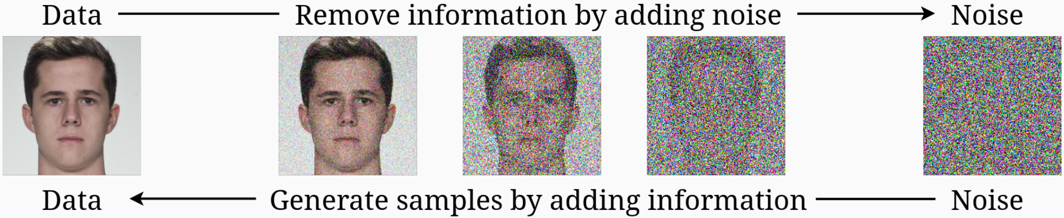
AdjointDEIS: Efficient Gradients for Diffusion Models

Zander W. Blasingame Chen Liu

Clarkson University
Potsdam, NY, USA

NeurIPS 2024

Diffusion Models



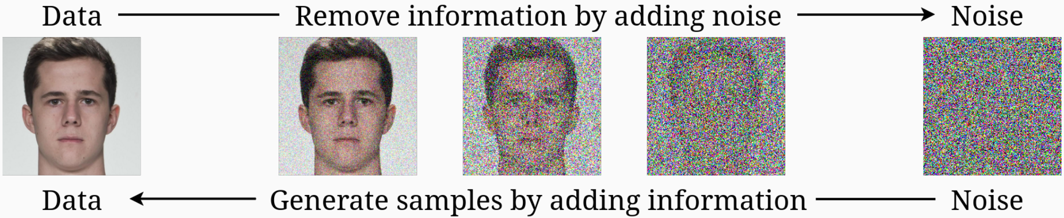
- Forward diffusion process is governed by the Itô SDE

$$\mathrm{d}\mathbf{x}_t = f(t)\mathbf{x}_t \,\mathrm{d}t + g(t)\,\mathrm{d}\mathbf{w}_t, \tag{1}$$

where $\{\mathbf{w}_t\}_{t\in[0,T]}$ is the standard Wiener process on $[0, T]$.

[1] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

Diffusion Models



- The diffusion equation can be reversed with

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

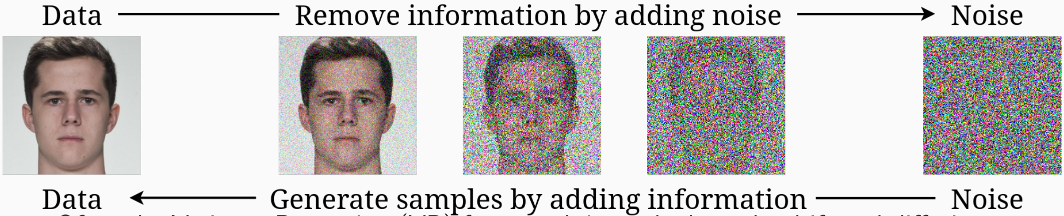
where $\bar{\mathbf{w}}_t$ is the *reverse* Wiener process and '$\mathrm{d}t$' is a *negative* timestep.

- The marginal distributions $p_t(\mathbf{x})$ follow the *probability flow* ODE[1]

$$\frac{\mathrm{d}\mathbf{x}_t}{\mathrm{d}t} = f(t)\mathbf{x}_t - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t). \tag{3}$$

[1] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

Diffusion Models



- Often the Variance Preserving (VP) framework is used where the drift and diffusion coefficients are

$$f(t) = \frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}, \qquad g^2(t) = \frac{\mathrm{d}\sigma_t^2}{\mathrm{d}t} - 2\frac{\mathrm{d}\log \alpha_t}{\mathrm{d}t}\sigma_t^2, \tag{4}$$

