiliary variables to model the primary coordinate $\mathbf{u}_T^z$.

Existing diffusion processes. We show in appendix B.4 that existing diffusion processes such as VPSDE and CLD can also fit in the $\mathbf{Q}/\mathbf{D}$ parameterization. In appendix B.4, we show that the error incurred in computing the transition kernel's mean and covariance is negligible.

[3]There are processes such as sub-VP (Song et al., 2020b) which are covered in the sense that they tend to members of this parameterization as T is large enough: sub-VP converges to VP.

Model	# of variables	# of params	CIFAR-10	# of params	MNIST
CLD	2	35.7 million	≤ 3.563	1.5 million	≤ 0.9200
VPSDE	1	35.7 million	≤ 3.36	1.5 million	≤ 1.0890
VPSDE*	1	108 million	≤ 2.92	-	-
Learned	2	35.7 million	≤ 3.009	1.5 million	≤ 0.9600
ALDA	3	35.7 million	≤ 4.700	1.5 million	≤ 2.0610
MALDA	3	35.7 million	$\leq$ **2.634**	1.5 million	$\leq$ **0.5069**

Table 2: BPD on CIFAR-10 and MNIST. * indicates numbers from Durkan & Song (2021)

5 Experiments

We test the MDM framework by testing two new specific diffusions and a learned diffusion. The two specific diffusions are (a) ALDA, introduced in Mou et al. (2019) as a scheme for accelerated gradient-based MCMC sampling (eq. (26)) and (b) MALDA: a modified version of ALDA (eq. (27)). Both of the specific diffusions have two auxiliary variables, see appendix B.4 for details. We compare the performance of these three diffusions against VPSDE and ELBO-trained CLD.

Following Song et al. (2020b); Dockhorn et al. (2021); Huang et al. (2021) we train DBGMs on the CIFAR-10 and MNIST datasets for image generation. For both datasets, we use the modified U-Net architecture used in Ho et al. (2020); Huang et al. (2021). We input the auxiliary variables as extra channels, which leads to a negligible increase in parameter count. We use simple uniform dequantization; we do not tighten the bounds with variational dequantization or IWAE (Burda et al., 2015) for evaluation.

Table 2 shows that the inference process matters. Despite similar parameter counts for the score model, MALDA and the learned inference processes are able to achieve significantly lower bits-per-dim compared to diffusions models with a similar parameter count.

In table 3, we show that our diffusion processes are able to achieve close to start of the art bits-per-dim for both CIFAR10 and MNIST, with a significantly smaller number of parameters. In fig. 1 samples we plot the generated samples from CIFAR10 from the different DBGMs we have introduced.

Figure 1: CIFAR10 samples generated from the learned and MALDA generative model.

6 Related Work

We now discuss several connections not already emphasized in our discussion of closely-related works such as Huang et al. (2021) and CLD (Dockhorn et al., 2021).

Model	CIFAR-10	MNIST
FFJORD Grathwohl et al. (2018)	3.40	0.99
Flow++ Ho et al. (2019)	3.08	-
Gated PixelCNN Van den Oord et al. (2016)	3.03	-
CLD Dockhorn et al. (2021)	$\leq 3.563^*$	$\leq 0.9200^*$
VPSDE Durkan & Song (2021)	≤ 2.92 (3.36^*)	$\leq 1.0890^*$
Sparse Transformer Child et al. (2019)	2.83	-
Learned (2 vars)	≤ 3.009	≤ 0.9600
ALDA (3 vars)	≤ 4.700	≤ 2.0610
MALDA (3 vars)	$\leq$ **2.634**	$\leq$ **0.5069**

Table 3: NLL on CIFAR-10 and MNIST. * indicates evaluations on baseline models we train.

A trend in generative models and in inference algorithms is to augment them with auxiliary variables. For example, (Neal et al., 2011) improves MCMC with Hamiltonian dynamics. (Ranganath et al., 2016; Sønderby et al., 2016) improve variational inference with hierarchy and hierarchical variational auto-encoders (Sønderby et al., 2016; Roberts et al., 2017; Hsu et al., 2018; Maaløe et al., 2019; Klushyn et al., 2019; Vahdat & Kautz, 2020; Child, 2020). We apply this principle and diffuse in an augmeneted space.

One motivation for MDMs is to enrich the class of distributions that diffusions model in practice. Song et al. (2020b) show that for any DBGM, there is a deterministic neural ODE (Chen et al., 2018; Grathwohl et al., 2018), called the probability flow ODE, whose trajectories match the diffusion's. Dupont et al. (2019) prove that there exist functions that neural ODEs cannot represent. However, augmenting the state space of neural ODEs increases the set of functions that can be represented marginally on the original dimension of interest. They demonstrate that smoother transformations due to the use of augmented variables led to fewer function evaluations for integrating and stable training. Huang et al. (2020) use this to improve normalizing flows by applying the flow in an extended space. The work we present on diffusions with auxiliary variables can be seen as an application of this principle to diffusion based models, or equivalently, continuous normalizing flows.

Durkan & Song (2021); Huang et al. (2021) derive the ISM and DSM lower-bounds on the model log likelihood. Our work extends their analysis to the multivariate diffusion setting to derive lower bounds on the joint and marginal data likelihoods.

Kingma et al. (2021) show how to learn the inference process for VPSDE, that is learning the $\beta(s)$ function. For MDMs, $\mathbf{Q}$ can be non-zero; this shows that there are more choices to be made and learned in multivariate diffusions, compared to the scalar case. Additionally, one can choose the matrix $\mathbf{D}$ to be diagonal or full, give $\mathbf{Q}$ and $\mathbf{D}$ different time-varying functions, and learn $\nabla\mathbf{H}$.

Vahdat et al. (2021); Rombach et al. (2022) transform the data using auto-encoders and diffuse in a latent space. Future work is to use MDMs in concert with such encodings to yield effectively non-linear multivariate diffusions with automatic transitions.

7 Discussion

In this work, we provide a complete recipe for training MDMs. For any linear diffusion, we automate the transition kernel computation and sampling, removing the need for model specific analysis. We provide a parameterization of diffusions that have a specified noise distribution. Following Durkan & Song (2021); Huang et al. (2021) we provide a tractable ELBO to train MDM models. Finally, we show that for any number of auxiliary variables, we can learn the inference process, extending Kingma et al. (2021) beyond VPSDE. Our experiments show that we achieve state of the art bits-per-dim with significantly smaller models.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

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

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv preprint arXiv:1806.07366*, 2018.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34, 2021.

Tim Dockhorn, Arash Vahdat, and Karsten Kreis. Score-based generative modeling with critically-damped langevin diffusion. *arXiv preprint arXiv:2112.07068*, 2021.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *Advances in Neural Information Processing Systems*, 32, 2019.

Conor Durkan and Yang Song. On maximum likelihood training of score-based generative models. *arXiv preprint arXiv:2101.09258*, 2021.

Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. In *International Conference on Machine Learning*, pp. 2722–2730. PMLR, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, 23: 47–1, 2022.

Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*, 2018.

Chin-Wei Huang, Laurent Dinh, and Aaron Courville. Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. *arXiv preprint arXiv:2002.07101*, 2020.

Chin-Wei Huang, Jae Hyun Lim, and Aaron C Courville. A variational perspective on diffusion-based generative models and score matching. *Advances in Neural Information Processing Systems*, 34, 2021.

Dongjun Kim, Seungjae Shin, Kyungwoo Song, Wanmo Kang, and Il-Chul Moon. Score matching model for unbounded data score. *arXiv preprint arXiv:2106.05527*, 2021.

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

Diederik P Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *arXiv preprint arXiv:2107.00630*, 2021.

Alexej Klushyn, Nutan Chen, Richard Kurle, Botond Cseke, and Patrick van der Smagt. Learning hierarchical priors in vaes. *Advances in neural information processing systems*, 32, 2019.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32, 2019.

Gautam Mittal, Jesse Engel, Curtis Hawthorne, and Ian Simon. Symbolic music generation with diffusion models. *arXiv preprint arXiv:2103.16091*, 2021.

Wenlong Mou, Yi-An Ma, Martin J Wainwright, Peter L Bartlett, and Michael I Jordan. High-order langevin diffusion yields an accelerated mcmc algorithm. *arXiv preprint arXiv:1908.10859*, 2019.

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

Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *arXiv preprint arXiv:2102.09672*, 2021.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

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

Adam Roberts, Jesse Engel, and Douglas Eck. Hierarchical variational autoencoders for music. In *NIPS Workshop on Machine Learning for Creativity and Design*, volume 3, 2017.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Simo Särkkä and Arno Solin. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Hiroshi Sasaki, Chris G Willcocks, and Toby P Breckon. Unit-ddpm: Unpaired image translation with denoising diffusion probabilistic models. *arXiv preprint arXiv:2104.05358*, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in neural information processing systems*, 29, 2016.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020a.

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.

Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Truncated diffusion probabilistic models. *arXiv preprint arXiv:2202.09671*, 2022.

A DOES MY MODEL USE AUXILIARY VARIABLES?

