Dynamics and Memorization Behaviour of Score-Based Diffusion Models

Ricardo Baptista

Computing and Mathematical Sciences

Caltech

IMSI: Statistical and Computational Challenges
in Probabilistic Scientific Machine Learning

June 9, 2025

Collaborators

Memorization and Regularization in Generative Diffusion Models

Ricardo Baptista[1], Agnimitras Dasgupta[2], Nikola Kovachki[3],
Assad Oberai[2], Andrew Stuart[1]

[1]Computing+Mathematical Sciences
California Institute of Technology

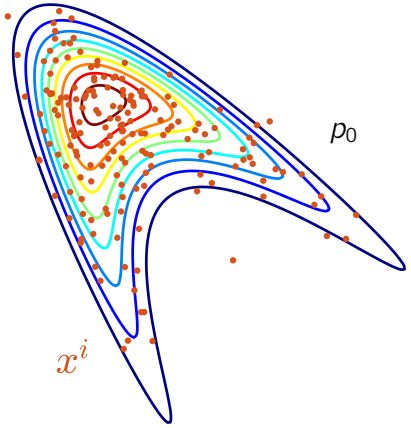
[2]Aerospace and Mechanical Engineering
University of Southern California

[3]NVIDIA Corporation

arXiv:2501.15785

Task of generative modeling

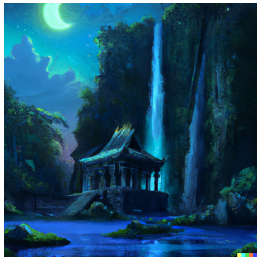
Setting: Collect i.i.d. samples $\{\mathbf{x}_0^i\}_{i=1}^N$ (e.g., images, text) from probability distribution p_0



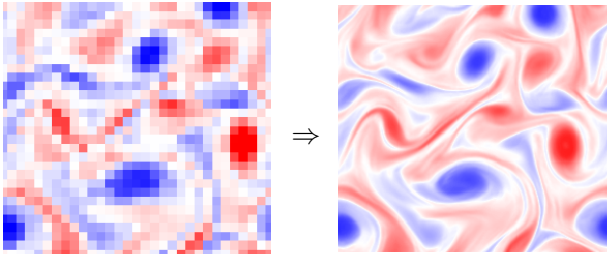
Goal: Generate *new* samples from p_0 that are *not present in the training dataset*

Diffusion Models Generate High-Quality Images

Machine learning: Prompt-to-image models (Ramesh et al., 2022)

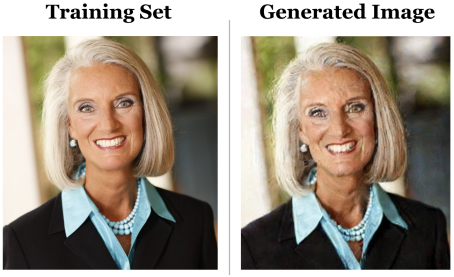


Scientific computing: Super-resolution inverse problems (Wan et al., 2023)



But Diffusion Models Can Lack Diversity

Memorizing training data (Carlini et al., 2023)



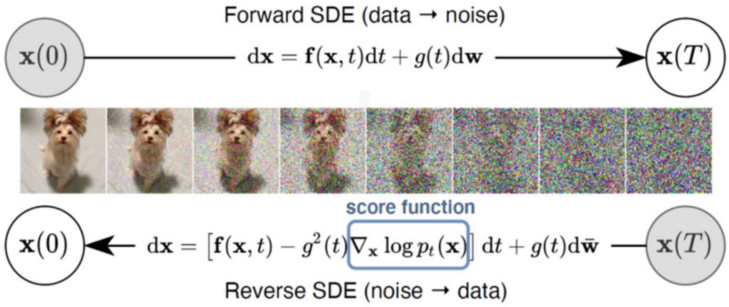
Memorizing subsets of images (Somepalli et al., 2023)

Table of Contents

Table of Contents

Generative Modeling by Learning Score Functions

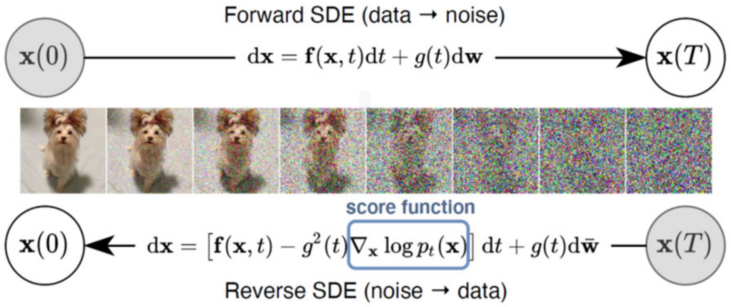
- Forward process adds noise to map data to noise at $t = T$
- Reverse process converts Gaussian noise to data at $t = 0$



[Song et al., 2021]

Generative Modeling by Learning Score Functions

- Forward process adds noise to map data to noise at $t = T$
- Reverse process converts Gaussian noise to data at $t = 0$



[Song et al., 2021]

Key Ideas:

- Diffusions rely on the *score* $\nabla_{\mathbf{x}} \log p(\mathbf{x}, t)$ of the forward process for each t
- In practice, the data distribution is prescribed by samples, i.e., $p_0 = \frac{1}{N} \sum_{i=1}^{N} \delta_{\mathbf{x}_0^i}$

Learning Score Functions From Data

Goal: Learn the score of $p(\mathbf{x}, t) = \int p(\mathbf{x}, t|\mathbf{x}_0) dp_0(\mathbf{x}_0)$ for each t

Learning Score Functions From Data

Goal: Learn the score of $p(\mathbf{x}, t) = \int p(\mathbf{x}, t|\mathbf{x}_0) dp_0(\mathbf{x}_0)$ for each t

Approach: Denoising score-matching (Vincent, 2011)

$$\arg\min_s \int_0^T \mathbb{E}_{\mathbf{x}} |s(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t)|^2 dt$$

$$= \arg\min_s \int_0^T \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{x}_0)} |s(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t|\mathbf{x}_0)|^2 dt}_{\text{Does not explicitly depend on the data density}} + C$$