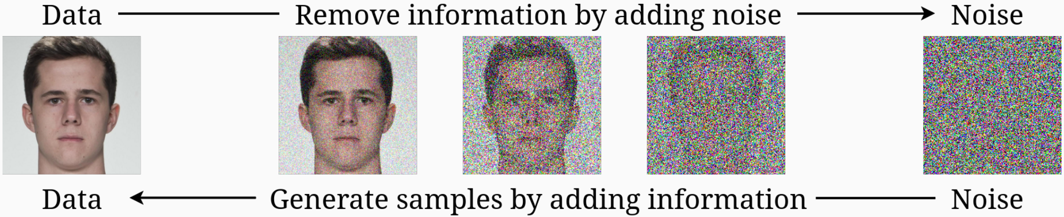
  for some noise schedule α_t, σ_t

- Sampling the forward trajectory then simplifies to

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\epsilon}_t \qquad \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

[1] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

Diffusion Models



- Train the model via score-matching to learn $\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)$.
- This is similar to learning the noise ϵ, *i.e.*,

$$\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \approx -\sigma_t \nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t). \tag{6}$$

[1] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=PxTIG12RRHS`.

Problem Statement

- Solve the following optimization problem:

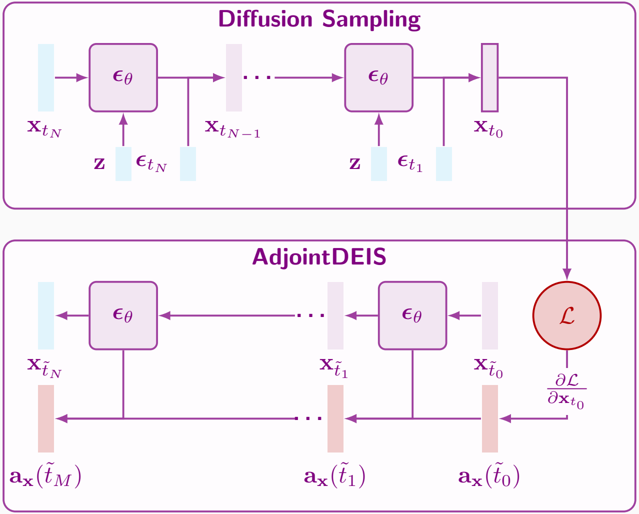
$$\operatorname*{arg\,min}_{\mathbf{x}_T,\mathbf{z},\theta} \mathcal{L}\left(\mathbf{x}_T + \int_T^0 f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,\mathbf{z},t)\;\mathrm{d}t\right). \tag{7}$$

- Or in the SDE case:

$$\operatorname*{arg\,min}_{\mathbf{x}_T,\mathbf{z},\theta} \mathcal{L}\left(\mathbf{x}_T + \int_T^0 f(t)\mathbf{x}_t + \frac{g^2(t)}{\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t,\mathbf{z},t)\;\mathrm{d}t + \int_T^0 g(t)\;\mathrm{d}\bar{\mathbf{w}}_t\right). \tag{8}$$

- To backpropagate through an ODE/SDE solve we solve the continuous adjoint equations.

AdjointDEIS



Continuous Adjoint Equations

- Let $\boldsymbol{f}_\theta$ describe a parameterized neural field of the probability flow ODE, defined as

$$\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t) = f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t). \tag{9}$$

Continuous Adjoint Equations

- Let $\boldsymbol{f}_\theta$ describe a parameterized neural field of the probability flow ODE, defined as

$$\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t) = f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t). \tag{9}$$

- Then $\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)$ describes a neural ODE which admits an adjoint state, $\mathbf{a}_\mathbf{x} := \partial\mathcal{L}/\partial\mathbf{x}_t$ (and likewise for $\mathbf{a}_\mathbf{z}(t)$ and $\mathbf{a}_\theta(t)$), which solve the continuous adjoint equations [6, Theorem 5.2] in the form of the following Initial Value Problem (IVP):

$$\begin{aligned}
\mathbf{a}_\mathbf{x}(0) &= \frac{\partial\mathcal{L}}{\partial\mathbf{x}_0}, & \frac{\mathrm{d}\mathbf{a}_\mathbf{x}}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\mathbf{x}_t},\\
\mathbf{a}_\mathbf{z}(0) &= \mathbf{0}, & \frac{\mathrm{d}\mathbf{a}_\mathbf{z}}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\mathbf{z}},\\
\mathbf{a}_\theta(0) &= \mathbf{0}, & \frac{\mathrm{d}\mathbf{a}_\theta}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\theta}.
\end{aligned} \tag{10}$$