In section 3 we gave the example choice of $q(\mathbf{y}_0^v|x) = q(\mathbf{y}_0^v) = \mathcal{N}(0,1)$ coordinate-wise. It is also a common choice to set π_θ as independent standard Gaussians for all dimensions. Because the optimum in diffusion models is $p_\theta = q$ for all t, we see a peculiar phenomenon under this choice: the model has main and auxiliary dimensions independent at both endpoints 0 and T. Does this mean that the model does not use auxiliary variables? We show that even when $q_\phi(\mathbf{y}_0)$ and π_θ have main and auxiliary variables independent, the model can use the auxiliary variables. A sufficient condition is $\mathbf{Q} + \mathbf{D}$ is non-diagonal.

To make this precise, we recall that we model with $p_\theta(\mathbf{u}_T^z = x)$. To show the model is using auxiliary variables, we just need to show that $\mathbf{u}_T^z$ (main coordinate at T) depends on $\mathbf{u}_t^v$ (aux. coordinate at t) for $T > t$. At optimum, $p_\theta(\mathbf{u}_T^z, \mathbf{u}_t^v) = q_\phi(\mathbf{y}_0^z, \mathbf{y}_{T-t}^v)$. Therefore it is sufficient to show that for some time s, $q_\phi(\mathbf{y}_s^v|\mathbf{y}_0^z) \neq q_\phi(\mathbf{y}_s^v)$. Because $\mathbf{y}_0^z$, is determined by x we need to show that $q_\phi(\mathbf{y}_s^v|x) \neq q_\phi(\mathbf{y}_s^v)$. To do that, we first derive $q(\mathbf{y}_s|x)$ and then marginalize to get $q(\mathbf{y}_s^v|x)$ from it. Since the former is 2D gaussian, the latter is available in terms of the former's mean and covariance. Suppose $\mathbb{E}[\mathbf{y}_0^v] = 0$, $Q = [[0,-1],[1,0]]$ and $D = [[1,0],[0,1]]$ and we have $s = .1$ We have:

$$\mathbb{E}[\mathbf{y}_s|x] = \exp\Big[-s(Q+D)\Big]\begin{pmatrix} x \\ 0 \end{pmatrix} = \exp\Big[\begin{bmatrix} -.1 & .1 \\ -.1 & -.1 \end{bmatrix}\Big]\begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{pmatrix} 0.9003x \\ -0.090x \end{pmatrix} \tag{20}$$

Regardless of the covariance any 1D of this 2D gaussian will have mean that is a function of x, meaning that $q(\mathbf{y}_s^v|x)$ does not equal $q(\mathbf{y}_s^v)$ (which is also a Gaussian but with mean depending on $x's$ mean rather than x itself. Therefore, even under the setup with independent endpoints, the optimal model makes use of the intermediate auxiliary variables in its final modeling distribution $p_\theta(\mathbf{u}_T^z = x)$.

Are there choices of $\mathbf{Q}$ and $\mathbf{D}$ that lead to learning models that don't make use of the extra dimensions? As mentioned, in the inference process, $\mathbf{Q}$ is responsible for mixing information among the coordinates, and is the only source of this when $\mathbf{D}$ is diagonal. Then, if $\mathbf{Q} = \mathbf{0}$ and $\mathbf{D}$ is diagonal, none of the coordinates for a given feature $\mathbf{x}_j$ (including $\mathbf{u}_{tj}^z, \mathbf{u}_{tj}^{v_1}, \ldots, \mathbf{u}_{tj}^{v_{K-1}}$) interact for any t. Then, since $p_\theta = q$ at optimum, independence of the coodinates at all t in q imply the same in p_θ and the model will not make use of any auxiliary variables when modeling the marginal $\log p_\theta(\mathbf{u}_T^z = x)$.

B STATIONARY PARAMETERIZATION

The general process family

$$d\mathbf{y} = f(\mathbf{y}, s)ds + g(s)d\widehat{\mathbf{B}}_s \tag{21}$$

can be restricted to those with stationary distributions. Ma et al. (2015) show a complete recipe to span all non-linear processes that have a given stationary distribution. We first observe that their formulation also applies to time-varying processes, though this is not mentioned in that work. We then restrict this to linear processes (per the other requirement for tractable transition kernels). The parameterization from Ma et al. (2015) specialized to linear processes is

$$d\mathbf{y} = \underbrace{-[\mathbf{Q}(s) + \mathbf{D}(s)]\nabla\mathbf{H}(\mathbf{y})}_{f}\,ds + \underbrace{\sqrt{2\mathbf{D}(s)}}_{g}\,d\widehat{\mathbf{B}}_s \tag{22}$$

where $\mathbf{Q}$ is skew-symmetric ($-\mathbf{Q} = \mathbf{Q}^\top$), $\mathbf{D}$ is positive semi-definite and $\mathbf{H}(\mathbf{y})$ determines the stationary distribution with $\mathbf{H}(\mathbf{y}) = -\log q_\infty(\mathbf{y})$. For a linear process $\nabla\mathbf{H}$ has to be linear, meanining these processes are all multivariate normal. This parameterization spans all linear Markov processes that have stationary distributions.

B.1 PARAMETERIZING $\mathbf{Q}_\phi$

Suppose $b_q(s)$ is a positive scalar function defined on the time domain with known integral. Suppose $\tilde{\mathbf{Q}}_\phi$ is any matrix. Then $\tilde{\mathbf{Q}}_\phi - \tilde{\mathbf{Q}}_\phi^\top$ is skew-symmetric with $\tilde{\mathbf{Q}}_{\phi,ij} = -\tilde{\mathbf{Q}}_{\phi,ji}$. We can set $\mathbf{Q}_\phi$ to

$$\mathbf{Q}_\phi(s) = b_q(s) \cdot \Big[\tilde{\mathbf{Q}}_\phi - \tilde{\mathbf{Q}}_\phi^\top\Big] \tag{23}$$

This is a general parameterization of time-independent skew-symmetric matrices, which have number of degrees of freedom equal to the number of entries in one of the triangles of the matrix, excluding the diagonal.

B.2 PARAMETERIZING $\mathbf{D}_\phi$

Suppose $b_d(s)$ is a positive scalar function defined on the time domain with known integral. Suppose $\tilde{\mathbf{D}}_\phi$ is any matrix. Then $\tilde{\mathbf{D}}_\phi \tilde{\mathbf{D}}_\phi^\top$ is positive semi-definite and spans all time-independent skew-symmetric matrices. We can set $\mathbf{D}_\phi$ to

$$\mathbf{D}_\phi(s) = b_d(s) \cdot \left[\tilde{\mathbf{D}}_\phi \tilde{\mathbf{D}}_\phi^\top\right] \tag{24}$$

To show $\tilde{\mathbf{D}}\tilde{\mathbf{D}}^\top$ spans all positive semi-definite matrices: suppose $\mathbf{M}$ is positive semi-definite. Then it is square. Then it can be eigen-decomposed into $\mathbf{M} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$ The degrees of freedom in $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top$ are just $\mathbf{R} = \mathbf{V}\sqrt{\mathbf{\Sigma}}$ since $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top = \mathbf{R}\mathbf{R}^\top$ and the square root is taken element-wise because $\mathbf{\Sigma}$ is diagonal and is real because each $\mathbf{\Sigma}_{ij} \geq 0$, which is true because $\mathbf{M}$ is positive semi-definite. Take $\mathbf{D} = \mathbf{R}$.

A simpler option is to just parameterize a non-negative vector to use as the diagonal of a diagonal-only matrix. But we stick with the fuller approach.

B.3 INTEGRALS

The known integral requirement comes from the integrals required in the transition kernel, and can be relaxed two possible ways:

- numerical integration of function with unknown integral, which might be okay since the functions are scalar-in scalar-out
- Directly parameterize the integral and use auto-grad when needing the functions not-integrated.

We stick with the known integrals. In conclusion, the underlying parameters are positive scalar functions $b_q(s), b_d(s)$ defined on the time domain and with known integral, and general matrices $\tilde{\mathbf{Q}}_\phi, \tilde{\mathbf{D}}_\phi$.

B.4 INSTANCES

CLD. The CLD process (eq 5 in Dockhorn et al. (2021)) is defined as

$$\begin{pmatrix} d\mathbf{x}_t \\ d\mathbf{v}_r \end{pmatrix} = d\mathbf{u}_t = \begin{pmatrix} 0 & \frac{\beta}{M} \\ -\beta & -\frac{\Gamma\beta}{M} \end{pmatrix} \mathbf{u}_t + \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{2\Gamma\beta} \end{pmatrix} d\tilde{\mathbf{w}}_t.$$

In Q, D, we have

$$H(\mathbf{u}) = \frac{1}{2}\|\mathbf{x}\|_2^2 + \frac{1}{2M}\|\mathbf{v}\|_2^2, \qquad \nabla_{\mathbf{u}} H(\mathbf{u}) = \begin{pmatrix} \mathbf{x} \\ \frac{1}{M}\mathbf{v} \end{pmatrix}$$
$$Q = \begin{pmatrix} 0 & -\beta \\ \beta & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 0 \\ 0 & \Gamma\beta \end{pmatrix}$$

And the stationary distribution is

$$q_{\phi,\infty} \propto \exp(-H(\mathbf{u})) = \mathcal{N}(\mathbf{x}; 0, I_d)\mathcal{N}(\mathbf{v}; 0, MI_d) \tag{25}$$

ALDA. (Mou et al., 2019) define a third-order diffusion process for the purpose of gradient-based MCMC sampling. The ALDA diffusion process can be specified as

$$Q = \begin{pmatrix} 0 & -\frac{1}{L}I & 0 \\ \frac{1}{L}I & 0 & -\gamma I \\ 0 & \gamma I & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{\xi}{L}I \end{pmatrix}. \tag{26}$$