Learning Score Functions From Data

Goal: Learn the score of $p(\mathbf{x}, t) = \int p(\mathbf{x}, t|\mathbf{x}_0) dp_0(\mathbf{x}_0)$ for each t

Approach: Denoising score-matching (Vincent, 2011)

$$\arg\min_s \int_0^T \mathbb{E}_{\mathbf{x}} |s(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t)|^2 dt$$

$$= \arg\min_s \int_0^T \underbrace{\mathbb{E}_{(\mathbf{x},\mathbf{x}_0)} |s(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t|\mathbf{x}_0)|^2 dt}_{\text{Does not explicitly depend on the data density}} + C$$

Recipe for sampling:

1. Given data $\{\mathbf{x}_0^i\}_{i=1}^N \sim p_0$, learn score

$$s^* \in \arg\min_s \int_0^T \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathbf{x}} \left| s(\mathbf{x}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t|\mathbf{x}_0^i) \right|^2 dt$$

2. Simulate the reverse process to generate new data

Example for Score Learning

Variance exploding forward process: $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

- Conditional distribution: $p(\mathbf{x}, t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0, \sigma^2(t)I_d)$ for $\sigma^2(t) = \int_0^t g(s)ds$
- Score function: $\nabla \log p(\mathbf{x}, t|\mathbf{x}_0) = -\frac{\mathbf{x}-\mathbf{x}_0}{\sigma^2(t)}$

Example for Score Learning

Variance exploding forward process: $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

- Conditional distribution: $p(\mathbf{x}, t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0, \sigma^2(t)I_d)$ for $\sigma^2(t) = \int_0^t g(s)ds$
- Score function: $\nabla \log p(\mathbf{x}, t|\mathbf{x}_0) = -\frac{\mathbf{x}-\mathbf{x}_0}{\sigma^2(t)}$

Learning problem:

$$s^* \in \arg\min_s \int_0^T \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathbf{x}} \left| s(\mathbf{x}, t) + \frac{\mathbf{x} - \mathbf{x}_0^i}{\sigma^2(t)} \right|^2 dt$$

Example for Score Learning

Variance exploding forward process: $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

- Conditional distribution: $p(\mathbf{x}, t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}; \mathbf{x}_0, \sigma^2(t)I_d)$ for $\sigma^2(t) = \int_0^t g(s)ds$
- Score function: $\nabla \log p(\mathbf{x}, t|\mathbf{x}_0) = -\frac{\mathbf{x}-\mathbf{x}_0}{\sigma^2(t)}$

Learning problem:

$$s^* \in \arg\min_s \int_0^T \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\mathbf{x}} \left| s(\mathbf{x}, t) + \frac{\mathbf{x} - \mathbf{x}_0^i}{\sigma^2(t)} \right|^2 dt$$

Reverse process:

- Reverse-time SDE

$$d\mathbf{x} = -g(t)s^*(\mathbf{x}, t)dt + \sqrt{g(t)}d\mathbf{w}, \qquad \mathbf{x}(T) \sim \mathcal{N}(0, \sigma^2(T)I_d)$$

- Reverse-time ODE

$$\frac{d\mathbf{x}}{dt} = -\frac{g(t)}{2}s^*(\mathbf{x}, t), \qquad \mathbf{x}(T) \sim \mathcal{N}(0, \sigma^2(T)I_d)$$

Table of Contents

How Does Memorization Arise?

Consider variance exploding process $d\mathbf{x}_t = \sqrt{g(t)}d\mathbf{w}_t,\ \mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \sigma^2(t))$

Optimal empirical score (Gu et al., 2023; Scarvelis, Borde, and Solomon, 2023)

For $p_0 = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_0^i}$, the optimal score is

$$s^*(\mathbf{x}, t) = -\frac{1}{\sigma^2(t)} \sum_{i=1}^{N} (\mathbf{x} - \mathbf{x}_0^i) w_i(\mathbf{x}, t),$$

where $w_i(\mathbf{x}, t) \in [0, 1]$ are normalized Gaussian weights

$$w_i(\mathbf{x}, t) = \frac{\tilde{w}_i(\mathbf{x}, t)}{\sum_{l=1}^{N} \tilde{w}_\ell(\mathbf{x}, t)}, \qquad \tilde{w}_\ell(\mathbf{x}, t) = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_0^i|^2}{2\sigma^2(t)}\right)$$

How Does Memorization Arise?

Consider variance exploding process $d\mathbf{x}_t = \sqrt{g(t)}d\mathbf{w}_t,\ \mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \sigma^2(t))$

Optimal empirical score (Gu et al., 2023; Scarvelis, Borde, and Solomon, 2023)

For $p_0 = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{x}_0^i}$, the optimal score is

$$s^*(\mathbf{x}, t) = -\frac{1}{\sigma^2(t)} \sum_{i=1}^{N} (\mathbf{x} - \mathbf{x}_0^i) w_i(\mathbf{x}, t),$$

where $w_i(\mathbf{x}, t) \in [0, 1]$ are normalized Gaussian weights

$$w_i(\mathbf{x}, t) = \frac{\tilde{w}_i(\mathbf{x}, t)}{\sum_{l=1}^{N} \tilde{w}_\ell(\mathbf{x}, t)}, \qquad \tilde{w}_\ell(\mathbf{x}, t) = \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_0^i|^2}{2\sigma^2(t)}\right)$$

Takeaway: The optimal score contains all of the training samples

Limiting Behaviour of the Empirical Score

- For $\mathbf{x}$ near $\mathbf{x}_0^i$, weights collapse $w_i(\mathbf{x}, t) \to 1$, $w_\ell(\mathbf{x}, t) \to 0$ for $\ell \neq i$ and score is

$$s^*(\mathbf{x}, t) \to -\frac{\mathbf{x} - \mathbf{x}_0^i}{\sigma^2(t)}, \qquad t \to 0.$$

