GREED IS GOOD: GUIDED GENERATION FROM A GREEDY PERSPECTIVE

Zander W. Blasingame*
Clarkson University

Chen Liu
Clarkson University

ABSTRACT

Training-free guided generation is a widely used and powerful technique that allows the end user to exert further control over the generative process of diffusion models. In this work, we explore the guided generation from the perspective of optimizing the solution trajectory of a neural differential equation in a greedy manner. We present such a strategy as a unifying view on training-free guidance by showing that the greedy strategy is a first-order discretization of end-to-end optimization techniques. We show that a greedy guidance strategy makes *good* decisions and compare it to a guidance strategy using the *ideal* gradients found via the continuous adjoint equations. We then show how other popular training-free guidance strategies can be viewed in a unified manner from this perspective.

1 INTRODUCTION

Guided generation greatly extends the utility of state-of-the-art generative models by allowing the end user to exert greater control over the generative process, ultimately making the tool more useful in a wide variety of applications ranging from conditional generation, editing of samples, inverse problems, *&c*. We focus on guided generation with diffusion/flow models (Song et al., 2021b; Lipman et al., 2023; Ho et al., 2020) as these models represent the current state-of-the-art across many different modalities, *e.g.*, audio (Liu et al., 2023a), images (Rombach et al., 2022), video (Blattmann et al., 2023), protein generation (Skreta et al., 2024), *&c*.

We can divide the techniques for guided generation into two broad categories: conditional training and training-free methods. The former of these two requires the training of the underlying diffusion/flow model on additional conditional information, either as a part of the training or at a later time as additional fine-tuning (Song et al., 2021a; Ho & Salimans, 2021; Hu et al., 2022). The latter category instead makes use of some known guidance function defined on the data distribution and incorporates this information back to the model to influence the generative process.

These training-free techniques can be further broken down into two sub-categories, *i.e.*, posterior sampling and end-to-end optimization. The former class of techniques uses a simple estimation of the posterior distribution that can be easily found in diffusion models (Chung et al., 2023) and *some* flow models (*cf*. Lipman et al., 2024, Section 4.8). This simple estimate of the posterior can then be fed to a guidance function to construct a gradient which can be brought back into the intermediate sampling state. This can then be used to update the ODE solve as a form of classifier guidance (Chung et al., 2023; Yu et al., 2023). The latter class of techniques, in contrast, performs backpropagation throughout the entire sampling process of the model to update the initial sample used in the solution trajectory via the guidance function. Many of these techniques (Blasingame & Liu, 2024a; Ben-Hamu et al., 2024; Pan et al., 2024; Marion et al., 2024; Wang et al., 2024) make use of the continuous adjoint equations (see Kidger, 2022, Section 5.1.2) to perform this backpropagation, this technique is known as optimize-then-discretize. However, some approaches (Liu et al., 2023c) prefer to perform the more memory-expensive, but potentially more accurate, discretize-then-optimize (see Kidger, 2022, Section 5.1.1) method, which amounts to performing backpropagation through the entire discretized sampling process. Finally, some works have viewed this through the lens of optimal control and introduce a control signal into the sampling process, which can be optimized through optimize-then-discretize (Wang et al., 2024) or discretize-then-optimize (Liu et al., 2023c).

*Correspondence to `blasinzw@clarkson.edu`

The aim of this paper is to present a unified view of these training-free techniques for guided generation from a *greedy perspective*.

> Our key insight is that we can *bridge* between techniques which use posterior sampling and techniques which use end-to-end optimization for guidance by viewing the former as a *greedy strategy* on the latter.

In light of this insight, we compare several state-of-the-art techniques from this perspective, showing how this perspective yields a unified and flexible framework for viewing guided generation with diffusion/flow models. We perform a detailed analysis of this greedy strategy, showing that it is not only a unifying view, but that it actually makes *good* decisions. We further illustrate how a greedy strategy can be viewed as a first-order method of an optimize-then-discretize or discreteize-then-optimize method, providing a flexible way to exchange compute costs with the needed accuracy of gradients.

2 Preliminaries

2.1 Diffusion Models

Diffusion models seek to learn a mapping from some simple prior distribution $p(\boldsymbol{x})$ to the data distribution $q(\boldsymbol{x})$. A forward process from the data distribution to the prior distribution is defined via an Itô Stochastic Differential Equation (SDE):

$$\mathrm{d}\boldsymbol{x}_t = f(t)\boldsymbol{x}_t \,\mathrm{d}t + g(t)\,\mathrm{d}\boldsymbol{w}_t, \tag{2.1}$$

on the time interval $[0, T]$ and where $\{\boldsymbol{w}_t\}_{t\in[0,T]}$ is the standard Wiener process. The *reverse-time* SDE (Anderson, 1982) is found to be:

$$\mathrm{d}\boldsymbol{x}_t = [f(t)\boldsymbol{x}_t + g^2(t)\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)]\,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\boldsymbol{w}}_t, \tag{2.2}$$

where $\mathrm{d}t$ is a *negative* timestep and $\{\bar{\boldsymbol{w}}_t\}_{t\in[0,T]}$ is the standard Wiener process in reverse time. The diffusion models then learn the score function, $\boldsymbol{s}_\theta(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t)$, (Song et al., 2021b) or some other closely related quantity, such as noise prediction (Song et al., 2021a; Ho et al., 2020) or data prediction (Kingma et al., 2021). Using this learned score function, we can then sample the *reverse-time* SDE, Equation (2.2), using numerical SDE solvers. Song et al. (2021b) showed that there exists an Ordinary Differential Equation (ODE) known as the *probability flow ODE* which shares the same marginals, $p_t(\boldsymbol{x})$, as Equation (2.2) given by

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = f(t)\boldsymbol{x}_t + \frac{g^2(t)}{2}\nabla_{\boldsymbol{x}} \log p_t(\boldsymbol{x}_t), \tag{2.3}$$

which also defined in *reverse-time*.

2.2 Flow Models

Flow models (Lipman et al., 2023) are another popular class of generative models that have theoretical overlaps with diffusion models. Consider two random variables: $\boldsymbol{X}_0 \sim p(\boldsymbol{x})$ and $\boldsymbol{X}_1 \sim q(\boldsymbol{x})$. Then consider a time-dependent vector field $\boldsymbol{u} \in \mathcal{C}^{1,r}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$[1] with $r \geq 1$ which determines[2] a time-dependent flow $\Phi_t \in \mathcal{C}^{1,r}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$ which satisfies the ODE

$$\Phi_0(\boldsymbol{x}) = \boldsymbol{x}, \quad \frac{\mathrm{d}}{\mathrm{d}t}\Phi_t(\boldsymbol{x}) = \boldsymbol{u}(t, \Phi_t(\boldsymbol{x})). \tag{2.4}$$

This is known as a $\mathcal{C}^r$-flow and this flow is diffeomorphism in its second argument for all $t \in [0, 1]$. Applying this flow to the random variable $\boldsymbol{X}_0$ we define a *continuous-time Markov process*

[1] For notational simplicity, we let $\mathcal{C}^{k_1,k_2,\ldots,k_n}(X_1 \times X_2 \times \cdots \times X_n; Y)$ denote the set of continuous functions that are k_i-times differentiable in the i-th argument mapping from $(X_1 \times X_2 \times \cdots \times X_n)$ to Y, if Y is omitted, then $Y = \mathbb{R}$.

[2] *N.B.*, to guarantee *global* existence and uniqueness a stronger constraint on the vector field, like global Lipschitzness or integrability is necessary, for further details we refer the reader to (Lipman et al., 2024).