Note that Q is skew-symmetric and D is positive semi-definite, therefore we have that $p_t(\mathbf{u}) \to$. In this case this is

$$q_{\phi,\infty} = \mathcal{N}(0, I_d)\mathcal{N}(\mathbf{z}; 0, \frac{1}{L}I_d)\mathcal{N}(\mathbf{r}; 0, \frac{1}{L}I_d)$$

MALDA. Similar to ALDA, we specify a diffusion process we term MALDA which we specify as

$$Q = \begin{pmatrix} 0 & -\frac{1}{L}I & -\frac{1}{L} \\ \frac{1}{L}I & 0 & -\gamma I \\ \frac{1}{L} & \gamma I & 0 \end{pmatrix}, \quad D = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{L}I & 0 \\ 0 & 0 & \frac{1}{L}I \end{pmatrix}. \tag{27}$$

Note that Q is skew-symmetric and D is positive semi-definite. In this case this is

$$q_{\phi,\infty} = \mathcal{N}(0, I_d)\mathcal{N}(\mathbf{v}_1; 0, \frac{1}{L}I_d)\mathcal{N}(\mathbf{v}_2; 0, \frac{1}{L}I_d)$$

C TRANSITIONS FOR LINEAR PROCESSES

For time variable s and Brownian motion $\widehat{\mathbf{B}}_s$ driving diffusions of the form

$$d\mathbf{y} = f(\mathbf{y}, s)ds + g(s)d\widehat{\mathbf{B}}_s, \tag{28}$$

when $f_\phi(\mathbf{y}_s, s), g_\phi(s)$ are linear, we can sample/compute transitions in closed form. The transition kernel $q_\phi(\mathbf{y}_s|\mathbf{y}_0)$ is always normal (Särkkä & Solin, 2019). Therefore, we just find the mean $\mathbf{m}_{s|0}$ and covariance $\mathbf{\Sigma}_{s|0}$ of $q(\mathbf{y}_s|\mathbf{y}_0)$. Let $f(\mathbf{y}, s) = \mathbf{A}(s)\mathbf{y}$. The un-conditional time s mean and covariance are solutions to

$$d\mathbf{m}_s/ds = \mathbf{A}(s)\mathbf{m}_s \tag{29}$$

$$d\mathbf{\Sigma}_s/ds = \mathbf{A}(s)\mathbf{\Sigma}_s + \mathbf{\Sigma}_s\mathbf{A}^\top(s) + g^2(s) \tag{30}$$

By (6.6) in the book for computing conditionals $q(\mathbf{y}_s|\mathbf{y}_0)$, we can take the marginal distribution ODEs and compute conditionals by simply setting the time 0 mean and covariance initial conditions to the conditioning value and to $\mathbf{0}$ respectively. We take (6.36-6.39) and set $\mathbf{m}_0 = \mathbf{u}_0$ and $\mathbf{\Sigma}_0 = 0$ to condition. Let $I_s[\mathbf{A}] = \int_0^s \mathbf{A}(\nu)d\nu$. The mean is

$$\mathbf{m}_{s|0} = \exp\left[\int_0^s \mathbf{A}(\nu)d\nu\right]\mathbf{y}_0 = \exp\left(I_s\Big[A\Big]\right) \underbrace{= \exp(s\mathbf{A})\mathbf{y}_0}_{\text{no integration if } \mathbf{A}(\nu) = \mathbf{A}}, \tag{31}$$

where exp denotes matrix exponential. (6.36-6.39) show how to express the covariance $q(\mathbf{y}_s|\mathbf{y}_0)$ as a matrix factorization[4] $\mathbf{\Sigma}_s = \mathbf{C}_s(\mathbf{H}_s)^{-1}$ for $\mathbf{C}_s, \mathbf{H}_s$ being the solutions of:

$$\begin{pmatrix} \frac{d}{ds}\mathbf{C}_s \\ \frac{d}{ds}\mathbf{H}_s \end{pmatrix} = \begin{pmatrix} \mathbf{A}(s) & g^2(s) \\ \mathbf{0} & -\mathbf{A}^\top(s) \end{pmatrix}\begin{pmatrix} \mathbf{C}_s \\ \mathbf{H}_s \end{pmatrix} \tag{32}$$

To condition and get $\mathbf{\Sigma}_{s|0}$ from $\mathbf{\Sigma}_s$, we set $\mathbf{\Sigma}_0 = \mathbf{0}$. To satisfy $\Sigma_0 = \mathbf{C}_0({\mathbf{H}_0}^{-1}) = \mathbf{0}$ we set $\mathbf{C}_0 = \mathbf{0}$ and $\mathbf{H}_0 = \mathbf{I}$.

$$\begin{pmatrix} \mathbf{C}_s \\ \mathbf{H}_s \end{pmatrix} = \exp\left[\begin{pmatrix} I_s[\mathbf{A}] & I_s[g^2] \\ \mathbf{0} & -I_s[\mathbf{A}^\top] \end{pmatrix}\right]\begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \underbrace{= \exp\left[s\begin{pmatrix} \mathbf{A} & g^2 \\ \mathbf{0} & -\mathbf{A}^\top \end{pmatrix}\right]\begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix}}_{\text{no integration if } \mathbf{A}(\nu) = \mathbf{A},\, g(\nu) = g}. \tag{33}$$

Finally, $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s(\mathbf{H}_s)^{-1}$.

C.1 HYBRID SCORE MATCHING

Alternatively, to get the distribution $q(\mathbf{y}_s|x)$ rather than $q(\mathbf{y}_s|\mathbf{y}_0)$, which corresponds to integrating out $\mathbf{v}_0$, we only need to integrate out $\mathbf{v}_0$. For the mean, we only need to replace the conditioning value of $\mathbf{v}_0$ with the mean of the chosen initial distribution. So the mean is

$$\mathbb{E}[\mathbf{y}_s|x] = \exp\left[\int_0^s A(\nu)d\nu\right]\begin{pmatrix} x \\ \mathbb{E}[\mathbf{v}_0] \end{pmatrix} \tag{34}$$

[4] We rename their $\mathbf{C}, \mathbf{D}$ to $\mathbf{C}, \mathbf{H}$ to avoid mixing up with $\mathbf{D}$ in our $\mathbf{Q}, \mathbf{D}$ stationary parameterization.

For the convariance, instead of using $\mathbf{C}_0 = \mathbf{\Sigma}_0 = \mathbf{0}$, we use a particular choice of $\mathbf{\Sigma}_0$ to condition on x but not $\mathbf{v}_0$. We decompose $\mathbf{\Sigma}_0$ into its blocks $\mathbf{\Sigma}_{0,xx}$, $\mathbf{\Sigma}_{0,vv}$,$\mathbf{\Sigma}_{0,xv}$. As before, to condition on x we set $\mathbf{\Sigma}_{0,xx} = \mathbf{0}$. Because $q(\mathbf{v}_0)$ is set to be independent of x, $\mathbf{\Sigma}_{0,xv}$ is also set to $\mathbf{0}$. Finally, instead of $\mathbf{0}$, to marginalize out $\mathbf{v}_0$, $\mathbf{\Sigma}_{0,vv}$ is set to the covariance of the chosen initial time zero distribution for $\mathbf{v}_0$. E.g. if $\mathbf{v}_{0,j} \sim N(0, \gamma)$ for each dimension, then $\mathbf{\Sigma}_{0,vv} = N(0, \gamma I)$.

We operationalize this in a simple piece of code, which makes the ELBO tractable and easy, i.e. skips both analytic derivations and numerical forward integration during training. Altogether, $q_{\phi,s,(x,v)} = q_\phi(\mathbf{y}_s | \mathbf{y}_0 = [x, v]) = \mathcal{N}(\mathbf{y}_s; \mathbf{m}_{s|0}, \mathbf{\Sigma}_{s|0})$.

C.2 Transitions in Stationary Parameterization

In terms of $\mathbf{Q}, \mathbf{D}$, the transitions $q(\mathbf{y}_s|\mathbf{y}_0)$ for time s are normal with mean $\mathbf{m}_{s|0}$ and $\mathbf{\Sigma}_{s|0}$ equal to:

$$\mathbf{m}_{s|0} = \exp\Big(- I_s\Big[\mathbf{Q} + \mathbf{D}\Big]\Big)\mathbf{y}_0, \qquad \begin{pmatrix}\mathbf{C}_s \\ \mathbf{H}_s\end{pmatrix} = \exp\left[\begin{pmatrix} -I_s[\mathbf{Q}+\mathbf{D}] & I_s[2\mathbf{D}] \\ \mathbf{0} & I_s[(\mathbf{Q}+\mathbf{D})^\top]\end{pmatrix}\right]\begin{pmatrix}\mathbf{0} \\ \mathbf{I}\end{pmatrix} \tag{35}$$

where $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s(\mathbf{H}_s)^{-1}$. For the time invariant case, this simplifies to

$$\mathbf{m}_{s|0} = \exp[-s(\mathbf{Q}+\mathbf{D})]\mathbf{y}_0, \qquad \begin{pmatrix}\mathbf{C}_s \\ \mathbf{H}_s\end{pmatrix} = \exp\left[s\begin{pmatrix} -(\mathbf{Q}+\mathbf{D}) & 2\mathbf{D} \\ \mathbf{0} & (\mathbf{Q}+\mathbf{D})^\top\end{pmatrix}\right]\begin{pmatrix}\mathbf{0} \\ \mathbf{I}\end{pmatrix} \tag{36}$$