- Behavior depends on the Voronoi partitioning into cells of nearest data points

$$V(\mathbf{x}_0^i) \equiv \{\mathbf{x} \in \mathbb{R}^d \text{ s.t. } |\mathbf{x} - \mathbf{x}_0^i| < |\mathbf{x} - \mathbf{x}_0^\ell|, \ell \neq i\}$$

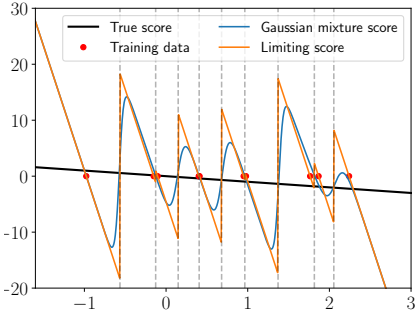
Limiting Behaviour of the Empirical Score

- For $\mathbf{x}$ near $\mathbf{x}_0^i$, weights collapse $w_i(\mathbf{x}, t) \to 1$, $w_\ell(\mathbf{x}, t) \to 0$ for $\ell \neq i$ and score is

$$s^*(\mathbf{x}, t) \to -\frac{\mathbf{x} - \mathbf{x}_0^i}{\sigma^2(t)}, \qquad t \to 0.$$

- Behavior depends on the Voronoi partitioning into cells of nearest data points

$$V(\mathbf{x}_0^i) \equiv \{\mathbf{x} \in \mathbb{R}^d \text{ s.t. } |\mathbf{x} - \mathbf{x}_0^i| < |\mathbf{x} - \mathbf{x}_0^\ell|, \ell \neq i\}$$



Rescaled score function and Voronoi partitioning for $\mathbf{x}_0^i \sim \mathcal{N}(0, 1)$

Reverse-Time Dynamical System

Recall: Variance exploding process $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

Today we will consider $g(t) = 2t,\ t \in [0, 1]$ but results generalize to other $g(t)$

Reverse-Time Dynamical System

Recall: Variance exploding process $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

Today we will consider $g(t) = 2t,\ t \in [0, 1]$ but results generalize to other $g(t)$

Reverse ODE:

- Integrate $\mathbf{x}$ backward-in-time starting from $\mathbf{x}(1) \sim \mathcal{N}(0, I)$

$$\frac{d\mathbf{x}}{dt} = -\frac{g(t)}{2}s^*(\mathbf{x}, t) = \frac{1}{t}(\mathbf{x} - \mathsf{x}_N(\mathbf{x}, t)), \qquad \mathsf{x}_N(\mathbf{x}, t) := \sum_{i=1}^{N} \mathbf{x}_0^i w_i(\mathbf{x}, t)$$

Reverse-Time Dynamical System

Recall: Variance exploding process $d\mathbf{x} = \sqrt{g(t)}d\mathbf{w}$

Today we will consider $g(t) = 2t,\ t \in [0, 1]$ but results generalize to other $g(t)$

Reverse ODE:

- Integrate $\mathbf{x}$ backward-in-time starting from $\mathbf{x}(1) \sim \mathcal{N}(0, I)$

$$\frac{d\mathbf{x}}{dt} = -\frac{g(t)}{2}s^*(\mathbf{x}, t) = \frac{1}{t}(\mathbf{x} - \mathrm{x}_N(\mathbf{x}, t)), \qquad \mathrm{x}_N(\mathbf{x}, t) \coloneqq \sum_{i=1}^{N} \mathbf{x}_0^i w_i(\mathbf{x}, t)$$

Change of variables: $s = -\log(t)$

- integrate $\mathbf{y}(s) = \mathbf{x}(e^{-s})$ forward-in-time starting from $\mathbf{y}(0) \sim \mathcal{N}(0, I)$

$$\frac{d\mathbf{y}}{ds} = -(\mathbf{y} - \mathrm{y}_N(\mathbf{y}, s)), \qquad \mathrm{y}_N(\mathbf{y}, s) \coloneqq \sum_{i=1}^{N} \mathbf{x}_0^i w_i(\mathbf{y}, e^{-s})$$

Limiting Behaviour

Limit Points: For any initial condition $\mathbf{y}(0)$, there are sequences $(s_k)_{k\in\mathbb{N}}$ so that

$$\lim_{s_k\to\infty} \mathbf{y}(s_k) = \mathbf{y}^*$$

Limit Points: For any initial condition $\mathbf{y}(0)$, there are sequences $(s_k)_{k\in\mathbb{N}}$ so that

$$\lim_{s_k\to\infty} \mathbf{y}(s_k) = \mathbf{y}^*$$

Main Theorem (Baptista et al., 2025)

The limit points $\mathbf{y}^*$ are attained at one the data points $\mathbf{x}_0^i$ or on the boundaries of the Voronoi tesselation $\{\partial V(\mathbf{x}_0^i)\}_{i=1}^N$

Limit Points: For any initial condition $\mathbf{y}(0)$, there are sequences $(s_k)_{k\in\mathbb{N}}$ so that

$$\lim_{s_k\to\infty} \mathbf{y}(s_k) = \mathbf{y}^*$$

Main Theorem (Baptista et al., 2025)

The limit points $\mathbf{y}^*$ are attained at one the data points $\mathbf{x}_0^i$ or on the boundaries of the Voronoi tesselation $\{\partial V(\mathbf{x}_0^i)\}_{i=1}^N$

Corollary: Exponential Convergence (Baptista et al., 2025)

When the limit point is $\mathbf{y}^* = \mathbf{x}_0^i$ for some i, then for all $s \geq s^*$

$$|\mathbf{y}(s) - \mathbf{y}^*| \leq Ke^{-s},$$

for constant K depending on the data p_0 and initial condition

Table of Contents

Reverse-Time Dynamical System

Lemma: Dynamics live in compact sets

$$|\mathbf{y}(s)| \leq \max\left(|\mathbf{y}(0)|, \max_{1\leq i\leq N} |\mathbf{x}_0^i|\right), \qquad \forall s$$