$\{X_t\}_{t\in[0,1]}$ with mapping $\boldsymbol{X}_t = \Phi_t(\boldsymbol{X}_0)$. The *goal*, then, is to learn a flow Φ_t such that $\boldsymbol{X}_1 = \Phi_1(\boldsymbol{X}_0) \sim q(\boldsymbol{x})$. This procedure amounts to learning a neural network parameterized vector field $\boldsymbol{u}_\theta \in \mathcal{C}^{1,r}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$ which is achieved through training procedures known as *flow matching* (Lipman et al., 2023).

The probability flow ODE formulation of diffusion models can be viewed as a type of flow model with Gaussian probability paths, *i.e.*, $(\boldsymbol{X}_0, \boldsymbol{X}_1) \sim \pi_{0,1}(\boldsymbol{x}_0, \boldsymbol{x}_1) = p(\boldsymbol{x}_0)q(\boldsymbol{x}_1)$ with $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}|\mathbf{0}, \sigma^2\mathbf{I})$ (Lipman et al., 2024).

Time conventions. Although these two formulations are, essentially for our purposes, identical; we choose to adopt the time conventions of flow models over those of diffusion models. *I.e.*, we let time $t = 0$ denote the prior distribution and let time $t = 1$ denote the data distribution. In the opinion of the authors, this convention is more easily understood than the common time conventions in the literature on diffusion models.

2.3 The Continuous Adjoint Equations

Let $\boldsymbol{u}_\theta \in \mathcal{C}^{1,1}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$ be a model that models the vector field of some ODE and be Lipschitz continuous in its second argument. Let $\boldsymbol{x} : [0,1] \to \mathbb{R}^d$ be the solution to the ODE with the initial condition $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\dot{\boldsymbol{x}}_t = \boldsymbol{u}_\theta(t, \boldsymbol{x}_t)$. For some scalar-valued loss function $\mathcal{L} \in \mathcal{C}^2(\mathbb{R}^d)$ in $\boldsymbol{x}_1$, let $\boldsymbol{a}_{\boldsymbol{x}} := \partial\mathcal{L}/\partial\boldsymbol{x}_t$ denote the gradient. Then $\boldsymbol{a}_{\boldsymbol{x}}$ and related quantity $\boldsymbol{a}_\theta := \partial\mathcal{L}/\partial\theta$ can be found by solving an augmented ODE of the form,

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{x}}(1) &= \frac{\partial\mathcal{L}}{\partial\boldsymbol{x}_1}, & \frac{\mathrm{d}\boldsymbol{a}_{\boldsymbol{x}}}{\mathrm{d}t}(t) &= -\boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\boldsymbol{u}_\theta}{\partial\boldsymbol{x}}(t, \boldsymbol{x}_t), \\ \boldsymbol{a}_\theta(1) &= \mathbf{0}, & \frac{\mathrm{d}\boldsymbol{a}_\theta}{\mathrm{d}t}(t) &= -\boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\boldsymbol{u}_\theta}{\partial\theta}(t, \boldsymbol{x}_t), \end{aligned} \tag{2.5}$$

this system of equations are known as the *continuous adjoint equations* (Kidger, 2022). *N.B.*, this technique was first proposed by Pontryagin et al. (1963) and popularized for neural differential equations by Chen et al. (2018). Solving the continuous adjoint equations to find the gradients described by the adjoint state is known as an *optimize-then-discretize* method. This is in contrast with *discretize-then-optimize* methods which first discretize the neural differential equation before calculating the gradients (this can be thought of as "vanilla" backprop through a neural network). For a more detailed discussion between these two methods we refer to Kidger's monograph on neural differential equations (2022, Section 5.1)

3 Related Works

Training-free methods for diffusion/flow models is an active area of research with many different techniques proposed with differing strengths. Diffusion Posterior Sampling (DPS) (Chung et al., 2023) is a guidance method that uses Tweedie's formula (Stein, 1981) to estimate the gradient of some guidance function defined in the output state w.r.t. the noisy state, *i.e.*, $\mathbb{E}[\boldsymbol{x}_1|\boldsymbol{x}_t]$. Likewise the work of Bansal et al. (2023); Wang et al. (2023); Yu et al. (2023) explore similar concepts by employing Tweedie's formula for diffusion models, with FreeDoM including an additional "time-travel strategy" (Wang et al., 2023) to help improve the visual fidelity of the generated images. More relevant to our work, Blasingame & Liu (2024b) explore guidance using posterior sampling from a greedy perspective; however, their analysis is limited as their work is only focused on a particular experimental application. Specifically, they only consider a first-order numerical scheme for the diffusion ODE solver and they do not connect the greedy strategy to other guided generation methods. Liu et al. (2023c) provide guidance by adding an additional time-dependent control term to the vector field that is optimized using a type of discretize-then-optimize method by finding an efficient non-uniform Euler discretization of the sampling trajectory and perform backpropagation through this discretization to update the control signal.

Several recent works have explored the use of continuous adjoint equations for diffusion/flow models. Nie et al. (2022) was the first to explore this topic, solving the continuous adjoint equations for adversarial purification with diffusion SDEs. Later work by Pan et al. (2024; 2023) explore special solvers for the continuous adjoint equations of VP-type diffusion ODEs. Blasingame & Liu (2024a) extends these works by developing bespoke solvers for VP-type diffusion ODEs and SDEs. Marion

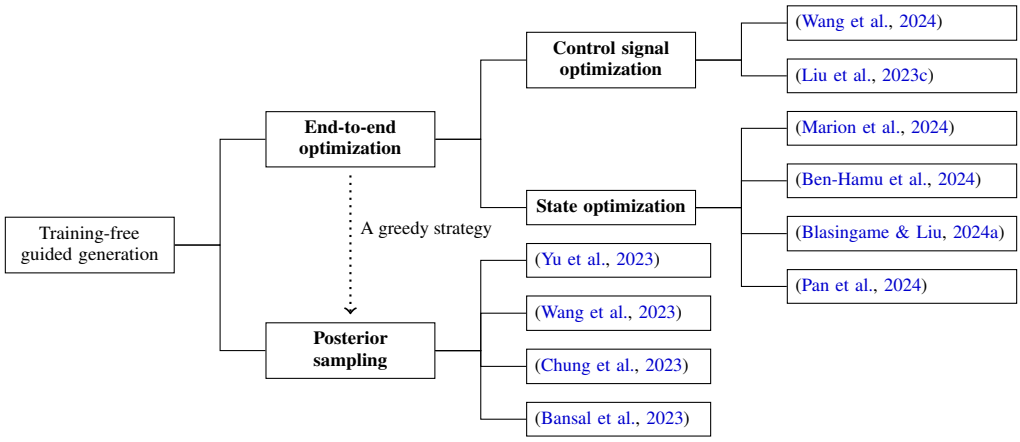

Figure 1: A taxonomy of *training-free guided generation* methods. We show that a greedy perspective is the *unifying bridge* between posterior and end-to-end methods, bring several different works together under one framework. In Section 5 we view posterior sampling techniques as a greedy strategy of end-to-end optimization techniques; moreover, in Section 5.1 we show how the greedy strategy is related to end-to-end optimization techniques as an explicit Euler scheme of a discretize-then-optimize method and an implicit Euler scheme of an optimize-then-discretize method see Theorem 5.2. Likewise, in Section 5.2 we present the greedy perspective on end-to-end optimization via a control signal. *N.B.*, end-to-end optimization techniques include both optimize-then-discretize and discretize-then-optimize techniques, although most fall into the former category.

et al. (2024) explore using the continuous adjoint equations as a part of a larger bi-level optimization scheme for guided generation. The work of Ben-Hamu et al. (2024) extends the analysis of continuous adjoint equations for diffusion models to flow-based models and provides an alternative perspective to the analyis performed by the earlier works. Wallace et al. (2023a) use EDICT (Wallace et al., 2023b), an invertible formulation of diffusion models, to perform backpropagation through the diffusion model; this can be viewed as a specific discretization scheme of continuous adjoint equations. Recent work by Wang et al. (2024) explores an extension of Ben-Hamu et al. (2024) to Riemannian manifolds which incorporates a control signal to the vector field and optimizes both the solution state and *co-state*, they call their approach OC-Flow.