Continuous Adjoint Equations

- Let $\boldsymbol{f}_\theta$ describe a parameterized neural field of the probability flow ODE, defined as

$$\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t) = \underbrace{f(t)\mathbf{x}_t + \frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t).}_{\text{Black box } \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t) \text{ looses known information of } f(t) \text{ and } g(t).} \tag{9}$$

- Then $\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)$ describes a neural ODE which admits an adjoint state, $\mathbf{a}_\mathbf{x} := \partial\mathcal{L}/\partial\mathbf{x}_t$ (and likewise for $\mathbf{a}_\mathbf{z}(t)$ and $\mathbf{a}_\theta(t)$), which solve the continuous adjoint equations [6, Theorem 5.2] in the form of the following Initial Value Problem (IVP):

$$\begin{aligned}
\mathbf{a}_\mathbf{x}(0) &= \frac{\partial\mathcal{L}}{\partial\mathbf{x}_0}, & \frac{\mathrm{d}\mathbf{a}_\mathbf{x}}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\mathbf{x}_t}, \\
\mathbf{a}_\mathbf{z}(0) &= \mathbf{0}, & \frac{\mathrm{d}\mathbf{a}_\mathbf{z}}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\mathbf{z}}, \\
\mathbf{a}_\theta(0) &= \mathbf{0}, & \frac{\mathrm{d}\mathbf{a}_\theta}{\mathrm{d}t}(t) &= -\mathbf{a}_\mathbf{x}(t)^\top \frac{\partial\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial\theta}.
\end{aligned} \tag{10}$$

The Continuous Adjoint Equations are also Semi-linear

- Like diffusion ODEs the adjoint diffusion ODE is also semi-linear

$$\frac{\mathrm{d}\mathbf{a}_{\mathbf{x}}}{\mathrm{d}t}(t) = -\underbrace{f(t)\mathbf{a}_{\mathbf{x}}(t)}_{\text{Linear}} - \frac{g^2(t)}{2\sigma_t}\mathbf{a}_{\mathbf{x}}(t)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial \mathbf{x}_t}. \tag{11}$$

[2] Cheng Lu et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 5775–5787. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf.

The Continuous Adjoint Equations are also Semi-linear

- Like diffusion ODEs the adjoint diffusion ODE is also semi-linear

$$\frac{\mathrm{d}\mathbf{a}_{\mathbf{x}}}{\mathrm{d}t}(t) = -f(t)\mathbf{a}_{\mathbf{x}}(t) - \frac{g^2(t)}{2\sigma_t}\mathbf{a}_{\mathbf{x}}(t)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial \mathbf{x}_t}. \tag{11}$$

- Then, the exact solution at time s given time $t < s$ is found to be

$$\mathbf{a}_{\mathbf{x}}(s) = \underbrace{e^{\int_s^t f(\tau)\,\mathrm{d}\tau}\mathbf{a}_{\mathbf{x}}(t)}_{\text{linear}} - \underbrace{\int_t^s e^{\int_s^u f(\tau)\,\mathrm{d}\tau}\frac{g^2(u)}{2\sigma_u}\mathbf{a}_{\mathbf{x}}(u)^\top \frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_u, \mathbf{z}, u)}{\partial \mathbf{x}_u}\,\mathrm{d}u}_{\text{non-linear}}. \tag{12}$$

[2] Cheng Lu et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 5775–5787. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf.

The Continuous Adjoint Equations are also Semi-linear

- Like diffusion ODEs the adjoint diffusion ODE is also semi-linear

$$\frac{\mathrm{d}\mathbf{a}_{\mathbf{x}}}{\mathrm{d}t}(t) = -f(t)\mathbf{a}_{\mathbf{x}}(t) - \frac{g^2(t)}{2\sigma_t}\mathbf{a}_{\mathbf{x}}(t)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)}{\partial \mathbf{x}_t}. \tag{11}$$