D Generic change of measure and Jensen's for approximate marginalization

Suppose $\mathbf{u} = [\mathbf{z}, \mathbf{v}]$ and we have an expression for $p(\mathbf{u} = [z, v]) = p(\mathbf{z} = z, \mathbf{v} = v)$. By marginalization, we can get $p(\mathbf{z} = z)$, and we can introduce another distribution q to pick a sampling distribution of our choice:

$$\begin{aligned} p(\mathbf{z} = z) &= \int_v p(\mathbf{z} = z, \mathbf{v} = v)dv \\ &= \int_v p(\mathbf{z} = z|\mathbf{v} = v)p(\mathbf{v} = v)dv \\ &= \int_v \frac{q(\mathbf{v} = v|\mathbf{z} = z)}{q(\mathbf{v} = v|\mathbf{z} = z)} p(\mathbf{z} = z|\mathbf{v} = v)p(\mathbf{v} = v)dv \\ &= \mathbb{E}_{q(\mathbf{v}=v|\mathbf{z}=z)}\Big[\frac{p(\mathbf{z} = z, \mathbf{v} = v)}{q(\mathbf{v} = v|\mathbf{z} = z)}\Big] \end{aligned} \tag{37}$$

We often work with these expressions in log space, and need to pull the expectation outside to use monte carlo. Jensen's bound allows us to do this:

$$\begin{aligned} \log p(\mathbf{z} = z) &= \log \mathbb{E}_{q(\mathbf{v}=v|\mathbf{z}=z)}\Big[\frac{p(\mathbf{z} = z, \mathbf{v} = v)}{q(\mathbf{v} = v|\mathbf{z} = z)}\Big] \\ &\geq \mathbb{E}_{q(\mathbf{v}=v|\mathbf{z}=z)}\Big[\log \frac{p(\mathbf{z} = z, \mathbf{v} = v)}{q(\mathbf{v} = v|\mathbf{z} = z)}\Big] \end{aligned}$$

The following shows us that the bound is tight when $q(\mathbf{v} = v|\mathbf{z} = z) = p(\mathbf{v} = v|\mathbf{z} = z)$:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{v}=v|\mathbf{z}=z)}\Big[\log \frac{p(\mathbf{z} = z, \mathbf{v} = v)}{q(\mathbf{v} = v|\mathbf{z} = z)}\Big] &=_{\text{assume}} \mathbb{E}_{p(\mathbf{v}=v|\mathbf{z}=z)}\Big[\log \frac{p(\mathbf{z} = z, \mathbf{v} = v)}{p(\mathbf{v} = v|\mathbf{z} = z)}\Big] \\ &= \mathbb{E}_{p(\mathbf{v}=v|\mathbf{z}=z)}\Big[\log\Big(\frac{p(\mathbf{z} = z, \mathbf{v} = v)}{p(\mathbf{v} = v, \mathbf{z} = z)} \cdot p(\mathbf{z} = z)\Big)\Big] \\ &= \mathbb{E}_{p(\mathbf{v}=v|\mathbf{z}=z)}\Big[\log p(\mathbf{z} = z)\Big] \\ &= \log p(\mathbf{z} = z) \end{aligned} \tag{38}$$

E ELBO FOR MDMS

We carry out a multivariate analog of the ELBO derivation from Huang et al. (2021) on the extended space $\mathbf{u}$, with extra marginalization of the time-zero auxiliary variables.

Later, in appendix F, we go from the multivariate ISM bound to the multivariate DSM bound.

E.1 SETUP

Let $\mathbf{u} = [\mathbf{u}^z, \mathbf{u}^v]$ have model distribution p_θ and $\mathbf{y} = [\mathbf{y}^z, \mathbf{y}^v]$ have inference distribution q_ϕ. For $\mathbf{y}$, we will consider two inference distributions, first Q_1 which has one definition of $d\mathbf{y}$ and Brownian motion $d\mathbf{B}'_s$, and then Q_2 which has a different definition of $\mathbf{y}$ and Brownian motion $d\widehat{\mathbf{B}}_s$ (Huang et al. (2021) use P and Q for these two distributions, but we rename to avoid confusion with model p_θ).

E.2 EXACT MARGINAL LIKELIHOOD

For a model of the form:

$$\mathbf{u}_0 \sim \pi_\theta, \qquad d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + \beta_\theta(t)d\mathbf{B}_t \tag{39}$$

where we model data $\mathbf{x} = x$ as $p_\theta(\mathbf{u}^z_T = x) = p_\theta(\mathbf{z}_T = x)$, theorem 1 of Huang et al. (2021) show that the marginal (over time) likelihood can be expressed as

$$p_\theta(\mathbf{u}_T = [x, v]) = \mathbb{E}_{Q_1(\mathbf{y}|\mathbf{y}_0=(x,v))}\left[\pi_\theta(\mathbf{y}_T)\exp\left(\int_0^T -\nabla\cdot h_\theta(\mathbf{y}, T-s)ds\right)\right] \tag{40}$$

where $\mathbf{y} \sim Q_1(\mathbf{y}|\mathbf{y}_0 = (x, v))$ is the randomness in:

$$d\mathbf{y}_s = -h_\theta(\mathbf{y}, T-s)ds + \beta_\theta(T-s)d\mathbf{B}'_s \tag{41}$$

E.3 MARGINALIZE AUX VARS

Theorem 1 in Huang et al. (2021) is originally applied to the process (that in our notation is) $\mathbf{z}$, which does not contain auxiliary variables so that $\mathbf{x}$ is modeled as $\mathbf{z}_T$. We used their expression for $\mathbf{u}$. We must now integrate to get $p_\theta(\mathbf{z}_T = x)$. We proceed in two steps, first, use eq. (37) with $z = x$ to note that, for any distribution $q_\gamma(\mathbf{y}^v_0|x)$,

$$p_\theta(\mathbf{z}_T = x) = \mathbb{E}_{q_\gamma(\mathbf{y}^v_0|x)}\left[\frac{p_\theta(\mathbf{u}_T = [x, \mathbf{y}^v_0])}{q_\gamma(\mathbf{y}^v_0|x)}\right] \tag{42}$$

We will choose the exact form of q_γ later. Next, let us substitute in the expression from eq. (40) for the joint into the numerator:

$$p_\theta(\mathbf{z}_T = x) = \mathbb{E}_{q_\gamma(\mathbf{y}^v_0|x)}\left[\frac{\mathbb{E}_{Q_1(\mathbf{y}|\mathbf{y}_0=(x,\mathbf{y}^v_0))}\left[\pi_\theta(\mathbf{y}_T)\exp\left(\int_0^T -\nabla\cdot h_\theta(\mathbf{y}, T-s)ds\right)\right]}{q_\gamma(\mathbf{y}^v_0|x)}\right] \tag{43}$$

where $\mathbf{y} \sim Q_1(\mathbf{y}|\mathbf{y}_0 = (x, \mathbf{y}^v_0))$ is again defined by:

$$d\mathbf{y}_s = -h_\theta(\mathbf{y}, T-s)ds + \beta_\theta(T-s)d\mathbf{B}'_s \tag{44}$$

E.4 CHANGE OF MEASURE

$$\begin{aligned}
\log p(\mathbf{z}_T = x) &= \log \mathbb{E}_{q_\gamma(\mathbf{y}^v_0|x)}\mathbb{E}_{Q_1(\mathbf{y}|\mathbf{y}_0=(x,\mathbf{y}^v_0))}\left[\frac{\left[\pi_\theta(\mathbf{y}_T)\exp\left(\int_0^T -\nabla\cdot h_\theta(\mathbf{y}, T-s)ds\right)\right]}{q_\gamma(\mathbf{y}^v_0|x)}\right] \\
&= \log \underbrace{\mathbb{E}_{q_\gamma(\mathbf{y}^v_0|x)}\mathbb{E}_{Q_2(\mathbf{y}|\mathbf{y}_0=(x,\mathbf{y}^v_0))}}_{:=Q_2(\mathbf{y}|x)}\left[\underbrace{\frac{dQ_1}{dQ_2}\frac{\left[\pi_\theta(\mathbf{y}_T)\exp\left(\int_0^T -\nabla\cdot h_\theta(\mathbf{y}, T-s)ds\right)\right]}{q_\gamma(\mathbf{y}^v_0|x)}}_{\ell(\mathbf{y},x)}\right]
\end{aligned}$$

for any Q_2 for where dQ_1/dQ_2 is defined and where still $\mathbf{y} \sim Q_1(\mathbf{y}|\mathbf{y}_0 = (x, v))$ defined by:

$$d\mathbf{y}_s = -h_\theta(\mathbf{y}, T-s)ds + \beta_\theta(T-s)d\mathbf{B}'_s$$

E.5 JENSENS

We can now lower bound through both expectations

$$\log p_\theta(\mathbf{z}_T = x) = \log \mathbb{E}_{Q_2(\mathbf{y}|x)}\Big[\ell(\mathbf{y}, s)\Big]$$
$$\geq \mathbb{E}_{Q_2(\mathbf{y}|x)}\left[\log \frac{dQ_1}{dQ_2} + \log \pi_\theta(\mathbf{y}_T) - \int_0^T \nabla \cdot h_\theta(\mathbf{y}, T-s)ds - \log q_\gamma(\mathbf{y}_0^v|x)\right]$$

We will simplify the likelihood ratio but we first take a brief detour to derive a multi sample bound.