4 A Motivation for Greed

Consider the case of Affine Gaussian Probability Paths (AGPP) and diffusion models in the ODE formulation with a data coupling, $(\boldsymbol{X}_0, \boldsymbol{X}_1) \sim p(\boldsymbol{x}_0)q(\boldsymbol{x}_1)$ with a Gaussian prior $p(\boldsymbol{x}_0)$. We define the new random variable $\boldsymbol{X}_t$ as an affine interpolation,

$$\boldsymbol{X}_t = \alpha_t \boldsymbol{X}_1 + \sigma_t \boldsymbol{X}_0, \tag{4.1}$$

where $\alpha_t, \sigma_t \in \mathcal{C}^\infty([0,1];[0,1])$ are smooth functions with boundary conditions $\alpha_0 = \sigma_1 = 0$ and $\alpha_1 = \sigma_0 = 1$ which define the *schedule* of $\{\boldsymbol{X}_t\}_{t\in[0,1]}$. Furthermore, α_t is strictly monotonically increasing in t; whereas, σ_t is strictly monotonically decreasing in t (Lipman et al., 2024). Then the *affine conditional flow* is defined as $\Phi_t(\boldsymbol{x}|\boldsymbol{x}_1) = \alpha_t \boldsymbol{x}_1 + \sigma_t \boldsymbol{x}$. It follows that the marginal vector field is given by

$$\boldsymbol{u}(t, \boldsymbol{x}) = \mathbb{E}[\dot{\alpha}_t \boldsymbol{X}_1 + \dot{\sigma}_t \boldsymbol{X}_0 | \boldsymbol{X}_t = \boldsymbol{x}], \tag{4.2}$$

where $\dot{\alpha}_t$ denotes the first derivative of α_t w.r.t. time. Now assume that we have trained our diffusion/flow model, $\boldsymbol{u}_\theta$, to zero loss.

4.1 Drawbacks of the Continuous Adjoint Equations

A natural choice to solve this problem is to apply the continuous adjoint equations to find the gradient $\partial\mathcal{L}/\partial\boldsymbol{x}_0$ and use that quantity for the optimization problem; however, there are several potential drawbacks to this approach which we will enumerate.

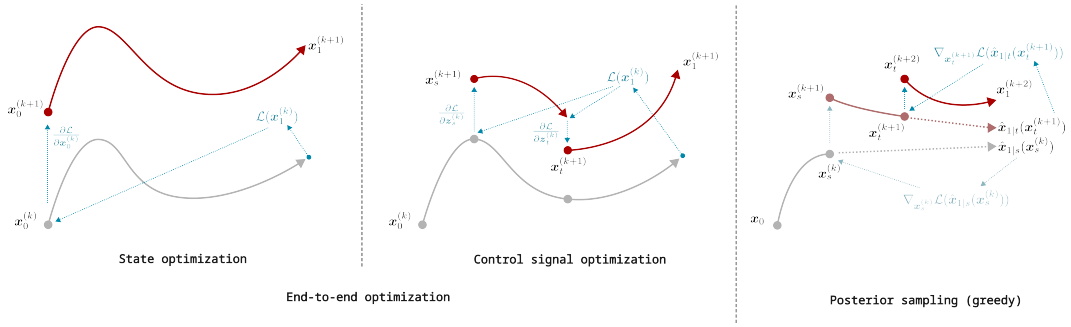

Figure 2: Visual comparison of different training-free guided generation techniques. For state optimization techniques (left) the gray curve represents the previous solution trajectory $\{\boldsymbol{x}_t^{(k)}\}_{t\in[0,1]}$ at the k-th iteration and the red curve shows the updated solution trajectory at the $(k+1)$-th iteration via the gradients in cyan. For control signal optimization techniques (middle) the solution trajectory is updated by changing the control signal $\boldsymbol{z}_t$, see Section 5.2, via the gradients $\partial\mathcal{L}/\partial\boldsymbol{z}_t$ calculated throughout the discretization scheme. For posterior sampling techniques (right) we use dull colors to denote the steps need to compute the $(k+1)$-th iteration and bright colors to denote the steps $(k+2)$-th iteration. *N.B.*, in contrast with the end-to-end optimization methods, posterior sampling methods quickly compute an estimate with the gradient with $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x})$ and do not backpropagate the gradient to any steps before the current timestep. The posterior sampling strategy makes a greedy decision at each timestep, hence we view these techniques as a greedy strategy of end-to-end optimization techniques. For all three diagrams, we let $0 < s < t < 1$.

Truncation errors. One area of concern is the potential mismatch between the forward trajectory $\{\boldsymbol{x}_{t_i}\}_{i=1}^N$ and the backward trajectory $\{\tilde{\boldsymbol{x}}_{t_i}\}_{i=1}^N$ when performing the backwards solve. One potential solution is to use an *algebraically reversible solver* (see Kidger et al., 2021; McCallum & Foster, 2024) which guarantees that the forward and backward trajectory match *perfectly*. Another option is to store the forward trajectory $\{\boldsymbol{x}_{t_i}\}_{i=1}^N$ in memory and use *interpolated adjoints* if the backward timesteps do not perfectly align with the forward timesteps (see Kim et al., 2021).

Stability concerns. Consider the simple ODE, $\dot{y}(t) = \lambda y(t)$ defined on $t \in [0, T]$ with $y(0) = y_0$ and $\lambda < 0$. Clearly, most ODE solvers with a non-trivial region of stability (see Harier & Wanner, 2002, Definition 2.1) will solve this ODE without an issue, as the errors will decrease exponentially with $\lambda < 0$. However, in the backwards in time solve from $y(T)$ the errors will *grow exponentially*. It can be shown that the adjoint state suffers from similar stability issues. The local behavior of a differential equation is described through the eigenvalues of the Jacobian of the vector field (see Butcher, 2016). For $\boldsymbol{x}_t$ this is given by $\frac{\partial \boldsymbol{u}_\theta}{\partial \boldsymbol{x}}$ and for $\boldsymbol{a}_{\boldsymbol{x}}$ this is given by

$$\frac{\partial}{\partial \boldsymbol{a}_{\boldsymbol{x}}}\left(- \boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial \boldsymbol{u}_\theta}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_t)\right) = -\frac{\partial \boldsymbol{u}_\theta}{\partial \boldsymbol{x}}(t, \boldsymbol{x}_t). \tag{4.3}$$

Clearly, the Jacobians for $\boldsymbol{a}_{\boldsymbol{x}}$ and $\boldsymbol{x}_t$ solved in reverse-time are identical, meaning the stability of the backward solve is pushed onto the solve for the adjoint state (see Kidger, 2022, Section 5.1.2) for more details. Reversible solvers eliminate truncation errors, but tend to suffer from poor stability, *e.g.*, the region of stability for reversible Heun applied to neural ODEs is the complex interval $[-i, i]$ (Kidger et al., 2021). Recent work by McCallum & Foster (2024), however, has shown a strategy for constructing reversible solvers with a non-trivial region of stability.

Computational cost. Clearly, by construction guidance procedures which solve the continuous adjoint equations[3] are much more computationally expensive than those which use some form of posterior optimization.