- Then, the exact solution at time s given time $t < s$ is found to be

$$\mathbf{a}_{\mathbf{x}}(s) = \underbrace{e^{\int_s^t f(\tau)\,\mathrm{d}\tau}\mathbf{a}_{\mathbf{x}}(t)}_{\text{linear}} \underbrace{- \int_t^s e^{\int_s^u f(\tau)\,\mathrm{d}\tau} \frac{g^2(u)}{2\sigma_u}\mathbf{a}_{\mathbf{x}}(u)^\top \frac{\boldsymbol{\epsilon}_\theta(\mathbf{x}_u, \mathbf{z}, u)}{\partial \mathbf{x}_u}\,\mathrm{d}u}_{\text{non-linear}}. \tag{12}$$

- Use the log-SNR trick[2] to further simplify the exact solution with $\lambda_t \coloneqq \log(\alpha_t/\sigma_t)$.

[2] Cheng Lu et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 5775–5787. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf.

Simplified Exact Solutions

Proposition 1 (Exact solution of adjoint diffusion ODEs)
Given initial values $[\mathbf{a}_{\mathbf{x}}(t), \mathbf{a}_{\mathbf{z}}(t), \mathbf{a}_{\theta}(t)]$ *at time* $t \in (0, T)$, *the solution* $[\mathbf{a}_{\mathbf{x}}(s), \mathbf{a}_{\mathbf{z}}(s), \mathbf{a}_{\theta}(s)]$ *at time* $s \in (t, T]$ *of adjoint diffusion ODEs in Eq.* (10) *is*

$$\mathbf{a}_{\mathbf{x}}(s) = \frac{\alpha_t}{\alpha_s}\mathbf{a}_{\mathbf{x}}(t) + \frac{1}{\alpha_s}\int_{\lambda_t}^{\lambda_s} \alpha_\lambda^2 e^{-\lambda} \mathbf{a}_{\mathbf{x}}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{x}_\lambda} \, \mathrm{d}\lambda, \tag{13}$$

$$\mathbf{a}_{\mathbf{z}}(s) = \mathbf{a}_{\mathbf{z}}(t) + \int_{\lambda_t}^{\lambda_s} \alpha_\lambda e^{-\lambda} \mathbf{a}_{\mathbf{x}}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{z}} \, \mathrm{d}\lambda, \tag{14}$$

$$\mathbf{a}_{\theta}(s) = \mathbf{a}_{\theta}(t) + \int_{\lambda_t}^{\lambda_s} \alpha_\lambda e^{-\lambda} \mathbf{a}_{\mathbf{x}}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \theta} \, \mathrm{d}\lambda. \tag{15}$$

Designing Bespoke ODE Solvers

- We denote the n-th derivative of the *scaled* vector-Jacobian product by

$$\mathbf{V}^{(n)}(\mathbf{x}; \lambda_t) = \frac{\mathrm{d}^n}{\mathrm{d}\lambda^n} \left[\alpha_\lambda^2 \mathbf{a}_\mathbf{x}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{x}_\lambda} \right]_{\lambda=\lambda_t}. \tag{16}$$

- Use Taylor Expansion on Eq. (13) to obtain and letting $h = \lambda_s - \lambda_t$ yields

$$\mathbf{a}_\mathbf{x}(s) = \underbrace{\frac{\alpha_t}{\alpha_s} \mathbf{a}_\mathbf{x}(t)}_{\substack{\text{Linear term} \\ \textbf{Exactly computed}}} + \frac{1}{\alpha_s} \sum_{n=0}^{k-1} \mathbf{V}^{(n)}(\mathbf{x}; \lambda_t) \int_{\lambda_t}^{\lambda_s} \frac{(\lambda - \lambda_t)^n}{n!} e^{-\lambda} \, \mathrm{d}\lambda + \mathcal{O}(h^{k+1}). \tag{17}$$

Designing Bespoke ODE Solvers

- We denote the n-th derivative of the *scaled* vector-Jacobian product by

$$\mathbf{V}^{(n)}(\mathbf{x};\lambda_t) = \frac{\mathrm{d}^n}{\mathrm{d}\lambda^n}\left[\alpha_\lambda^2 \mathbf{a}_\mathbf{x}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{x}_\lambda}\right]_{\lambda=\lambda_t}. \tag{16}$$