Takeaway: We can extract limit points $\mathbf{y}^* = \sup_{s\in\mathbb{R}_+} \mathbf{y}(s)$ from convergent subsequences

$$\mathbf{y}^* \in B(0, R), \qquad R = \max_{1\leq i\leq N} |\mathbf{x}_0^i|$$

Reverse-Time Dynamical System

Lemma: Dynamics live in compact sets

$$|\mathbf{y}(s)| \leq \max\left(|\mathbf{y}(0)|, \max_{1\leq i\leq N} |\mathbf{x}_0^i|\right), \qquad \forall s$$

Takeaway: We can extract limit points $\mathbf{y}^* = \sup_{s\in\mathbb{R}_+} \mathbf{y}(s)$ from convergent subsequences

$$\mathbf{y}^* \in B(0, R), \qquad R = \max_{1\leq i\leq N} |\mathbf{x}_0^i|$$

Lemma: Voronoi cells are invariant

For each $\delta > 0$ separation from the boundary, consider subset V^δ of each Voronoi cell

$$V^\delta(\mathbf{x}_0^i) := \{\mathbf{x} \in \mathbb{R}^d, |\mathbf{x} - \mathbf{x}_0^i| \leq |\mathbf{x} - \mathbf{x}_0^\ell| - \delta, \ell \neq i\}$$

If $\mathbf{y}(s^*) \in V^\delta(\mathbf{x}_0^i) \cap B(0, R)$ for some time $s^*(N, R, \delta)$, then

$$\mathbf{y}(s) \in V^\delta(\mathbf{x}_0^i), \qquad \forall s \geq s^*$$

Collapse Onto the Data Points

Main idea: dynamics are approximately linear within Voronoi cell

$$\begin{aligned}\frac{d(\mathbf{y}-\mathbf{x}_0^i)}{ds} = \frac{d\mathbf{y}}{ds} &= -(\mathbf{y}-\mathrm{y}_N(\mathbf{y},s)) \\ &= -(\mathbf{y}-\mathbf{x}_0^i) - (\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s)) \\ &\approx -(\mathbf{y}-\mathbf{x}_0^i)\end{aligned}$$

Collapse Onto the Data Points

Main idea: dynamics are approximately linear within Voronoi cell

$$\begin{aligned}\frac{d(\mathbf{y}-\mathbf{x}_0^i)}{ds} = \frac{d\mathbf{y}}{ds} &= -(\mathbf{y}-\mathrm{y}_N(\mathbf{y},s)) \\ &= -(\mathbf{y}-\mathbf{x}_0^i) - (\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s)) \\ &\approx -(\mathbf{y}-\mathbf{x}_0^i)\end{aligned}$$

Reasons: From cell invariance, the weights $w_j(\mathbf{y},s) \propto \exp(-e^{2s}|\mathbf{y}-\mathbf{x}_0^i|^2)$ for $\mathbf{y} \in V^\delta(\mathbf{x}_0^i)$ are

$$w_i(\mathbf{y},s) \approx 1, \qquad w_\ell(\mathbf{y},s) \approx 0$$

Collapse Onto the Data Points

Main idea: dynamics are approximately linear within Voronoi cell

$$\begin{aligned}\frac{d(\mathbf{y}-\mathbf{x}_0^i)}{ds} = \frac{d\mathbf{y}}{ds} &= -(\mathbf{y}-\mathrm{y}_N(\mathbf{y},s)) \\ &= -(\mathbf{y}-\mathbf{x}_0^i) - (\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s)) \\ &\approx -(\mathbf{y}-\mathbf{x}_0^i)\end{aligned}$$

Reasons: From cell invariance, the weights $w_j(\mathbf{y},s) \propto \exp(-e^{2s}|\mathbf{y}-\mathbf{x}_0^i|^2)$ for $\mathbf{y} \in V^\delta(\mathbf{x}_0^i)$ are

$$w_i(\mathbf{y},s) \approx 1, \qquad w_\ell(\mathbf{y},s) \approx 0$$

The nonlinear part of dynamics is small:

$$\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s) = (1-w_i(\mathbf{y},s))\mathbf{x}_0^i + \sum_{\ell \neq i} w_\ell(\mathbf{y},s)\mathbf{x}_0^\ell \approx 0$$

Collapse Onto the Data Points

Main idea: dynamics are approximately linear within Voronoi cell

$$\begin{aligned}\frac{d(\mathbf{y}-\mathbf{x}_0^i)}{ds} = \frac{d\mathbf{y}}{ds} &= -(\mathbf{y}-\mathrm{y}_N(\mathbf{y},s)) \\ &= -(\mathbf{y}-\mathbf{x}_0^i) - (\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s)) \\ &\approx -(\mathbf{y}-\mathbf{x}_0^i)\end{aligned}$$

Reasons: From cell invariance, the weights $w_j(\mathbf{y},s) \propto \exp(-e^{2s}|\mathbf{y}-\mathbf{x}_0^i|^2)$ for $\mathbf{y} \in V^\delta(\mathbf{x}_0^i)$ are

$$w_i(\mathbf{y},s) \approx 1, \qquad w_\ell(\mathbf{y},s) \approx 0$$

The nonlinear part of dynamics is small:

$$\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y},s) = (1-w_i(\mathbf{y},s))\mathbf{x}_0^i + \sum_{\ell \neq i} w_\ell(\mathbf{y},s)\mathbf{x}_0^\ell \approx 0$$

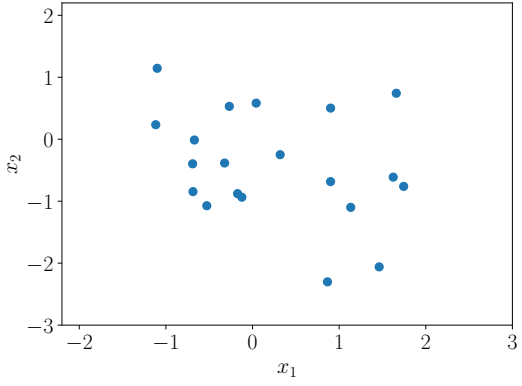
Takeaway: Exponential convergence within Voronoi cell $V(\mathbf{x}_0^i)$