[3] *N.B.*, this would also include methods like FlowGrad which perform a full backpropagation through the ODE solver, but do not necessarily solve the continuous adjoint equations.

5 A GREEDY PERSPECTIVE

The computational cost of adjoint methods can be quite high, as is admitted in papers exploring this method (Ben-Hamu et al., 2024; Blasingame & Liu, 2024a; Pan et al., 2024; Wang et al., 2024). As such, it is worth considering whether techniques such as FreeDoM (Yu et al., 2023) and DPS (Chung et al., 2023), which are much cheaper, can be used as a reasonable alternative to techniques that use adjoint methods. In other words, we ask the following.

> Rather than performing the full optimization back to $\boldsymbol{x}_0$, what if we greedily took an optimal step at each $\boldsymbol{x}_t$ instead?

The next question is how to define a locally optimal step at time t. For this purpose we can use posterior sampling, which is defined as $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1 | \boldsymbol{X}_t = \boldsymbol{x}]$, Note that this is also commonly referred to as the denoiser, data prediction model, or $\boldsymbol{x}_1$-prediction model (Lipman et al., 2024). This has the advantage of being easily calculated from the $\boldsymbol{x}_0$-prediction model or the model of the vector field, requiring little computation to calculate this quantity. Then using this estimate of the posterior, we can use a guidance function defined in the data space and take the gradient of the loss w.r.t. as the local step, *i.e.*, use $\nabla_{\boldsymbol{x}} \mathcal{L}(\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}))$ as guidance. As such, we have our connection between a greedy strategy and *posterior sampling* approaches.

5.1 GREED AS A UNIFYING THEORY OF GUIDANCE

By construction we have our connection between the greedy strategy and posterior sampling methods. Next, we will establish the connection between posterior sampling methods and the much larger class of end-to-end optimization methods. To make our analysis simpler, let us write the flow from s to t in terms of the denoiser. The vector field can be written as a function of the denoiser with,

$$\boldsymbol{u}(t, \boldsymbol{x}) = \underbrace{\frac{\dot{\sigma}_t}{\sigma_t}}_{:=a_t} \boldsymbol{x} + \underbrace{\dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}}_{:=b_t} \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}). \tag{5.1}$$

The flow from time s to time t can then be expressed as the integral of the right-hand side of Equation (5.1) over time. Thus, the flow is now expressed as a semi-linear integral equation with linear term $a_t \boldsymbol{x}$ and non-linear term $b_t \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x})$. Due to this semi-linear structure, we can apply the technique of *exponential integrators* Hochbruck & Ostermann (2010) that has been successfully used to simplify numerical solvers for diffusion models (Lu et al., 2022a; Zhang & Chen, 2023; Gonzalez et al., 2024).

Let $\gamma_t = \gamma(t) \coloneqq \alpha_t / \sigma_t$ denote the signal-to-noise ratio (SNR), then γ_t is a monotonically increasing sequence in t, due to the properties of (α_t, σ_t) and thus has an inverse t_γ such that $t_\gamma(\gamma(t)) = t$. With abuse of notation, we let $\boldsymbol{x}_\gamma \coloneqq \boldsymbol{x}_{t_\gamma(\gamma)}$ and $\hat{\boldsymbol{x}}_{1|\gamma}(\cdot) = \hat{\boldsymbol{x}}_{1|t_\gamma(\gamma)}(\cdot)$. As such, we can rewrite the solution to the flow model in terms of γ by making use of exponential integrators, which we show in Proposition 5.1 with the full proof provided in Appendix B.

> **Proposition 5.1** (Exact solution of affine probability paths)**.** *Given an initial value of $\boldsymbol{x}_s$ at time $s \in [0, 1]$ the solution $\boldsymbol{x}_t$ at time $t \in [0, 1]$ of an ODE governed by the vector field in Equation (4.2) is:*
>
> $$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s} \boldsymbol{x}_s + \sigma_t \int_{\gamma_s}^{\gamma_t} \hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma) \, \mathrm{d}\gamma. \tag{5.2}$$

N.B., our result bears some similarity to the exact solution of diffusion ODEs that use a denoiser model; however, they integrate w.r.t. one-half of the logarithmic SNR (*cf*. Lu et al., 2022b).

Discretize-then-optimize. From Proposition 5.1 we can see that using the denoiser to estimate $\boldsymbol{x}_1$ is akin to taking a first-order approximation of the flow. More specifically, we can construct a $(k-1)$-th Taylor expansion of Equation (5.2) with:

$$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s} \boldsymbol{x}_s + \sigma_t \sum_{n=0}^{k-1} \frac{\mathrm{d}^n}{\mathrm{d}\gamma^n} \left[\hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma) \right]_{\gamma=\gamma_s} \frac{h^{n+1}}{n!} + \mathcal{O}(h^{k+1}), \tag{5.3}$$

where $h := \gamma_t - \gamma_s$ is the step size. Then it follows that for $k = 1$ the first-order discretization of the flow, omitting high-order error terms becomes,

$$\tilde{\boldsymbol{x}}_t = \frac{\sigma_t}{\sigma_s}\boldsymbol{x}_s + (\alpha_t + \frac{\sigma_t \alpha_s}{\sigma_s})\hat{\boldsymbol{x}}_{1|s}(\boldsymbol{x}_s). \tag{5.4}$$

In the limit as $t \to 1$ we have $\tilde{\boldsymbol{x}}_t = \hat{\boldsymbol{x}}_{1|s}(\boldsymbol{x}_s)$, note despite $\sigma_t \to 0$ the asymptotic behavior is well-defined (see Ben-Hamu et al., 2024). Thus we can view the gradients produced by a greedy strategy as an discretize-then-optimize technique for an explicit Euler scheme with step size h.

Optimize-then-discretize. Next we consider the continuous *ideal* of the gradients of the solution trajectory, *i.e.*, $\{\boldsymbol{a}_{\boldsymbol{x}}(t)\}_{t\in[0,1]}$. We then explore how the gradients calculated via the greedy strategy compare to this continuous ideal of the gradient flow. In Theorem 5.2 we show that a greedy strategy can be viewed as the first iteration of a fixed-point method of an implicit Euler discretization of the continous adjoint equations.

Theorem 5.2 (Greedy as an implicit Euler method)**.** *For some trajectory state $\boldsymbol{x}_t$ at time t, the greedy gradient given by $\nabla_{\boldsymbol{x}_t}\mathcal{L}(\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t))$ is an implicit Euler discretization of the continuous adjoint equations for the true gradients with step size $h = \gamma_1 - \gamma_t$.*

Proof sketch. First we use the technique of exponential integrators to simplify the continuous adjoint equations. Then we perform a first-order Taylor expansion around γ_t which is equivalent to an implicit Euler scheme as we are calculating the gradient flow from 1 to t. The full proof is provided in Appendix D.

An important question is if a greedy strategy makes *good* decisions at each timestep. *I.e.*, if we make a good decision at time t, does that ensure that an optimal solution was made in the sense of $\boldsymbol{x}_1 = \Phi_{1|t}(\boldsymbol{x}_t)$, where $\Phi_{t|s}$ denotes the flow from time s to time t. A natural way to examine this question is to consider whether convergence in the local case implies convergence of the whole solution trajectory. We find that up to a bound dependent on the step size convergence in the greedy solution implies convergence in the flow, which we state more formally in Theorem 5.3.