- Use Taylor Expansion on Eq. (13) to obtain and letting $h = \lambda_s - \lambda_t$ yields

$$\mathbf{a}_\mathbf{x}(s) = \underbrace{\frac{\alpha_t}{\alpha_s}\mathbf{a}_\mathbf{x}(t)}_{\substack{\text{Linear term}\\ \textbf{Exactly computed}}} + \frac{1}{\alpha_s}\sum_{n=0}^{k-1} \underbrace{\mathbf{V}^{(n)}(\mathbf{x};\lambda_t)}_{\substack{\text{Derivatives}\\ \textbf{Approximated}}} \int_{\lambda_t}^{\lambda_s} \frac{(\lambda-\lambda_t)^n}{n!} e^{-\lambda}\,\mathrm{d}\lambda + \mathcal{O}(h^{k+1}). \tag{17}$$

Designing Bespoke ODE Solvers

- We denote the n-th derivative of the *scaled* vector-Jacobian product by

$$\mathbf{V}^{(n)}(\mathbf{x};\lambda_t) = \frac{\mathrm{d}^n}{\mathrm{d}\lambda^n}\left[\alpha_\lambda^2 \mathbf{a}_{\mathbf{x}}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{x}_\lambda}\right]_{\lambda=\lambda_t}. \tag{16}$$

- Use Taylor Expansion on Eq. (13) to obtain and letting $h = \lambda_s - \lambda_t$ yields

$$\mathbf{a}_{\mathbf{x}}(s) = \underbrace{\frac{\alpha_t}{\alpha_s}\mathbf{a}_{\mathbf{x}}(t)}_{\substack{\text{Linear term}\\ \textbf{Exactly computed}}} + \frac{1}{\alpha_s}\sum_{n=0}^{k-1} \underbrace{\mathbf{V}^{(n)}(\mathbf{x};\lambda_t)}_{\substack{\text{Derivatives}\\ \textbf{Approximated}}} \underbrace{\int_{\lambda_t}^{\lambda_s} \frac{(\lambda-\lambda_t)^n}{n!} e^{-\lambda}\,\mathrm{d}\lambda}_{\substack{\text{Coefficients}\\ \textbf{Analytically computed}}} + \mathcal{O}(h^{k+1}). \tag{17}$$

Designing Bespoke ODE Solvers

- We denote the n-th derivative of the *scaled* vector-Jacobian product by

$$\mathbf{V}^{(n)}(\mathbf{x};\lambda_t) = \frac{\mathrm{d}^n}{\mathrm{d}\lambda^n}\left[\alpha_\lambda^2 \mathbf{a}_{\mathbf{x}}(\lambda)^\top \frac{\partial \boldsymbol{\epsilon}_\theta(\mathbf{x}_\lambda, \mathbf{z}, \lambda)}{\partial \mathbf{x}_\lambda}\right]_{\lambda=\lambda_t}. \tag{16}$$

- Use Taylor Expansion on Eq. (13) to obtain and letting $h = \lambda_s - \lambda_t$ yields

$$\mathbf{a}_{\mathbf{x}}(s) = \underbrace{\frac{\alpha_t}{\alpha_s}\mathbf{a}_{\mathbf{x}}(t)}_{\substack{\text{Linear term}\\ \textbf{Exactly computed}}} + \frac{1}{\alpha_s}\sum_{n=0}^{k-1} \underbrace{\mathbf{V}^{(n)}(\mathbf{x};\lambda_t)}_{\substack{\text{Derivatives}\\ \textbf{Approximated}}} \underbrace{\int_{\lambda_t}^{\lambda_s} \frac{(\lambda-\lambda_t)^n}{n!} e^{-\lambda}\,\mathrm{d}\lambda}_{\substack{\text{Coefficients}\\ \textbf{Analytically computed}}} + \underbrace{\mathcal{O}(h^{k+1})}_{\substack{\text{High-order errors}\\ \textbf{Omitted}}}. \tag{17}$$

- And analogously for $\mathbf{a}_{\mathbf{z}}(t)$ and $\mathbf{a}_\theta(t)$.