E.6 PICKING Q_2 WITH GIRSANOV

We have a likelihood ratio in

$$\ell(\mathbf{y}, x) = \frac{dQ_1}{dQ_2}\frac{\pi_\theta(\mathbf{y}_T)\exp\Big(\int_0^T -\nabla \cdot h_\theta(\mathbf{y}, T-s)ds\Big)}{q_\gamma(\mathbf{y}_0^v|x)} \tag{45}$$

Use Girsanov to pick Q_2 shifted from Q_1 to get known dQ_1/dQ_2. By Tzen & Raginsky (2019); Huang et al. (2021) and Girsanov, ℓ is equal to

$$\ell(\mathbf{y}, x) = \log \pi_\theta(\mathbf{y}_T) + \int_0^T -\frac{1}{2}||a_\theta(\omega, s)||_2^2 - \nabla \cdot h_\theta(\mathbf{y}, T-s)ds + -\log q_\gamma(\mathbf{y}_0^v|x) \tag{46}$$

where $Q_2(\mathbf{y}|\mathbf{y}_0 = (x, v))$ is defined by

$$d\mathbf{y}_s = \Big[-h_\theta(\mathbf{y}, T-s) + \beta_\theta(T-s)a_\theta(\omega, s)\Big]ds + \beta_\theta(T-s)d\widehat{\mathbf{B}}_s \tag{47}$$

Notice we now have a different SDE defining the process $\mathbf{y}_s$ with an independent Brownian motion $d\widehat{\mathbf{B}}_s$. Recall the model is:

$$d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + \beta_\theta(t)d\mathbf{B}_t \tag{48}$$

Q_1 and the model already had the same diffusion coefficient β_θ by Huang's theorem. Q_2 and Q_1 had the same diffusion coefficient due to a constraint for using Girsanov. Therefore Q_2 and the model have the same diffusion coefficient β_θ.

E.7 SIMPLIFYING a_θ

In general a_θ depends on an infinite-dimensional element ω of the Wiener measure's sample space. Huang et al. (2021) simplify this according to the Markov assumption, to just $a_\theta(\omega, s) = a_\theta(\mathbf{y}_s(\omega), s)$, where $\mathbf{y}_s(\omega)$ is commonly notated just as $\mathbf{y}_s$. This is motivated by the fact that there exists an a that is just a function of $(\mathbf{y}_s(\omega), s)$ that closes the Jensen gap. In this notation,

$$\ell(\mathbf{y}, x) = \log \pi_\theta(\mathbf{y}_T) + \int_0^T -\frac{1}{2}||a_\theta(\mathbf{y}, s)||_2^2 - \nabla \cdot h_\theta(\mathbf{y}, T-s)ds - \log q_\gamma(\mathbf{y}_0^v|x) \tag{49}$$

E.8 SUMMARY OF LOWER BOUND WITH Q_2 AND SIMPLIFIED a_θ

The model is

$$d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + \beta_\theta(t)d\mathbf{B}_t \tag{50}$$

The inference is

$$d\mathbf{y}_s = \Big[-h_\theta(\mathbf{y}, T-s) + \beta_\theta(T-s)a_\theta(\mathbf{y}, s)\Big]ds + \beta_\theta(T-s)d\widehat{\mathbf{B}}_s \tag{51}$$

and the ELBO is

$$\log p_\theta(\mathbf{z}_T = x) \geq$$
$$\mathbb{E}_{Q_2(\mathbf{y}|s)}\left[\log \pi_\theta(\mathbf{y}_T) + \int_0^T -\frac{1}{2}||a_\theta(\mathbf{y}, s)||_2^2 - \nabla \cdot h_\theta(\mathbf{y}, T-s)ds - \log q_\gamma(\mathbf{y}_0^v|x)\right] \tag{52}$$

E.9 REWRITE MODEL AND INFERENCE

We re-write with choices of h_θ and a_θ to make the parameterization look more familiar. Important to note that the names f_ϕ, g_ϕ, and s_θ that we are about to introduce have not appeared in this appendix section and have no connotation. Start at

$$d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + \beta_\theta(t)d\mathbf{B}_t \tag{53}$$

$$d\mathbf{y}_s = \Big[- h_\theta(\mathbf{y}, T-s) + \beta_\theta(T-s)a_\theta(\mathbf{y}, s)\Big]ds + \beta_\theta(T-s)d\widehat{\mathbf{B}}_s \tag{54}$$

Next, replace $\beta_\theta(t)$ with new function g_ϕ such that $\beta_\theta(t) = g_\phi(T-t)$, which also means $\beta_\theta(T-s) = g_\phi(s)$

$$d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + g_\phi(T-t)d\mathbf{B}_t \tag{55}$$

$$d\mathbf{y}_s = \Big[- h_\theta(\mathbf{y}, T-s) + g_\phi(s)a_\theta(\mathbf{y}, s)\Big]ds + g_\phi(s)d\widehat{\mathbf{B}}_s \tag{56}$$

Next, introduce new function $f_\phi(\cdot, s)$ such that

$$f_\phi(\cdot, s) = g_\phi(s)a_\theta(\cdot, s) - h_\theta(\cdot, T-s) \tag{57}$$

which implies

$$f_\phi(\cdot, s) = g_\phi(s)a_\theta(\cdot, s) - h_\theta(\cdot, T-s) \tag{58}$$

$$f_\phi(\cdot, s) + h_\theta(\cdot, T-s) = g_\phi(s)a_\theta(\cdot, s) \tag{59}$$

$$h_\theta(\cdot, T-s) = g_\phi(s)a_\theta(\cdot, s) - f_\phi(\cdot, s) \tag{60}$$

$$h_\theta(\cdot, t) = g_\phi(T-t)a_\theta(\cdot, T-t) - f_\phi(\cdot, T-t) \tag{61}$$

$$\tag{62}$$

That means we have

$$d\mathbf{u}_t = h_\theta(\mathbf{u}, t)dt + g_\phi(T-t)d\mathbf{B}_t \tag{63}$$

$$= \Bigg[g_\phi(T-t)a_\theta(\mathbf{u}, T-t) - f_\phi(\mathbf{u}, T-t)\Bigg]dt + g_\phi(T-t)d\mathbf{B}_t \tag{64}$$

$$\tag{65}$$

and

$$d\mathbf{y}_s = \Big[- h_\theta(\mathbf{y}, T-s) + g_\phi(s)a_\theta(\mathbf{y}, s)\Big]ds + g_\phi(s)d\widehat{\mathbf{B}}_s \tag{66}$$

$$= \Bigg[f_\phi(\mathbf{y}, s) - g_\phi(s)a_\theta(\mathbf{y}, s) + g_\phi(s)a_\theta(\mathbf{y}, s)\Bigg]ds + g_\phi(s)d\widehat{\mathbf{B}}_s \tag{67}$$

$$= f_\phi(\mathbf{y}, s)ds + g_\phi(s)d\widehat{\mathbf{B}}_s \tag{68}$$

At this point, we are free to introduce new function $s_\theta(\mathbf{u}, t)$ and possibly set

$$a_\theta(\mathbf{u}, T-t) = g_\phi(T-t)^\top s_\theta(\mathbf{u}, T-t) \tag{69}$$

in which case, letting g^2 denote $gg^\top$ we would have

$$d\mathbf{u}_t = \Bigg[g_\phi^2(T-t)s_\theta(\mathbf{u}, T-t) - f_\phi(\mathbf{u}, T-t)\Bigg]dt + g_\phi(T-t)d\mathbf{B}_t \tag{70}$$

$$d\mathbf{y}_s = f_\phi(\mathbf{y}, s)ds + g_\phi(s)d\widehat{\mathbf{B}}_s \tag{71}$$

E.10 ELBO WITH RE-WRITTEN MODEL AND INFERENCE

The ELBO with re-written model and inference is:

$$\log p_\theta(\mathbf{z}_T = x) \geq \mathbb{E}_{Q_2(\mathbf{y}|x)}\Bigg[\log \pi_\theta(\mathbf{y}_T) + \int_0^T -\frac{1}{2}||s_\theta(\mathbf{y}, s)||^2_{g^2_\phi(s)} - \nabla \cdot h_\theta(\mathbf{y}, T-s)ds - \log q_\gamma(\mathbf{y}_0^v|x)\Bigg] \tag{72}$$

Let's give the above ELBO some shorthand. Define:

$$\begin{aligned}
\ell_{s_\theta} &= -\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi} \\
\ell_{\text{div-gsf}} &= \nabla_{\mathbf{y}_s} \cdot -\Big[\Big(g^2_\phi(s) s_\theta(\mathbf{y}_s, s)\Big) - f_\phi(\mathbf{y}_s, s)\Big] \\
\ell_s^{(ism)} &= \mathbb{E}_{s\sim\text{Unif}(0,T)}\Big[\ell_{s_\theta} + \ell_{\text{div-gsf}}\Big] \\
\ell_T &= \mathbb{E}_{q_{\phi,T},(x,v)}\Big[\log \pi_\theta(\mathbf{y}_T)\Big] \\
\ell_q &= -\log q_\gamma(\mathbf{y}_0^v|x)
\end{aligned} \tag{73}$$