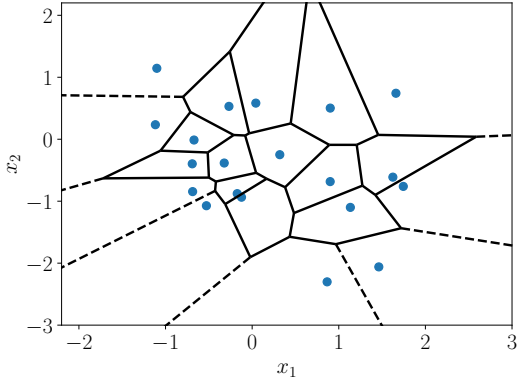
$$|\mathbf{y}(s)-\mathbf{x}_0^i| \leq Ke^{-s}, \qquad \text{for all } s \geq s^*$$

Table of Contents

Data with Voronoi Tesslations

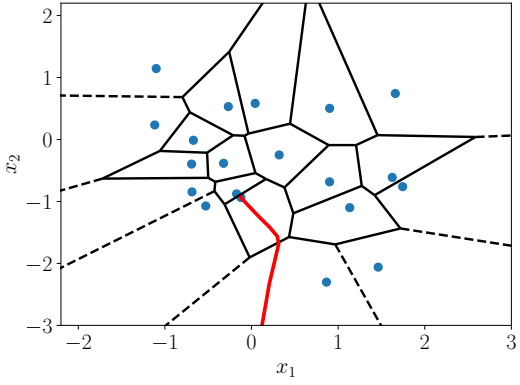
Data: $N = 20$ i.i.d. samples from $\mathcal{N}(0, I_2)$

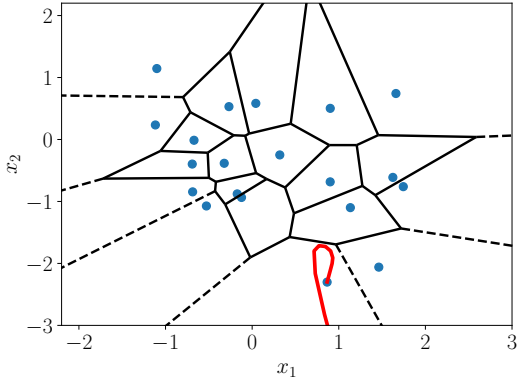




ODE Dynamics with Voronoi Tesselations

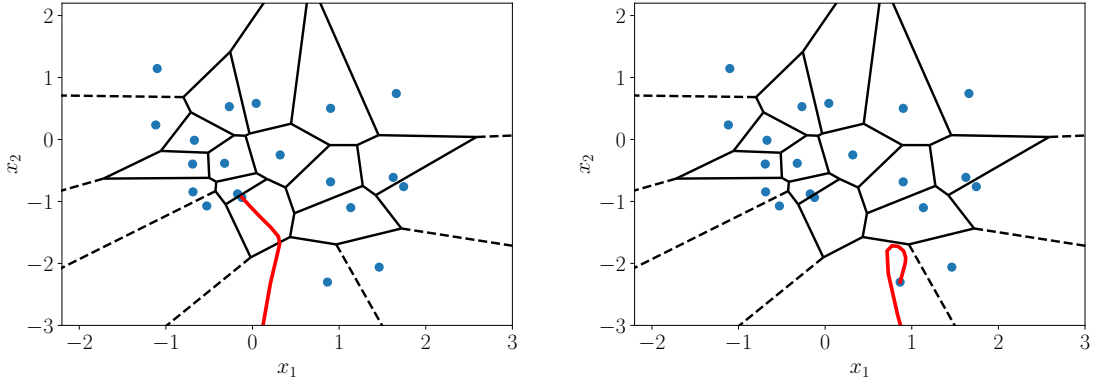
Integrate ODE with empirical score using $N = 20$ i.i.d. samples from $\mathcal{N}(0, I_2)$





ODE Dynamics with Voronoi Tesselations

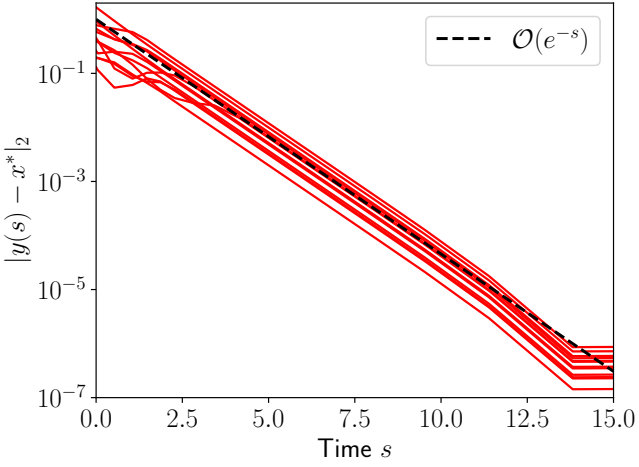
Integrate ODE with empirical score using $N = 20$ i.i.d. samples from $\mathcal{N}(0, I_2)$



Takeaway: Dynamics cross boundaries and explore before change in direction and collapse

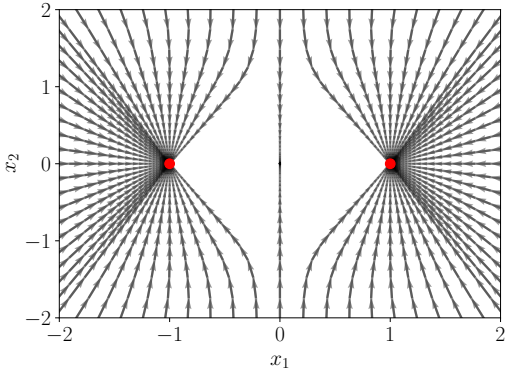
Fast Convergence

- Measured the Euclidean distance of each trajectory to its limit point
- Dynamics match the expected exponential convergence rate