Certain Adjoint SDEs are Actually ODEs

Theorem 1
Let $\boldsymbol{f} : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ *be in* $\mathcal{C}_b^{\infty,1}$ *and* $\boldsymbol{g} : \mathbb{R} \to \mathbb{R}^{d \times w}$ *be in* $\mathcal{C}_b^1$*. Let* $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ *be a scalar-valued differentiable function. Let* $\mathbf{w}_t : [0, T] \to \mathbb{R}^w$ *be a* w*-dimensional Wiener process. Let* $\mathbf{x} : [0, T] \to \mathbb{R}^d$ *solve the Stratonovich SDE*

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

with initial condition $\mathbf{x}_0$*. Then the adjoint process* $\mathbf{a}_{\mathbf{x}}(t) := \partial \mathcal{L}(\mathbf{x}_T) / \partial \mathbf{x}_t$ *is a strong solution to the backwards-in-time ODE*

$$\mathrm{d}\mathbf{a}_{\mathbf{x}}(t) = -\mathbf{a}_{\mathbf{x}}(t)^\top \frac{\partial \boldsymbol{f}}{\partial \mathbf{x}_t}(\mathbf{x}_t, t)\, \mathrm{d}t. \tag{18}$$

ODE Solvers for the Adjoint Diffusion SDE

- The Probability Flow ODEs are related to the diffusion SDEs by the manipulations of the Kolmogorov equations[3].

[3] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

ODE Solvers for the Adjoint Diffusion SDE

- The Probability Flow ODEs are related to the diffusion SDEs by the manipulations of the Kolmogorov equations[3].
- The drift term is identical to the vector field of the ODE, sans a factor of two:

$$\underbrace{\mathrm{d}\mathbf{x}_t = f(t)\mathbf{x}_t + 2\frac{g^2(t)}{2\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)\ \mathrm{d}t}_{\text{Probabilty Flow ODE}} + g(t)\ \mathrm{d}\bar{\mathbf{w}}_t. \tag{19}$$

[3] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

ODE Solvers for the Adjoint Diffusion SDE

- The Probability Flow ODEs are related to the diffusion SDEs by the manipulations of the Kolmogorov equations[3].
- The drift term is identical to the vector field of the ODE, sans a factor of two:

$$\mathrm{d}\mathbf{x}_t = \underbrace{f(t)\mathbf{x}_t + \frac{g^2(t)}{\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)}_{=\boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)} \,\mathrm{d}t + g(t)\,\mathrm{d}\bar{\mathbf{w}}_t. \tag{19}$$

- By Theorem 1 the adjoint SDE evolves with an ODE with vector field $-\mathbf{a}_\mathbf{x}(t)^\top \partial \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)/\partial \mathbf{x}_t$.

[3] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=PxTIG12RRHS`.

ODE Solvers for the Adjoint Diffusion SDE

- The Probability Flow ODEs are related to the diffusion SDEs by the manipulations of the Kolmogorov equations[3].
- The drift term is identical to the vector field of the ODE, sans a factor of two:

$$\mathrm{d}\mathbf{x}_t = \underbrace{f(t)\mathbf{x}_t + \frac{g^2(t)}{\sigma_t}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, \mathbf{z}, t)}_{=\boldsymbol{f}_\theta(\mathbf{x}_t,\mathbf{z},t)} \ \mathrm{d}t + g(t)\ \mathrm{d}\bar{\mathbf{w}}_t. \tag{19}$$

- By Theorem 1 the adjoint SDE evolves with an ODE with vector field $-\mathbf{a}_{\mathbf{x}}(t)^\top \partial \boldsymbol{f}_\theta(\mathbf{x}_t, \mathbf{z}, t)/\partial \mathbf{x}_t$.
- Therefore, we can use the *same* bespoke ODE solvers for adjoint diffusion ODEs with the added factor of 2!

[3] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: `https://openreview.net/forum?id=PxTIG12RRHS`.

Experiment - Face Morphing

(a) Identity a **(b)** Face morphing with AdjointDEIS **(c)** Identity b

Figure 1: Create a morphed face which causes a Face Recognition (FR) system to accept it with **both** identities.

Experiment - Face Morphing

- Goal is to adversarially attack an FR system by finding the $\mathbf{x}_T, \mathbf{z}$ which creates the optimal morph.