In this notation,

$$\log p_\theta(x;\theta,\phi,\gamma)^{(ism)} \geq \mathbb{E}_q\Big[\ell_s^{(ism)} + \ell_T + \ell_q\Big] \tag{74}$$

F ISM ELBO TO DSM ELBO

We take the above ISM ELBO and derive a form of the ELBO that looks closer to DSM, which we call the DSM elbo. Recall the ISM form of the ELBO is

$$\mathbb{E}_{\mathbf{y}|x}\Bigg[-\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} - \underbrace{\nabla_{\mathbf{y}_s}\cdot\Big(g^2_\phi(s)s_\theta(\mathbf{y}, s)\Big)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg]$$

Using appendix F.1, we can swap the div term for one containing a marginal score:

$$\mathbb{E}_{\mathbf{y}|x}\Big[\nabla_{\mathbf{y}_s}\cdot\Big(g^2_\phi(s)s_\theta(\mathbf{y}, s)\Big)\Big] = \mathbb{E}_{\mathbf{y}|x}\Bigg[-s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)\Bigg] \tag{75}$$

This means the ELBO equals

$$\mathbb{E}_{\mathbf{y}|x}\Bigg[-\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} + \underbrace{s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg]$$

Using appendix F.2, we can turn the marginal score into one of two kinds of conditionals:

$$\begin{aligned}
\mathbb{E}_{\mathbf{y}|x}\Bigg[s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\underbrace{\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)}\Bigg] &= \mathbb{E}_{\mathbf{y}|x}\Big[s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\underbrace{\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)}\Big] \\
&= \mathbb{E}_{\mathbf{y}|x}\Big[s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\underbrace{\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)}\Big]
\end{aligned} \tag{76}$$

Meaning the ELBO also equals both of these terms

$$\begin{aligned}
&\mathbb{E}_{\mathbf{y}|x}\Bigg[-\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} + s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\underbrace{\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg] \\
=&\mathbb{E}_{\mathbf{y}|x}\Bigg[-\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} + s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\underbrace{\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg]
\end{aligned}$$

Next, for "score" equal to either $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)$ or $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)$, note

$$-\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} + s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s)\text{score} \quad = \quad -\frac{1}{2}||s_\theta(\mathbf{y}_s, s) - \text{score}||^2_{g^2_\phi(s)} + \frac{1}{2}||\text{score}||^2_{g^2_\phi(s)}$$

This means the ism ELBO is also equal to both of these:

$$\begin{aligned}
&\mathbb{E}_q\Bigg[-\frac{1}{2}||s_\theta(\mathbf{y}_s, s) - \nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)||^2_{g^2_\phi(s)} + \frac{1}{2}||\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)||^2_{g^2_\phi(s)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg] \\
=&\mathbb{E}_q\Bigg[-\frac{1}{2}||s_\theta(\mathbf{y}_s, s) - \nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)||^2_{g^2_\phi(s)} + \frac{1}{2}||\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)||^2_{g^2_\phi(s)} + \nabla_{\mathbf{y}_s}\cdot f_\phi(\mathbf{y}, s) + \log\pi_\theta(\mathbf{y}_T) - \log q(\mathbf{y}_0^{(v)})\Bigg]
\end{aligned}$$

For shorthand,

$$
\begin{aligned}
\ell_{\text{div-f}} &= \nabla_{\mathbf{y}_s} \cdot f_\phi(\mathbf{y}_s, s) \\
\ell_{\text{fwd-score}} &= \frac{1}{2}\Big|\Big|\nabla_{\mathbf{y}_s} \log q_\phi(\mathbf{y}_s|\mathbf{y}_0)\Big|\Big|^2_{g^2_\phi(s)} \\
\ell_{\text{neg-scorediff}} &= -\frac{1}{2}\Big|\Big|s_\theta - \nabla_{\mathbf{y}_s} \log q_\phi(\mathbf{y}_s|\mathbf{y}_0)\Big|\Big|^2_{g^2_\phi(s)} \\
\ell_s^{(dsm)} &= \mathbb{E}_{s\sim\text{Unif}(0,T)}\Big[\ell_{\text{neg-scorediff}} + \ell_{\text{fwd-score}} + \ell_{\text{div-f}}\Big]
\end{aligned}
$$

and swap out just the $\ell_s^{(ism)}$ term with $\ell_s^{(dsm)}$ to get the DSM elbo:

$$
\log p_\theta(x;\theta,\phi,\gamma)^{(dsm)} \geq \mathbb{E}_q\left[\ell_s^{(dsm)} + \ell_T + \ell_q\right] \tag{77}
$$

We can also use the score conditional on x instead of $\mathbf{y}_0$.

F.1 Trick 1: Reverse Integration by parts to remove divergence term

$$
\mathbb{E}_{\mathbf{y}_s}\Big[\nabla_{\mathbf{y}_s} \cdot \Big(g^2_\phi(s) s_\theta(\mathbf{y}, s)\Big)\Big] = \underbrace{\int_{\mathbf{y}_s} \nabla_{\mathbf{y}_s} \cdot \Big(q g^{2\top}(s) s_\theta(\mathbf{y}_s, s)\Big) d\mathbf{y}_s}_{0} - \int_{\mathbf{y}_s} q(\mathbf{y}_s) \nabla_{\mathbf{y}_s}\Big(g^{2\top}(s) s_\theta(\mathbf{y}_s, s)\Big) d\mathbf{y}_s \tag{78}
$$

$$
= \int_{\mathbf{y}_s} s_\theta(\mathbf{y}, s)^\top g^2_\phi(s) \nabla_{\mathbf{y}_s} q(\mathbf{y}_s, s) d\mathbf{y}_s \tag{79}
$$

$$
= \int_{\mathbf{y}_s} q(\mathbf{y}_s) s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s) \nabla_{\mathbf{y}_s} \log q(\mathbf{y}_s) d\mathbf{y}_s \tag{80}
$$

$$
= \mathbb{E}_{\mathbf{y}_s}\left[s_\theta(\mathbf{y}_s, s)^\top g^2_\phi(s) \nabla_{\mathbf{y}_s} \log q(\mathbf{y}_s)\right] \tag{81}
$$

F.2 TRICK 2: MARGINAL TO CONDITIONAL SCORE

We derive this generically for function f. First, to change the marginal to $\mathbf{y}_s|\mathbf{y}_0$

$$\mathbb{E}_{\mathbf{y}}\Big[f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)\Big] \tag{82}$$

$$= \mathbb{E}_{\mathbf{y}_s}\Big[f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)\Big] \tag{83}$$

$$= \int_{\mathbf{y}_s} q(\mathbf{y}_s)f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)d\mathbf{y}_s \tag{84}$$

$$= \int_{\mathbf{y}_s} f(\mathbf{y}_s)\nabla_{\mathbf{y}_s} q(\mathbf{y}_s)d\mathbf{y}_s \tag{85}$$

$$= \int_{\mathbf{y}_s} f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\int_{\mathbf{y}_0} q(\mathbf{y}_s|\mathbf{y}_0)q(\mathbf{y}_0)d\mathbf{y}_0 d\mathbf{y}_s \tag{86}$$

$$= \int_{\mathbf{y}_s} f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\int_{\mathbf{y}_0} q(\mathbf{y}_s|\mathbf{y}_0)q(\mathbf{y}_0)d\mathbf{y}_0 d\mathbf{y}_s \tag{87}$$

$$= \int_{\mathbf{y}_0}\int_{\mathbf{y}_s} f(\mathbf{y}_s)\nabla_{\mathbf{y}_s} q(\mathbf{y}_s|\mathbf{y}_0)q(\mathbf{y}_0)d\mathbf{y}_s d\mathbf{y}_0 \tag{88}$$

$$= \mathbb{E}_{\mathbf{y}_0}\left[\int_{\mathbf{y}_s} f(\mathbf{y}_s)\nabla_{\mathbf{y}_s} q(\mathbf{y}_s|\mathbf{y}_0)d\mathbf{y}_s\right] \tag{89}$$

$$= \mathbb{E}_{\mathbf{y}_0}\left[\int_{\mathbf{y}_s} f(\mathbf{y}_s)q(\mathbf{y}_s|\mathbf{y}_0)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)d\mathbf{y}_s\right] \tag{90}$$

$$= E_{\mathbf{y}_0}\mathbb{E}_{\mathbf{y}_s|\mathbf{y}_0}\left[f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\right] \tag{91}$$

$$= \mathbb{E}_{\mathbf{y}_0}\mathbb{E}_{\mathbf{y}_s|\mathbf{y}_0}\left[f(\mathbf{y}_s)\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\right] \tag{92}$$

We just need to pick $f(\mathbf{y}_s) = s_\theta(\mathbf{y}_s, s)^\top g_\phi^2(s)$. We can repeat the same derivation to turn $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s)$ into $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|x)$ rather than $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)$.