Trajectories Can Remain On Hyper-Planes

- Trajectories of the ODE starting from initial conditions along a square around data
- Most trajectories collapse onto the $N = 2$ data points (red)
- Some trajectories remain on Voronoi boundaries



Explicit Regularization

Tikhonov-regularized score matching problem:

$$s^*_{\text{reg}} \in \arg\min_s \int_0^T \mathbb{E}|s(\mathbf{x}, t) - \nabla \log p(\mathbf{x}, t)|^2 + \gamma^2(t)\mathbb{E}|s(\mathbf{x}, t)|^2 dt.$$

- Objective can also be minimized via denoising score matching

Explicit Regularization

Tikhonov-regularized score matching problem:

$$s^*_{\text{reg}} \in \arg\min_s \int_0^T \mathbb{E}|s(\mathbf{x}, t) - \nabla \log p(\mathbf{x}, t)|^2 + \gamma^2(t)\mathbb{E}|s(\mathbf{x}, t)|^2 dt.$$

- Objective can also be minimized via denoising score matching

Optimal regularized score function

For empirical p_0 with $\gamma^2(t)\sigma^2(t) = c$, the score is

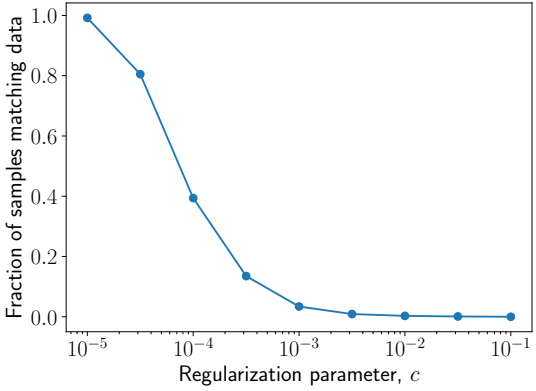
$$s^*_{\text{reg}}(\mathbf{x}, t) = -\frac{1}{\sigma^2(t) + c} \sum_{i=1}^{N} (\mathbf{x} - \mathbf{x}_0^i) w_i(\mathbf{x}, t)$$

The score remains bounded:

$$s^*_{\text{reg}}(\mathbf{x}, t) \to \frac{-(\mathbf{x} - \mathbf{x}_0^i)}{c}, \text{ as } t \to 0$$

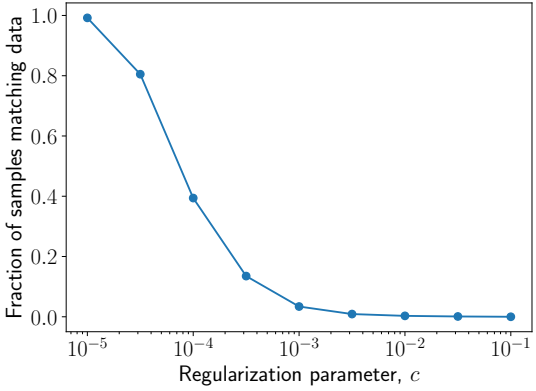
Memorization Versus Regularization using Tikhonov

- Evaluated the fraction of 2000 generated samples $\mathbf{x}(0)$ that match the data samples
- Compared different regularization parameters $c \in [10^{-5}, 10^{-1}]$



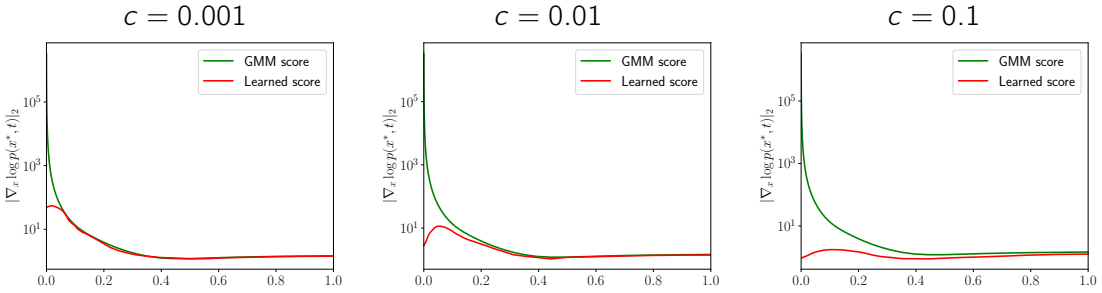
Memorization Versus Regularization using Tikhonov

- Evaluated the fraction of 2000 generated samples $\mathbf{x}(0)$ that match the data samples
- Compared different regularization parameters $c \in [10^{-5}, 10^{-1}]$



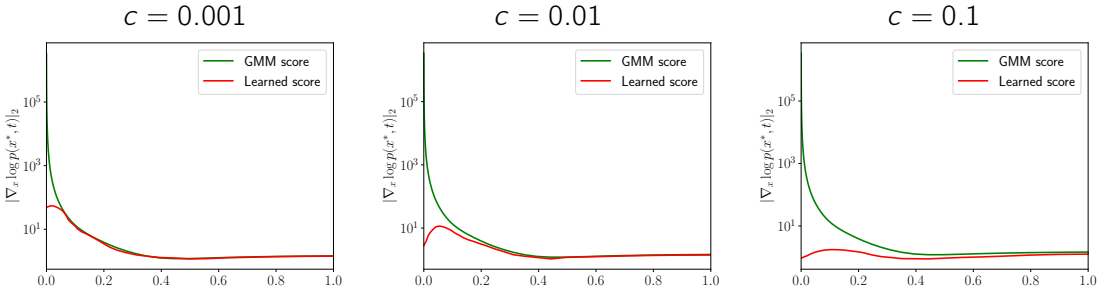
Takeaway: Increasing Tikhonov regularization on Gaussian mixture prevents memorization

Regularization and Learned Score using Tikhonov



Time-dependence of the learned score function $s(x^*, t)$ at fixed x^* (disjoint from the data)