Theorem 5.3 (Greedy convergence)**.** *For affine probability paths, if there exists a sequence of states $\boldsymbol{x}_n$ at time t such that it converges to the locally optimal solution $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) \to \boldsymbol{x}_1^*$. Then, $\|\Phi_{1|t}(\boldsymbol{x}_n) - \boldsymbol{x}_1^*\|$ is $\mathcal{O}(h^2)$ as $n \to \infty$.*

Proof sketch. First, we take a first-order Taylor expansion of the flow described in Equation (5.2) which as shown above simplifies to the denoiser plus $\mathcal{O}(h^2)$ truncation error terms. Clearly, if $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) \to \boldsymbol{x}_1^*$ then the first term of the Taylor expansion does as well. The full proof is provided in Appendix C.

Note, the tightness on the convergence of the flow from the greedy strategy is controlled by the step size, but also the geometry of the ODE. Clearly, if the curvature[4] was zero, *i.e.*, the paths were straight, then there would be no truncation error and convergence in greedy would necessarily imply convergence in flow. This positive result shows that a greedy strategy makes good decisions that justify the potential application of such a technique over the more expensive adjoint methods. This complements the observations of Ben-Hamu et al. (2024) who report they only needed a few discretization steps of the continuous adjoint equations to achieve adequate performance.

Importantly, because the flow is a diffeomorphism, we can use the optimal solution at time $t \in [0, 1]$ to find the optimal solution at any time $s \in [0, 1]$. This combined with our upper bound in the error shows that a greedy strategy could be in theory used to solve problems which seek to find the optimal $\boldsymbol{x}_0^*$. *N.B.*, in practice we still need to discretize the flow Φ_t which introduces error in estimating $\boldsymbol{x}_0$ from $\boldsymbol{x}_t$.

[4]The curvature (see Finlay et al., 2020; Liu et al., 2023b; Lee et al., 2023) can be defined as

$$S(\boldsymbol{X}) = \int_0^1 \mathbb{E}\left[\left\|(\boldsymbol{X}_1 - \boldsymbol{X}_0) - \dot{\boldsymbol{X}}_t\right\|^2\right] \mathrm{d}t.$$

5.2 Guidance via a Control Signal

Some works on guidance (Liu et al., 2023c; Wang et al., 2024) have considered the problem from the perspective of optimal control. In essence, inject an additional control signal, $\boldsymbol{z} \in \mathcal{C}^1(\mathbb{R}; \mathbb{R}^d)$, to the vector field, $\boldsymbol{u}_\theta$, such that

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{u}_\theta(t, \boldsymbol{x}_t) + \boldsymbol{z}(t). \tag{5.5}$$

Thus, instead of optimizing $\{\boldsymbol{x}_t\}_{t\in[0,t]}$ directly, this control signal can instead be optimized, serving as one of the key insights in (Liu et al., 2023c; Wang et al., 2024). We can model the gradient to this signal by augmenting the continuous adjoint equations with the adjoint state $\boldsymbol{a}_{\boldsymbol{z}}(t) \coloneqq \partial\mathcal{L}/\partial\boldsymbol{z}(t)$. In Theorem 5.4 we show that this gradient is simply an integral of the adjoint state $\boldsymbol{a}_{\boldsymbol{x}}(t)$, the full proof can be found in Appendix E.

Theorem 5.4 (Continuous adjoint equations for the control term)**.** *Let $\boldsymbol{u}_\theta \in \mathcal{C}^{1,1}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$ be a parameterization of some time-dependent vector field of a neural ODE that is Lipschitz continuous in its second argument, and let $\boldsymbol{z} \in \mathcal{C}^1([0,1]; \mathbb{R}^d)$ be an additional control signal such that the new dynamics are given by Equation (5.5). Let $\boldsymbol{a}_{\boldsymbol{z}}(t) \coloneqq \partial\mathcal{L}/\partial\boldsymbol{z}(t)$ then*

$$\boldsymbol{a}_{\boldsymbol{z}}(t) = -\int_1^t \boldsymbol{a}_{\boldsymbol{x}}(s)\,\mathrm{d}s. \tag{5.6}$$

This result can be viewed as a more generalized version of Wang et al. (2024, Theorem 2) (*cf*. Liu et al., 2023c, Equation (8)).

The next natural question then is to ask about the behavior of a greedy strategy applied to $\boldsymbol{z}(t)$. To simplify the analysis, we now consider a control signal applied to the posterior model $\hat{\boldsymbol{x}}_{1|t}$ such that it is replaced by $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t) + \boldsymbol{z}(t)$ which amounts to simply rescaling $\boldsymbol{z}(t)$ from Equation (5.5) with b_t. From this construction it should be clear that the greedy gradient for the control signal is merely $\nabla_{\tilde{\boldsymbol{x}}_1}\mathcal{L}(\tilde{\boldsymbol{x}}_1)$. If using the original formulation where the control signal is applied to the vector field, rather than the denoiser, the gradient is simply scaled by a weighting function dependent on time. Note this approach is similar to the greedy approach taken by Blasingame & Liu (2024b); however, they inject the control signal onto the $\boldsymbol{x}_0$-prediction model.

6 Conclusion

In this work we show that posterior sampling methods can be viewed as a greedy strategy of end-to-end optimization methods, creating a unified framework for examining training-free guided generation techniques. More specifically, we show that posterior sampling methods are simply a single-step first-order discretization of either discretize-then-optimize or optimize-then-discretize methods. Hence we can move between the two different strategies by simply altering the number of discretization steps. Moreover, we show that a greedy strategy can actually make *good* decisions, justifying the choice of such a strategy over the more computationally intensive end-to-end optimization techniques in certain scenarios. Our hope is that this unified perspective on guided generation can inform future research directions when applying training-free guided generation to different applications.

Limitations. In this work we only consider affine probability paths which includes many flow models and diffusion models in the ODE formulation; however, we did not explore a greedy strategy for SDE based generative models. With future work, we could likely tighten the bound in the convergence proof by incorporating the curvature of the ODE. We did not perform a sensitivity analysis of how the flow changes w.r.t. to the greedy gradients. As this work was theoretical and focused on drawing connections between existing guided generation techniques, we did not include experimental work.

References

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

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 843–852, 2023.

Heli Ben-Hamu, Omri Puny, Itai Gat, Brian Karrer, Uriel Singer, and Yaron Lipman. D-flow: Differentiating through flows for controlled generation. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=SE20BFqj6J.

Zander W. Blasingame and Chen Liu. AdjointDEIS: Efficient gradients for diffusion models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 2449–2483. Curran Associates, Inc., 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/04badd3b048315c8c3a0ca17eff723d7-Paper-Conference.pdf.

Zander W. Blasingame and Chen Liu. Greedy-dim: Greedy algorithms for unreasonably effective face morphs. In *2024 IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–11, 2024b. doi: 10.1109/IJCB62174.2024.10744517.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023.

John Charles Butcher. *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016. Third Edition.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. The International Conference on Learning Representations, 2023.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: the world of jacobian and kinetic regularization. In *International conference on machine learning*, pp. 3154–3164. PMLR, 2020.

Martin Gonzalez, Nelson Fernandez Pinto, Thuy Tran, Hatem Hajri, Nader Masmoudi, et al. Seeds: Exponential sde solvers for fast high-quality sampling from diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Ernst Harier and Gerhard Wanner. *Solving Ordinary Differential Equations II Stiff and Differential-Algebraic Problems.* Springer, Berlin, 2002. Second Revised Edition.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Patrick Kidger. *On Neural Differential Equations*. Ph.d. thesis, Oxford University, 2022. Available at https://arxiv.org/abs/2202.02435.

Patrick Kidger, James Foster, Xuechen Chen Li, and Terry Lyons. Efficient and accurate gradients for neural sdes. *Advances in Neural Information Processing Systems*, 34:18747–18761, 2021.

Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9), 2021.

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

Sangyun Lee, Beomsu Kim, and Jong Chul Ye. Minimizing trajectory curvature of ode-based generative models. In *International Conference on Machine Learning*, pp. 18957–18973. PMLR, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, pp. 21450–21474. PMLR, 2023a.