Experiment - Face Morphing

- Goal is to adversarially attack an FR system by finding the $\mathbf{x}_T, \mathbf{z}$ which creates the optimal morph.
- Optimality is defined with respect to an identity loss which is simply the average distance between the embeddings in the FR space.

Experiment - Face Morphing

- Goal is to adversarially attack an FR system by finding the $\mathbf{x}_T, \mathbf{z}$ which creates the optimal morph.
- Optimality is defined with respect to an identity loss which is simply the average distance between the embeddings in the FR space.
- Use AdjointDEIS massively improves the performance of Diffusion Morphs (DiM).

Table 1: Vulnerability of different FR systems across different morphing attacks on the SYN-MAD 2022 dataset. FMR = 0.1%.

		MMPMR [9](↑)		
Morphing Attack	**NFE**(↓)	**AdaFace** [7]	**ArcFace** [4]	**ElasticFace** [3]
Webmorph [5]	-	97.96	96.93	98.36
MIPGAN-I [11]	-	72.19	77.51	66.46
MIPGAN-II [11]	-	70.55	72.19	65.24
DiM-A [2]	350	92.23	90.18	93.05
Fast-DiM [1]	300	92.02	90.18	93.05
Morph-PIPE [12]	2350	95.91	92.84	95.5
DiM + AdjointDEIS-1 (ODE)	2250	**99.8**	**98.77**	**99.39**
DiM + AdjointDEIS-1 (SDE)	2250	98.57	97.96	97.75

Summary

- We propose a highly simplified formulation of the exact solution to the continuous adjoint equations for diffusion ODEs/SDEs.
- We propose a bespoke family of k-th order solvers for diffusion ODEs/SDEs to obtain gradients efficiently.
- We show that the adjoint SDE evolves with a much simpler ODE.

(a) Paper

(b) Code

References

[1] Zander W. Blasingame and Chen Liu. "Fast-DiM: Towards Fast Diffusion Morphs". In: *IEEE Security & Privacy* 22.4 (June 2024), pp. 103–114. DOI: 10.1109/MSEC.2024.3410112.

[2] Zander W. Blasingame and Chen Liu. "Leveraging Diffusion for Strong and High Quality Face Morphing Attacks". In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 6.1 (2024), pp. 118–131. DOI: 10.1109/TBIOM.2024.3349857.

[3] Fadi Boutros et al. "ElasticFace: Elastic Margin Loss for Deep Face Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2022, pp. 1578–1587.

[4] Jiankang Deng et al. "Arcface: Additive angular margin loss for deep face recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4690–4699.

[5] Marco Huber et al. "SYN-MAD 2022: Competition on Face Morphing Attack Detection Based on Privacy-aware Synthetic Training Data". In: *2022 IEEE International Joint Conference on Biometrics (IJCB)*. 2022, pp. 1–10. DOI: 10.1109/IJCB54206.2022.10007950.

[6] Patrick Kidger. "On Neural Differential Equations". PhD thesis. Oxford University, 2022.

[7] Minchul Kim, Anil K Jain, and Xiaoming Liu. "AdaFace: Quality Adaptive Margin for Face Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

[8] Cheng Lu et al. "DPM-Solver: A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 5775–5787. URL: https://proceedings.neurips.cc/paper_files/paper/2022/file/260a14acce2a89dad36adc8eefe7c59e-Paper-Conference.pdf.

[9] Ulrich Scherhag et al. "Biometric Systems under Morphing Attacks: Assessment of Morphing Techniques and Vulnerability Reporting". In: *2017 International Conference of the Biometrics Special Interest Group (BIOSIG)*. 2017, pp. 1–7. DOI: 10.23919/BIOSIG.2017.8053499.

[10] Yang Song et al. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=PxTIG12RRHS.

[11] Haoyu Zhang et al. "MIPGAN—Generating Strong and High Quality Morphing Attacks Using Identity Prior Driven GAN". In: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 3.3 (2021), pp. 365–383. DOI: 10.1109/TBIOM.2021.3072349.

References iv

[12] Haoyu Zhang et al. “Morph-PIPE: Plugging in Identity Prior to Enhance Face Morphing Attack Based on Diffusion Model”. In: *Norwegian Information Security Conference (NISK)*. 2023.