Regularization and Learned Score using Tikhonov

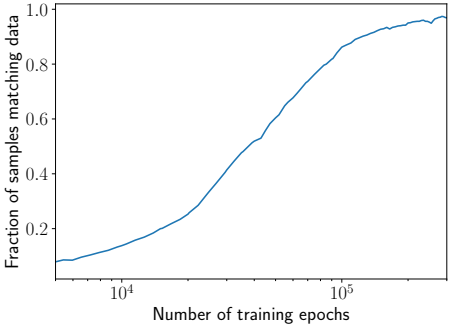


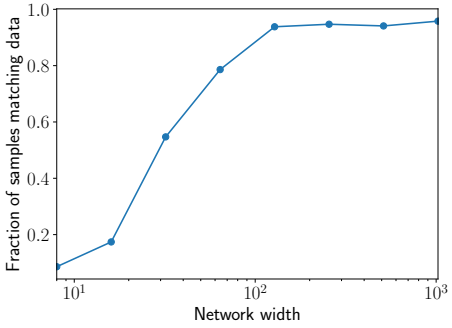
Time-dependence of the learned score function $s(x^*, t)$ at fixed x^* (disjoint from the data)

Takeaway: Increasing Tikhonov regularization reduces singular behaviour in Gaussian mixture

Memorization Versus Regularization using Neural Networks

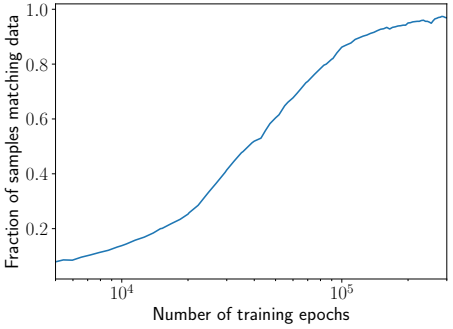
- Parameterized the score using a three-layer feedforward NN
- Evaluated the effect of increasing training iterations and model parameters (NN width)

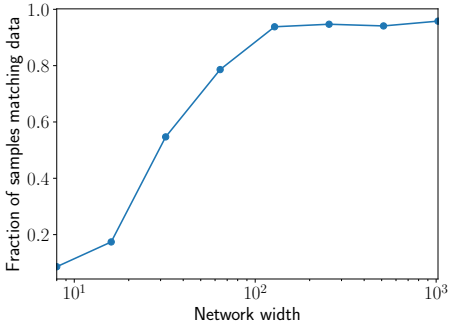




Memorization Versus Regularization using Neural Networks

- Parameterized the score using a three-layer feedforward NN
- Evaluated the effect of increasing training iterations and model parameters (NN width)





Takeaway: Early stopping in training and under-parameterization avoids memorization

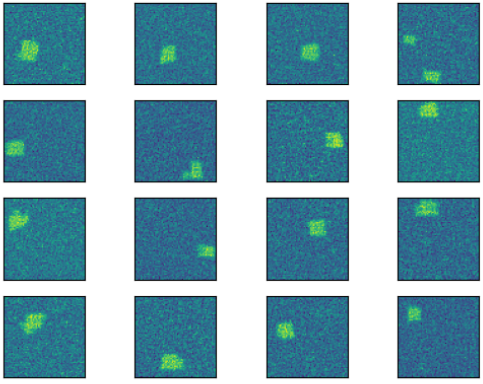
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

$2k$ epochs



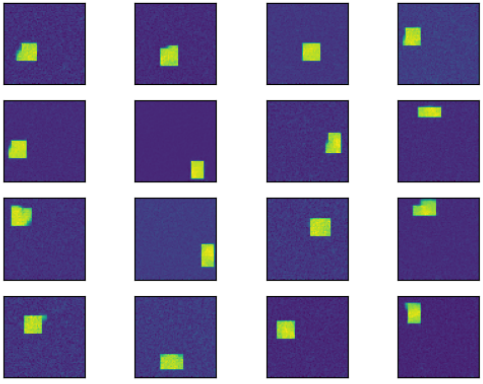
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

$4k$ epochs



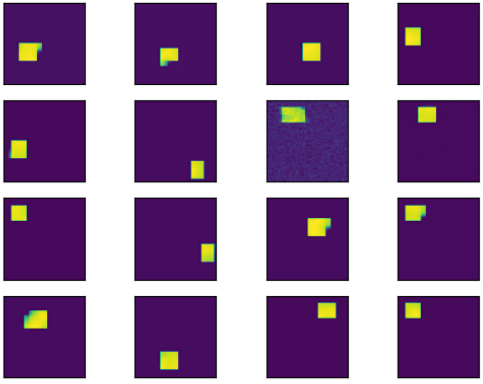
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

$6k$ epochs



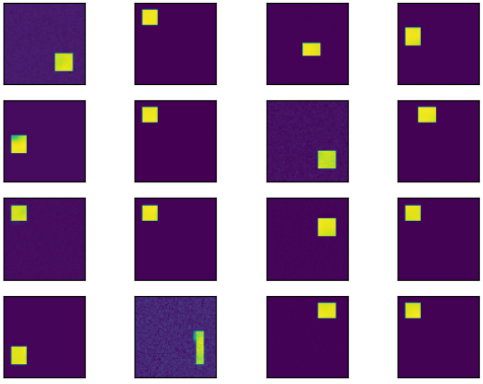
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

$10k$ epochs



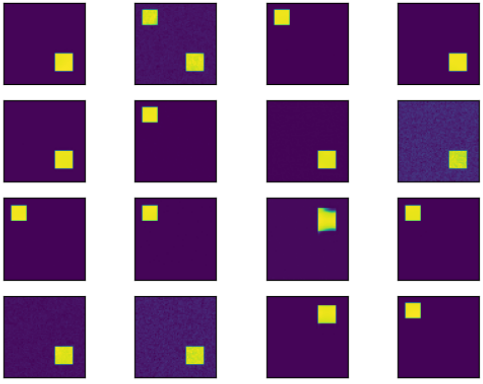
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