Xingchao Liu, Chengyue Gong, and qiang liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=XVjTT1nw5z.

Xingchao Liu, Lemeng Wu, Shujian Zhang, Chengyue Gong, Wei Ping, and Qiang Liu. Flowgrad: Controlling the output of generative odes with gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24335–24344, 2023c.

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. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022a.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022b.

Pierre Marion, Anna Korba, Peter Bartlett, Mathieu Blondel, Valentin De Bortoli, Arnaud Doucet, Felipe Llinares-López, Courtney Paquette, and Quentin Berthet. Implicit diffusion: Efficient optimization through stochastic sampling. *arXiv preprint arXiv:2402.05468*, 2024.

Sam McCallum and James Foster. Efficient, accurate and stable gradients for neural odes. *arXiv preprint arXiv:2410.11648*, 2024.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16805–16827. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/nie22a.html.

Jiachun Pan, Hanshu Yan, Jun Hao Liew, Jiashi Feng, and Vincent YF Tan. Towards accurate guided diffusion sampling through symplectic adjoint method. *arXiv preprint arXiv:2312.12030*, 2023.

Jiachun Pan, Jun Hao Liew, Vincent Tan, Jiashi Feng, and Hanshu Yan. AdjointDPM: Adjoint sensitivity method for gradient backpropagation of diffusion probabilistic models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=y33lDRBgWI.

Lev Semenovich Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishechenko. The mathematical theory of optimal processes. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, 43(10-11):514–515, 1963. doi: https://doi.org/10.1002/zamm.19630431023. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19630431023.

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.

Marta Skreta, Lazar Atanackovic, Avishek Joey Bose, Alexander Tong, and Kirill Neklyudov. The superposition of diffusion models using the Itô density estimator. *arXiv preprint arXiv:2412.17762*, 2024.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=St1giarCHLP.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=PxTIG12RRHS.

Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pp. 1135–1151, 1981.

Bram Wallace, Akash Gokul, Stefano Ermon, and Nikhil Naik. End-to-end diffusion latent optimization improves classifier guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7280–7290, 2023a.

Bram Wallace, Akash Gokul, and Nikhil Naik. Edict: Exact diffusion inversion via coupled transformations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22532–22541, 2023b.

Luran Wang, Chaoran Cheng, Yizhen Liao, Yanru Qu, and Ge Liu. Training free guided flow matching with optimal control. *arXiv preprint arXiv:2410.18070*, 2024.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-shot image restoration using denoising diffusion null-space model. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mRieQgMtNTQ.

Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23174–23184, 2023.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Loek7hfb46P.

A Review of Our Nomenclature

To enhance the readability of our work, we provide a quick summary of our nomenclature and preferred conventions. We use a bold uppercase letter $\boldsymbol{X}_t \sim p_t(\boldsymbol{x})$ to denote a random variable distributed according to $p_t(\boldsymbol{x})$ and $\boldsymbol{x}_t$ to denote the particular realization of said random variable. We use the hat symbol to denote a prediction function, *e.g.*, we let

$$\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_1 | \boldsymbol{X}_t = \boldsymbol{x}], \tag{A.1}$$

$$\hat{\boldsymbol{x}}_{0|t}(\boldsymbol{x}) = \mathbb{E}[\boldsymbol{X}_0 | \boldsymbol{X}_t = \boldsymbol{x}], \tag{A.2}$$

denote the $\boldsymbol{x}_1$-prediction (data) and $\boldsymbol{x}_0$-prediction (noise) *deterministic* functions, respectively.

B Proof of Proposition 5.1

We restate Proposition 5.1 here:

Proposition B.1 (Exact solution of affine probability paths)**.** *Given an initial value of $\boldsymbol{x}_s$ at time $s \in [0, 1]$, the solution $\boldsymbol{x}_t$ at time $t \in [0, 1]$ of an ODE governed by the vector field in Equation (4.2) is:*

$$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s}\boldsymbol{x}_s + \sigma_t \int_{\gamma_s}^{\gamma_t} \hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)\, \mathrm{d}\gamma. \tag{B.1}$$

Additionally, we assume that the following holds:

Assumption B.2. The function $a_t := \frac{\dot{\sigma}_t}{\sigma_t}$ is integrable in $[0, 1]$.

This assumption is necessary for the simplification that we perform with exponential integrators. *N.B.*, Ben-Hamu et al. (2024) make the same assumption in their analysis of the continuous adjoint equations for affine probability paths.

Proof. Recall that we uniquely define a flow model through the vector field $\boldsymbol{u} \in \mathcal{C}^{1,1}([0, 1] \times \mathbb{R}^d; \mathbb{R}^d)$. The vector field which models the affine conditional flow with schedule (α_t, σ_t), (see Section 4), is defined as

$$\boldsymbol{u}(t, \boldsymbol{x}) = \mathbb{E}[\dot{\alpha}_t \boldsymbol{X}_1 + \dot{\sigma}_t \boldsymbol{X}_0 | \boldsymbol{X}_t = \boldsymbol{x}]. \tag{B.2}$$

With some simple algebra we can rewrite the vector field in terms of $\hat{\boldsymbol{x}}_{1|t}$,

$$\boldsymbol{u}(t, \boldsymbol{x}) = a_t \boldsymbol{x} + b_t \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}), \tag{B.3}$$

$$a_t = \frac{\dot{\sigma}_t}{\sigma_t} \qquad b_t = \dot{\alpha}_t - \alpha_t \frac{\dot{\sigma}_t}{\sigma_t}.$$

Now using this definition we can rewrite the solution for $\boldsymbol{x}_t$ from $\boldsymbol{x}_s$ in terms of $\hat{\boldsymbol{x}}_{1|t}$,

$$\boldsymbol{x}_t = \boldsymbol{x}_s + \int_s^t \boldsymbol{u}(\tau, \boldsymbol{x}_\tau)\, \mathrm{d}\tau, \tag{B.4}$$

$$\boldsymbol{x}_t = \boldsymbol{x}_s + \int_s^t a_\tau \boldsymbol{x}_\tau + b_\tau \hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)\, \mathrm{d}\tau. \tag{B.5}$$

Note the semi-linear form of the integral equation. We can exploit this structure using the technique of *exponential integrators*, (see Lu et al., 2022a; Zhang & Chen, 2023; Gonzalez et al., 2024), to simplify Equation (B.5), under Assumption B.2, to

$$\boldsymbol{x}_t = e^{\int_s^t a_u \, \mathrm{d}u} \boldsymbol{x}_s + \int_s^t e^{\int_\tau^t a_u \, \mathrm{d}u} b_\tau \hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)\, \mathrm{d}\tau. \tag{B.6}$$

Now, the integrating factor simplifies quite nicely to

$$e^{\int_s^t a_u \, \mathrm{d}u} = e^{\int_s^t \frac{\dot{\sigma}_u}{\sigma_u} \, \mathrm{d}u} = e^{\int_{\sigma_s}^{\sigma_t} \frac{1}{\sigma} \, \mathrm{d}\sigma} = \frac{\sigma_t}{\sigma_s}, \tag{B.7}$$

such that Equation (B.6) becomes

$$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s}\boldsymbol{x}_s + \sigma_t \int_s^t \frac{b_\tau}{\sigma_\tau} \hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)\, \mathrm{d}\tau. \tag{B.8}$$