F.3 NOISE PREDICTION

For DSM elbo, the negative of the objective terms with s_θ (ie the thing to be minimized) is

$$\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}, s) - \nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\Big|\Big|^2_{g_\phi^2(s)} \tag{93}$$

$$= \frac{1}{2}\Big(s_\theta(\mathbf{y}, s) - \nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\Big)^\top g_\phi^2(s)\Big(s_\theta(\mathbf{y}, s) - \nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\Big) \tag{94}$$

Let us now hard-code our knowledge about $\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)$ to reduce variance. First, consider the scalar (no aux vars) case. We know that

$$q(\mathbf{y}_s|\mathbf{y}_0) = \mathcal{N}(\mathbf{y}_s; \gamma_s\mathbf{y}_0, \sigma_s^2) \tag{95}$$

meaning $\mathbf{y}_s = \gamma_s\mathbf{y}_0 + \sigma_s\epsilon$ where $\epsilon \sim \mathcal{N}(0, 1)$ and γ_s, σ_s are mean coefficients and std dev for $\mathbf{y}_s|\mathbf{y}_0$ such that $\mathbf{m}_{s|0} = \gamma_s\mathbf{y}_0$. This implies

$$\nabla_{\mathbf{y}_s}\log q(\mathbf{y}_s|\mathbf{y}_0)\Big|_{\mathbf{y}_s = \gamma_s\mathbf{y}_0 + \sigma_s\epsilon} = -\frac{\epsilon}{\sigma_s}$$

So lets parameterize $s_\theta(\mathbf{y}_s, s)$ as

$$s_\theta(\mathbf{y}_s, s) = -\frac{\epsilon_\theta(\mathbf{y}_s, s)}{\sigma_s} \tag{96}$$

This gives

$$\frac{1}{2}\Big|\Big| - \frac{\epsilon_\theta(\mathbf{y}_s, s)}{\sigma_s} - -\frac{\epsilon}{\sigma_s}\Big|\Big|^2_{g^2_\phi(s)} \tag{97}$$

$$= \frac{1}{2}\Big|\Big| \frac{\epsilon}{\sigma_s} - \frac{\epsilon_\theta(\mathbf{y}_s, s)}{\sigma_s}\Big|\Big|^2_{g^2_\phi(s)} \tag{98}$$

$$= \frac{1}{2\sigma_s^2}\Big|\Big| \epsilon - \epsilon_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{g^2_\phi(s)} \tag{99}$$

Now let's consider the multivariate case. We have that for normal $\mathcal{N}(\mathbf{y}_s; \mathbf{m}_{s|0}, \mathbf{\Sigma}_{s|0})$, we can sample $\mathbf{y}_s$ with normal noise $\epsilon \sim \mathcal{N}(0, I)$ and $\mathbf{y}_s = \mathbf{m}_{s|0} + \mathbf{L}\epsilon$ where $\mathbf{L}$ is the cholesky decomposition of $\mathbf{\Sigma}_{s|0}$ Then, the score is

$$\nabla_{\mathbf{y}_s} \log q(\mathbf{y}_s|\mathbf{y}_0)\Big|_{\mathbf{y}_s = \mathbf{m}_{s|0} + \mathbf{L}\epsilon} \tag{100}$$

$$= -\mathbf{\Sigma}_{s|0}^{-1}\Big(\mathbf{y}_s - \mathbf{m}_{s|0}\Big) \tag{101}$$

$$= -\mathbf{\Sigma}_{s|0}^{-1}\Big(\Big[\mathbf{m}_{s|0} + \mathbf{L}\epsilon\Big] - \mathbf{m}_{s|0}\Big) \tag{102}$$

$$= -\mathbf{\Sigma}_{s|0}^{-1}\Big(\mathbf{L}\epsilon\Big) \tag{103}$$

$$= -\Big(\mathbf{L}\mathbf{L}^\top\Big)^{-1}\Big(\mathbf{L}\epsilon\Big) \tag{104}$$

$$= -\Big(\mathbf{L}^\top\Big)^{-1}\mathbf{L}^{-1}\mathbf{L}\epsilon \tag{105}$$

$$= -\Big(\mathbf{L}^\top\Big)^{-1}\epsilon = -\Big(\mathbf{L}^{-1}\Big)^\top\epsilon = -\mathbf{L}^{\top,-1}\epsilon \tag{106}$$

So lets parameterize $s_\theta(\mathbf{y}_s, s)$ as

$$s_\theta(\mathbf{y}_s, s) = -\mathbf{L}^{\top,-1}\epsilon_\theta(\mathbf{y}, s) \tag{107}$$

This gives

$$\frac{1}{2}\Big|\Big| - \mathbf{L}^{\top,-1}\epsilon_\theta(\mathbf{y}, s) \quad - \quad -\mathbf{L}^{\top,-1}\epsilon\Big|\Big|^2_{g^2_\phi(s)} \tag{108}$$

$$= \frac{1}{2}\Big|\Big|\mathbf{L}^{\top,-1}\epsilon \quad - \quad \mathbf{L}^{\top,-1}\epsilon_\theta(\mathbf{y}, s)\Big|\Big|^2_{g^2_\phi(s)} \tag{109}$$

$$= \frac{1}{2}\Bigg(\mathbf{L}^{\top,-1}\epsilon \quad - \quad \mathbf{L}^{\top,-1}\epsilon_\theta(\mathbf{y}, s)\Bigg)^\top g^2_\phi(s)\Bigg(\mathbf{L}^{\top,-1}\epsilon \quad - \quad \mathbf{L}^{\top,-1}\epsilon_\theta(\mathbf{y}, s)\Bigg) \tag{110}$$

$$= \frac{1}{2}\Bigg(\mathbf{L}^{\top,-1}\Big[\epsilon - \epsilon_\theta(\mathbf{y}, s)\Big]\Bigg)^\top g^2_\phi(s)\Bigg(\mathbf{L}^{\top,-1}\Big[\epsilon - \epsilon_\theta(\mathbf{y}, s)\Big]\Bigg) \tag{111}$$

$$= \frac{1}{2}\Big[\epsilon - \epsilon_\theta(\mathbf{y}, s)\Big]^\top \underbrace{\mathbf{L}^{-1}g^2_\phi(s)\mathbf{L}^{\top,-1}}\Big[\epsilon - \epsilon_\theta(\mathbf{y}, s)\Big] \tag{112}$$

G ELBOS IN STATIONARY PARAMETERIZATION

We use the stationary parmeterization described in appendix B. We described the transitions for linear processes including the specialization of this to the stationary parameterization (appendix C). We derived two forms of the elbo, based on ISM and DSM (appendix E). We now specialize the ELBO to the linear stationary parameterization.

Recall $f_\phi(\mathbf{y}, s) = -[\mathbf{Q}_\phi(s) + \mathbf{D}_\phi(s)]\mathbf{y}$. Recall $g_\phi(s) = \sqrt{2\mathbf{D}_\phi(s)}$ We have $g^2_\phi(s) = 2\mathbf{D}_\phi(s)$. For the ISM elbo

$$\log p_\theta(x; \theta, \phi, \gamma)^{(ism)} \geq \mathbb{E}_{v\sim q_\gamma}\Bigg[\mathbb{E}_{s\sim \text{Unif}(0,T)}\Big[\ell_s^{(ism)}\Big] + \ell_T + \ell_q\Bigg] \tag{113}$$

We can set

$$
\begin{aligned}
\ell_{s_\theta} &= -\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s)\Big|\Big|^2_{\underbrace{2\mathbf{D}_\phi(s)}_{g_\phi^2}} \\
\ell_{\text{div-fgs}} &= \nabla_{\mathbf{y}_s} \cdot \Big[\underbrace{-[\mathbf{Q}_\phi(s) + \mathbf{D}_\phi(s)]\mathbf{y}_s}_{f_\phi} - \underbrace{2\mathbf{D}_\phi(s)}_{g_\phi^2} s_\theta(\mathbf{y}_s, s)\Big] \\
\ell_s^{\text{ism}} &= \mathbb{E}_{\underbrace{q_{\phi,s,(x,v)}}_{\text{depends on } \mathbf{Q},\mathbf{D}}}\Big[\ell_{s_\theta} + \ell_{\text{div-fgs}}\Big] \\
\ell_T &= \mathbb{E}_{\underbrace{q_{\phi,T},(x,v)}_{\text{depends on } \mathbf{Q},\mathbf{D}}}\Big[\log \pi_\theta(\mathbf{y}_T)\Big] \\
\ell_q &= -\log q_\gamma(v|x)
\end{aligned}
\tag{114}
$$

For the DSM form,

$$
\log p_\theta(x;\theta,\phi,\gamma)^{(dsm)} \geq \mathbb{E}_{v\sim q_\gamma}\left[\mathbb{E}_{s\sim \text{Unif}(0,T)}\left[\ell_s^{(dsm)}\right] + \ell_T + \ell_q\right] \tag{115}
$$

We can set

$$
\begin{aligned}
\ell_{\text{div-f}} &= \nabla_{\mathbf{y}_s} \cdot \underbrace{-[\mathbf{Q}_\phi(s) + \mathbf{D}_\phi(s)]\mathbf{y}_s}_{f_\phi} \\
\ell_{\text{fwd-score}} &= \frac{1}{2}\Big|\Big|\underbrace{\nabla_{\mathbf{y}_s}\log q_\phi(\mathbf{y}_s|\mathbf{y}_0)}_{\text{depends on } \mathbf{Q},\mathbf{D}}\Big|\Big|^2_{\underbrace{2\mathbf{D}_\phi(s)}_{g_\phi^2}} \\
\ell_{\text{neg-scorediff}} &= -\frac{1}{2}\Big|\Big|s_\theta(\mathbf{y}_s, s) - \underbrace{\nabla_{\mathbf{y}_s}\log q_\phi(\mathbf{y}_s|\mathbf{y}_0)}_{\text{depends on } \mathbf{Q},\mathbf{D}}\Big|\Big|^2_{\underbrace{2\mathbf{D}_\phi(s)}_{g_\phi^2}} \\
\ell_s^{(dsm)} &= \mathbb{E}_{\underbrace{q_{\phi,s,(x,v)}}_{\text{depends on } \mathbf{Q},\mathbf{D}}}\Big[\ell_{\text{neg-scorediff}} + \ell_{\text{fwd-score}} + \ell_{\text{div-f}}\Big]
\end{aligned}
$$