We can simplify b_t/σ_t to find:

$$\frac{b_t}{\sigma_t} = \frac{\dot{\alpha}_t\sigma_t - \alpha_t\dot{\sigma}_t}{\sigma_t^2} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\alpha_t}{\sigma_t}\right) = \frac{\mathrm{d}}{\mathrm{d}t}\gamma_t, \tag{B.9}$$

where $\gamma_t := \alpha_t/\sigma_t$, *i.e.*, the signal-to-noise ratio. As such, we can rewrite Equation (B.8) with a change of variables $\boldsymbol{x}_\gamma = \boldsymbol{x}_{\gamma_t^{-1}(\gamma)} = \boldsymbol{x}_t$,

$$\boldsymbol{x}_t = \frac{\sigma_t}{\sigma_s}\boldsymbol{x}_s + \sigma_t \int_{\gamma_s}^{\gamma_t} \hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)\,\mathrm{d}\gamma, \tag{B.10}$$

concluding the proof. □

C Proof of Theorem 5.3

We restate Theorem 5.3 here:

Theorem C.1 (Greedy convergence)**.** *For affine probability paths, if there exists a sequence of states $\boldsymbol{x}_n$ at time t such that it converges to the locally optimal solution $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) \to \boldsymbol{x}_1^*$. Then, $\|\Phi_{1|t}(\boldsymbol{x}_n) - \boldsymbol{x}_1^*\|$ is $\mathcal{O}(h^2)$ as $n \to \infty$.*

Throughout the proof the norm $\|\cdot\|$ corresponds to the Euclidean norm $\|\cdot\|_2$. Additionally, we make the following (mild) regularity assumptions:

Assumption C.2. The $\boldsymbol{x}_1$-prediction model $\hat{\boldsymbol{x}}_{1|t} : [0,1] \times \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz in its second argument and continuous in its first.

Assumption C.3. The total derivatives $\frac{\mathrm{d}^n \hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x})}{\mathrm{d}\gamma^n}$ exist and are continuous for $0 \le n \le k+1$.

Proof. Let $\boldsymbol{x}_1 = \Phi_{1|t}(\boldsymbol{x}_t)$. Then by Assumption C.3, we can take a $(k-1)$-th order Taylor expansion around γ_t of the flow:

$$\begin{aligned}
\boldsymbol{x}_1 &= \frac{\sigma_1}{\sigma_t}\boldsymbol{x}_t + \sigma_1 \int_{\gamma_t}^{\gamma_1} \sum_{n=0}^{k-1} \frac{\mathrm{d}^n}{\mathrm{d}\gamma^n}\left[\hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)\right]_{\gamma=\gamma_t} \frac{(\gamma-\gamma_t)^n}{n!}\,\mathrm{d}\gamma + \mathcal{O}(h^{k+1}),\\
&= \frac{\sigma_1}{\sigma_t}\boldsymbol{x}_t + \sigma_1 \sum_{n=0}^{k-1} \frac{\mathrm{d}^n}{\mathrm{d}\gamma^n}\left[\hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)\right]_{\gamma=\gamma_t} \int_{\gamma_t}^{\gamma_1} \frac{(\gamma-\gamma_t)^n}{n!}\,\mathrm{d}\gamma + \mathcal{O}(h^{k+1}),\\
&= \frac{\sigma_1}{\sigma_t}\boldsymbol{x}_t + \sigma_1 \sum_{n=0}^{k-1} \frac{\mathrm{d}^n}{\mathrm{d}\gamma^n}\left[\hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)\right]_{\gamma=\gamma_t} \frac{h^{n+1}}{(n+1)!} + \mathcal{O}(h^{k+1}),
\end{aligned} \tag{C.1}$$

where $h := \gamma_1 - \gamma_t$ is the stepsize. Let $k = 1$, then we have:

$$\boldsymbol{x}_1 = \frac{\sigma_1}{\sigma_t}\boldsymbol{x}_n + \sigma_1\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n)h + \mathcal{O}(h^2), \tag{C.2}$$

$$= \frac{\sigma_1}{\sigma_t}\boldsymbol{x}_n + (\alpha_1 - \frac{\sigma_1\alpha_t}{\sigma_t})\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) + \mathcal{O}(h^2). \tag{C.3}$$

By definition $\sigma_1 = 0$ and $\alpha_1 = 1$, then

$$\boldsymbol{x}_1 = \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) + \mathcal{O}(h^2), \tag{C.4}$$

which is equivalent to

$$\|\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n)\| \le C_1 h^2, \tag{C.5}$$

for some constant $C_1 > 0$. Since $\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n) \to \boldsymbol{x}_1^*$ we know that for any $\epsilon > 0$ there exists some $n \ge N$ such that $\|\boldsymbol{x}_1^* - \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n)\| < \epsilon$. Thus,

$$\|\boldsymbol{x}_1 - \boldsymbol{x}_1^*\| \le \|\boldsymbol{x}_1 - \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n)\| + \|\boldsymbol{x}_1^* - \hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_n)\| < \underbrace{\epsilon + C_1 h^2}_{:=C_2}. \tag{C.6}$$

Thus the error between the optimal solution $\boldsymbol{x}_1^*$ and the solution from the flow $\boldsymbol{x}_1$ is $\mathcal{O}(h^2)$ thereby finishing the proof. □

D Proof of Theorem 5.2

We restate Theorem 5.2 here:

Theorem D.1 (Greedy as an implicit Euler method)**.** *For some trajectory state $\boldsymbol{x}_t$ at time t, the greedy gradient given by $\nabla_{\boldsymbol{x}_t}\mathcal{L}(\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t))$ is an implicit Euler discretization of the continuous adjoint equations for the true gradients with step size $h = \gamma_1 - \gamma_t$.*

For clarity we restate the definition of the continuous adjoint equations. Let $\boldsymbol{u}_\theta \in \mathcal{C}^{1,1}([0,1]\times\mathbb{R}^d;\mathbb{R}^d)$ be a model that models the vector field of some ODE and be Lipschitz continuous in its second argument. Let $\boldsymbol{x} : [0,1] \to \mathbb{R}^d$ be the solution to the ODE with the initial condition $\boldsymbol{x}_0 \in \mathbb{R}^d$, $\dot{\boldsymbol{x}}_t = \boldsymbol{u}_\theta(t, \boldsymbol{x}_t)$. For some scalar-valued loss function $\mathcal{L} \in \mathcal{C}^2(\mathbb{R}^d)$ in $\boldsymbol{x}_1$, let $\boldsymbol{a}_{\boldsymbol{x}} := \partial\mathcal{L}/\partial\boldsymbol{x}_t$ denote the gradient. Then $\boldsymbol{a}_{\boldsymbol{x}}$ and related quantity $\boldsymbol{a}_\theta := \partial\mathcal{L}/\partial\theta$ can be found by solving an augmented ODE of the form,

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{x}}(1) &= \frac{\partial\mathcal{L}}{\partial\boldsymbol{x}_1}, & \frac{\mathrm{d}\boldsymbol{a}_{\boldsymbol{x}}}{\mathrm{d}t}(t) &= -\boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\boldsymbol{u}_\theta}{\partial\boldsymbol{x}}(t, \boldsymbol{x}_t), \\ \boldsymbol{a}_\theta(1) &= \boldsymbol{0}, & \frac{\mathrm{d}\boldsymbol{a}_\theta}{\mathrm{d}t}(t) &= -\boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\boldsymbol{u}_\theta}{\partial\theta}(t, \boldsymbol{x}_t). \end{aligned} \tag{D.1}$$

Now we present the proof.

Proof. The adjoint state can be simplified by rewriting the vector field in terms of the denoiser to find

$$\frac{\mathrm{d}\boldsymbol{a}_{\boldsymbol{x}}}{\mathrm{d}t}(t) = -a_t\boldsymbol{a}_{\boldsymbol{x}}(t) - b_t\boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t)}{\partial\boldsymbol{x}_t}. \tag{D.2}$$

We can express this *backwards-in-time* ODE as an integral equation in the form of

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{x}}(s) &= \boldsymbol{a}_{\boldsymbol{x}}(t) - \int_t^s a_\tau\boldsymbol{a}_{\boldsymbol{x}}(t) + b_\tau\boldsymbol{a}_{\boldsymbol{x}}(\tau)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)}{\boldsymbol{x}_\tau}\,\mathrm{d}\tau, \\ &= \boldsymbol{a}_{\boldsymbol{x}}(t) + \int_s^t a_\tau\boldsymbol{a}_{\boldsymbol{x}}(t) + b_\tau\boldsymbol{a}_{\boldsymbol{x}}(\tau)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)}{\partial\boldsymbol{x}_\tau}\,\mathrm{d}\tau. \qquad \text{(time-reversal)} \end{aligned} \tag{D.3}$$

Using the technique of exponential integrators we rewrite the integral as

$$\begin{aligned} \boldsymbol{a}_{\boldsymbol{x}}(s) &= e^{\int_s^t a_u\,\mathrm{d}u}\boldsymbol{a}_{\boldsymbol{x}}(t) + \int_s^t e^{\int_\tau^t a_u\,\mathrm{d}u} b_\tau\boldsymbol{a}_{\boldsymbol{x}}(\tau)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)}{\partial\boldsymbol{x}_\tau}\,\mathrm{d}\tau, \\ &= \frac{\sigma_t}{\sigma_s}\boldsymbol{a}_{\boldsymbol{x}}(t) + \sigma_t\int_s^t \frac{b_\tau}{\sigma_\tau}\boldsymbol{a}_{\boldsymbol{x}}(\tau)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|\tau}(\boldsymbol{x}_\tau)}{\partial\boldsymbol{x}_\tau}\,\mathrm{d}\tau, \\ &= \frac{\sigma_t}{\sigma_s}\boldsymbol{a}_{\boldsymbol{x}}(t) + \sigma_t\int_{\gamma_s}^{\gamma_t} \boldsymbol{a}_{\boldsymbol{x}}(\gamma)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|\gamma}(\boldsymbol{x}_\gamma)}{\partial\boldsymbol{x}_\gamma}\,\mathrm{d}\gamma. \end{aligned} \tag{D.4}$$

By Assumption C.3 it follows that the vector-Jacobian product has $(k+1)$-th total derivatives, allowing us to define a first-order Taylor expansion around γ_s:

$$\boldsymbol{a}_{\boldsymbol{x}}(s) = \frac{\sigma_t}{\sigma_s}\boldsymbol{a}_{\boldsymbol{x}}(t) + (\alpha_t - \frac{\sigma_t}{\sigma_s}\alpha_s)\boldsymbol{a}_{\boldsymbol{x}}(s)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|s}(\boldsymbol{x}_s)}{\partial\boldsymbol{x}_s} + \mathcal{O}(h^2). \tag{D.5}$$

Thus the first-order approximation of the adjoint state at time t with a stepsize of $h = \gamma_1 - \gamma_t$ is the implicit equation

$$\boldsymbol{a}_{\boldsymbol{x}}(t) = \boldsymbol{a}_{\boldsymbol{x}}(t)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t)}{\partial\boldsymbol{x}_t}. \tag{D.6}$$

Now to solve the implicit equation we can use the fixed-point iteration method. Let $\boldsymbol{a}_{\boldsymbol{x}}(t)^{(0)} = \boldsymbol{a}_{\boldsymbol{x}}(1)$, then the first iteration has

$$\boldsymbol{a}_{\boldsymbol{x}}(t)^{(1)} = \boldsymbol{a}_{\boldsymbol{x}}(1)^\top \frac{\partial\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t)}{\partial\boldsymbol{x}_t} = \nabla_{\boldsymbol{x}_t}\mathcal{L}(\hat{\boldsymbol{x}}_{1|t}(\boldsymbol{x}_t)). \tag{D.7}$$

Thus we have shown that the greedy gradients are equivalent to the first iteration of an implicit Euler discretization of the continuous adjoint equations.

□

E PROOF OF THEOREM 5.4

We restate Theorem 5.4 here:

Theorem E.1 (Continuous adjoint equations for the control term)**.** *Let $\boldsymbol{u}_\theta \in \mathcal{C}^{1,1}([0,1] \times \mathbb{R}^d; \mathbb{R}^d)$ be a parameterization of some time-dependent vector field of a neural ODE that is Lipschitz continuous in its second argument, and let $\boldsymbol{z} \in \mathcal{C}^1([0,1]; \mathbb{R}^d)$ be an additional control signal such that the new dynamics are given by*

$$\frac{\mathrm{d}\boldsymbol{x}_t}{\mathrm{d}t} = \boldsymbol{u}_\theta(t, \boldsymbol{x}_t) + \boldsymbol{z}(t). \tag{E.1}$$

Let $\boldsymbol{a}_{\boldsymbol{z}}(t) \coloneqq \partial\mathcal{L}/\partial\boldsymbol{z}(t)$ then

$$\boldsymbol{a}_{\boldsymbol{z}}(t) = -\int_1^t \boldsymbol{a}_{\boldsymbol{x}}(s)\,\mathrm{d}s. \tag{E.2}$$

Our proof follows the structure of the then modern proof of Pontryagin's original result (Pontryagin et al., 1963) presented by (Chen et al., 2018).

Proof. For notational clarity, we use the notation $\boldsymbol{x}(t) = \boldsymbol{x}_t$. We define the augmented state on $[0, T]$ as

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{z}\end{bmatrix}(t) = \boldsymbol{f}_{\text{aug}} = \begin{bmatrix}\boldsymbol{u}_\theta(t, \boldsymbol{x}(t)) + \boldsymbol{z}(t)\\ \frac{\mathrm{d}\boldsymbol{z}}{\mathrm{d}t}(t)\end{bmatrix}, \tag{E.3}$$

and the augmented adjoint state as

$$\boldsymbol{a}_{\text{aug}}(t) \coloneqq \begin{bmatrix}\boldsymbol{a}_{\boldsymbol{x}}\\ \boldsymbol{a}_{\boldsymbol{z}}\end{bmatrix}(t). \tag{E.4}$$

The Jacobian of $\boldsymbol{f}_{\text{aug}}$ has form

$$\frac{\partial \boldsymbol{f}_{\text{aug}}}{\partial[\boldsymbol{x}, \boldsymbol{z}]} = \begin{bmatrix}\frac{\partial \boldsymbol{u}_\theta(t, \boldsymbol{x}(t))}{\partial \boldsymbol{x}} & \mathbf{1}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}. \tag{E.5}$$

The evolution of the adjoint state is given by

$$\frac{\mathrm{d}\boldsymbol{a}_{\text{aug}}}{\mathrm{d}t}(t) = -\begin{bmatrix}\boldsymbol{a}_{\boldsymbol{x}} & \boldsymbol{a}_{\boldsymbol{z}}\end{bmatrix}(t)\frac{\partial \boldsymbol{f}_{\text{aug}}}{\partial[\boldsymbol{x}, \boldsymbol{z}]}(t). \tag{E.6}$$

Therefore, $\boldsymbol{a}_{\boldsymbol{u}}(t)$ evolves with

$$\boldsymbol{a}_{\boldsymbol{u}}(T) = \mathbf{0}, \qquad \frac{\mathrm{d}\boldsymbol{a}_{\boldsymbol{u}}}{\mathrm{d}t}(t) = -\boldsymbol{a}_{\boldsymbol{x}}(t), \tag{E.7}$$

thereby finishing the proof. □