H ALGORITHMS

H.1 GENERIC TRANSITION KERNEL

Algorithm 1 Get transition distribution $\mathbf{y}_s|x$ or $\mathbf{y}_s|\mathbf{y}_0$

Input: data x. time s. $\mathbf{A}, g$. `hybrid` either true or false.
compute: $\mathbf{A}(s)$ and $g(s)$
compute: $\mathbf{M}_s = \int_0^s \mathbf{A}(t)dt$ (integrated drift)
compute: $\mathbf{N}_s = \int_0^s g^2(t)dt$ (integrated diffusions squared)
compute: $\gamma_{s|0} = \exp\left(\mathbf{M}_s\right)$ (mean coefficient)
if `hybrid` **then**
  set: $\mathbf{y}_0 = [x, 0_1, \ldots, 0_{K-1}]$, $\mathbf{\Sigma}_{0,zz} = \mathbf{0}$, and $\mathbf{\Sigma}_{0,zv}, \mathbf{\Sigma}_{0,vv}$ to chosen initial distribution
else
  sample: $\mathbf{y}_0^v \sim q_\phi(\mathbf{y}_0^v|\mathbf{y}_0^z)$
  set: $\mathbf{y}_0 = [x, \mathbf{y}_0^v]$, $\mathbf{\Sigma}_0 = \mathbf{0}$
end if
compute: $\mathbf{m}_{s|0} = \gamma_{s|0}\mathbf{y}_0$ (**mean**)
compute:

$$\begin{pmatrix}\mathbf{C}_s\\ \mathbf{H}_s\end{pmatrix} = \exp\left[\begin{pmatrix}\mathbf{M}_s & \mathbf{N}_s\\ \mathbf{0} & -\mathbf{M}_s^\top\end{pmatrix}\right]\begin{pmatrix}\mathbf{\Sigma}_0\\ \mathbf{I}\end{pmatrix} \quad \text{(ingredients for cov.)} \tag{116}$$

compute: $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s(\mathbf{H}_s)^{-1}$ (**cov.**)
Output: $\mathcal{N}(\mathbf{m}_{s|0}, \mathbf{\Sigma}_{s|0})$

H.2 TRANSITIONS WITH Q, D

Current param matrices $\tilde{\mathbf{Q}}_\phi, \tilde{\mathbf{D}}_\phi$ and along with fixed time-in scalar-out functions $b_q(s), b_d(s)$ and their known integrals $B_q(s), B_d(s)$. $q_\gamma(v_0|z_0 = x)$ taken to be parameterless for now $v_0 \sim \mathcal{N}(0, I)$. Model params are s_θ fixed π_θ.

Algorithm 2 Get $\mathbf{Q}, \mathbf{D}$ and their integrated terms $\mathbf{M}, \mathbf{N}$

Input: time s and current params ϕ
compute: $I_s[b_q] = \int_0^s b_q(\nu)d\nu$ using known integral $B_q(s) - B_q(0)$
compute: $I_s[b_d] = \int_0^s b_d(\nu)d\nu$ using known integral $B_d(s) - B_d(0)$.
compute: $I_s[\mathbf{Q}_\phi] = I_s[b_q] \cdot \left[\tilde{\mathbf{Q}}_\phi - \tilde{\mathbf{Q}}_\phi^\top\right]$ for current params $\tilde{\mathbf{Q}}_\phi$.
compute: $I_s[\mathbf{D}_\phi] = I_s[b_d] \cdot \left[\tilde{\mathbf{D}}_\phi\tilde{\mathbf{D}}_\phi^\top\right]$ for current params $\tilde{\mathbf{D}}_\phi$.
compute: $\mathbf{M}_s = -(I_s[\mathbf{Q}_\phi] + I_s[\mathbf{D}_\phi])$ ($\mathbf{M}$ just a variable name)
compute: $\mathbf{N}_s = I_s[2\mathbf{D}_\phi] = 2 \cdot I_s[\mathbf{D}_\phi]$ ($\mathbf{N}$ just a variable name)
compute: $\mathbf{Q}_s = b_q(s) \cdot \left[\tilde{\mathbf{Q}}_\phi - \tilde{\mathbf{Q}}_\phi^\top\right]$ (not integrated)
compute: $\mathbf{D}_s = b_d(s) \cdot \left[\tilde{\mathbf{D}}_\phi\tilde{\mathbf{D}}_\phi^\top\right]$ (not integrated)
compute: $A_s = -[\mathbf{Q}_s + \mathbf{D}_s]$ (drift coef.)
compute: $g_s^2 = 2\mathbf{D}_s$ (diffusion coef. squared)
Output: $\mathbf{A}_s, g_s^2, \mathbf{M}_s, \mathbf{N}_s$

H.3 ELBO ALGORITHMS

Algorithm 3 Get transition distributions

Input: Sample $\mathbf{y}_0 = (x, v)$ and time s. Current params ϕ
set: $\mathbf{A}_s, g_s^2, \mathbf{M}_s, \mathbf{N}_s \leftarrow$ algorithm 2
compute: $\mathbf{m}_{s|0} = \exp\Big(\mathbf{M}_s\Big)\mathbf{y}_0$ (transition mean)
compute: ingredients for transition cov. matrix:

$$\begin{pmatrix} \mathbf{C}_s \\ \mathbf{H}_s \end{pmatrix} = \exp\left[\begin{pmatrix} \mathbf{M}_s & \mathbf{N}_s \\ \mathbf{0} & -\mathbf{M}_s^\top \end{pmatrix}\right]\begin{pmatrix} \mathbf{0} \\ \mathbf{I} \end{pmatrix} \tag{117}$$

compute: $\mathbf{\Sigma}_{s|0} = \mathbf{C}_s(\mathbf{H}_s)^{-1}$ (transition cov).
instantiate: $q_{\phi,s,(x,v)} = q_\phi(\mathbf{y}_s|\mathbf{y}_0) = \mathcal{N}(\mathbf{m}_{s|0}, \mathbf{\Sigma}_{s|0})$.
Output: $q_{\phi,s,(x,v)}, A_s, g_s^2$

Algorithm 4 Compute ELBO with ism or dsm

input: Data point x and current params θ, ϕ, γ
draw: an aux. sample $v \sim q_\gamma(v|x)$
draw: a sample $s \sim \text{Unif}(0, T)$
set: $\mathbf{y}_0 = (x, v)$
set: $q_{\phi,s,\mathbf{y}_0}, A_s, g_s^2 \leftarrow$ algorithm 3 called on $\mathbf{y}_0, s, \phi$
draw: $\mathbf{y}_s \sim q_{\phi,s,\mathbf{y}_0}$
compute: ℓ_s with dsm(s) (algorithm 5) or ism(s) (algorithm 6) on $\mathbf{y}_s, \theta, A_s, g_s^2, q_{\phi,s,\mathbf{y}_0}$
set: $q_{\phi,T,\mathbf{y}_0}$, --, -- $\leftarrow$ algorithm 3 called on $\mathbf{y}_0, T, \phi$
draw: $\mathbf{y}_T \sim q_{\phi,T,\mathbf{y}_0}$
output: $\ell_s + \log \pi_\theta(\mathbf{y}_T) - \log q_\gamma(v)$

Algorithm 5 Compute dsm(s)

input: $\mathbf{y}_s$, θ, A_s, g_s^2, $q_{\phi,s,\mathbf{y}_0}$.
compute: fwd-score $= \nabla_{\mathbf{y}_s} \log q_\phi(\mathbf{y}_s|\mathbf{y}_0)$
compute: model-score $= s_\theta(\mathbf{y}_s, s)$
compute: fwd-score-term $= \frac{1}{2}$(fwd-score)$^\top g_s^2$(fwd-score)
compute: score-diff $=$ model-score $-$ fwd-score
compute: diff-term $= -\frac{1}{2}$score-diff$^\top g_s^2$score-diff
compute: div-f $= \nabla_{\mathbf{y}_s} \cdot A_s\mathbf{y}_s$
output: dsm(s) $=$ fwd-score-term $+$ diff-term $+$ div-f

Algorithm 6 Compute ism(s)

input: $\mathbf{y}_s$, θ, A_s, g_s^2, $q_{\phi,s,\mathbf{y}_0}$.
compute: model-score $= s_\theta(\mathbf{y}_s, s)$
compute: score-term $= -\frac{1}{2}$model-score$^\top g_s^2$model-score
compute: div-gs $= \nabla_{\mathbf{y}_s} \cdot g_s^2 s_\theta(\mathbf{y}_s, s)$
compute: div-f $= \nabla_{\mathbf{y}_s} \cdot A_s\mathbf{y}_s$
compute: div-term $= -$div-gs $+$ div-f
output: ism(s) $=$ score-term $+$ div-term