$20k$ epochs



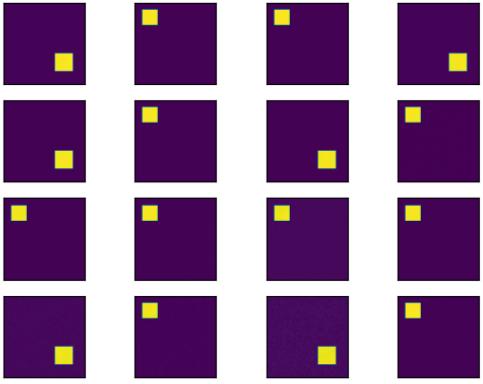
Collapse Is Observed With Images

Imaging example:

- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process

Training data

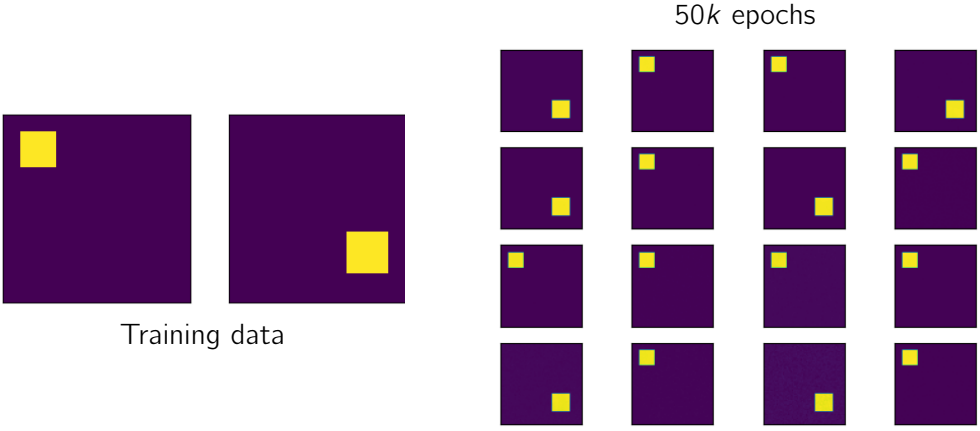
$50k$ epochs



Collapse Is Observed With Images

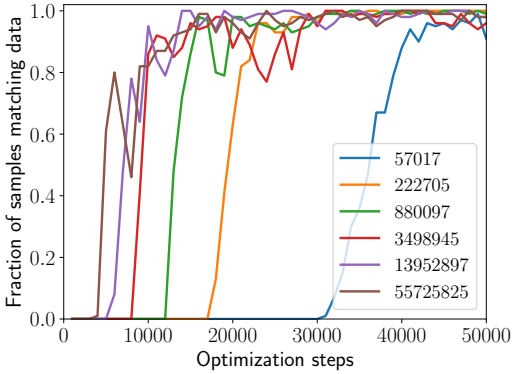
Imaging example:

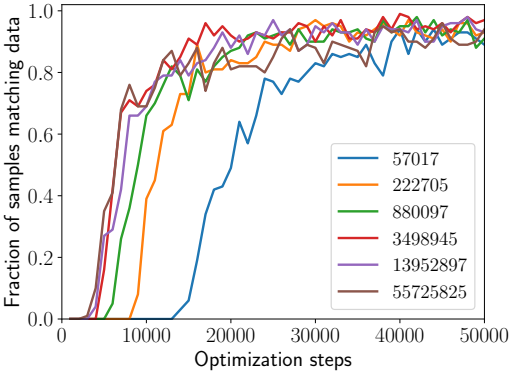
- Learned score function using EDM model (Karras et al., 2022) with U-Net architecture
- Training set of $N = 2$ images of small squares embedded in empty background
- Generated samples after each epoch with fixed noise process



Takeaway: Early stopping of training is one way to prevent data collapse

Collapse Is Observed With Images





Fraction of memorized samples. Legend indicates number of parameters in a U-Net model for the score. The left plot uses $N = 2$ training samples while the right plot uses $N = 8$.

Takeaway: Using fewer model parameters also prevents memorization

Table of Contents

Summary and Outlook

Main ideas

- Empirical score function has closed form expression
- Limit points of dynamics with empirical score contain data and Voronoi boundaries
- Dynamics converge exponentially fast to training data

Future work

- Dynamics with regularized score functions
- Explicit regularization for conditioning

Summary and Outlook

Main ideas

- Empirical score function has closed form expression
- Limit points of dynamics with empirical score contain data and Voronoi boundaries
- Dynamics converge exponentially fast to training data

Future work

- Dynamics with regularized score functions
- Explicit regularization for conditioning

Thank You for your attention

Supported by AFOSR, DoD and von Kármán Instructorship

References I

- Baptista, Ricardo et al. (2025). "Memorization and Regularization in Generative Diffusion Models". In: *arXiv preprint arXiv:2501.15785*.
- Carlini, Nicolas et al. (2023). "Extracting training data from diffusion models". In: *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270.
- Gu, Xiangming et al. (2023). "On Memorization in Diffusion Models". In: *arXiv:2310.02664*.
- Karras, Tero et al. (2022). "Elucidating the design space of diffusion-based generative models". In: *Advances in Neural Information Processing Systems* 35, pp. 26565–26577.
- Li, Sixu et al. (2024). "A good score does not lead to a good generative model". In: *arXiv:2401.04856*.
- Ramesh, Aditya et al. (2022). "Hierarchical text-conditional image generation with CLIP latents". In: *arXiv:2204.06125* 1.2, p. 3.
- Scarvelis, Christopher et al. (2023). "Closed-form diffusion models". In: *arXiv:2310.12395*.
- Somepalli, Gowthami et al. (2023). "Understanding and mitigating copying in diffusion models". In: *Advances in Neural Information Processing Systems* 36, pp. 47783–47803.

References II

- Vincent, Pascal (2011). "A connection between score matching and denoising autoencoders". In: *Neural computation* 23.7, pp. 1661–1674.
- Wan, Zhong Yi et al. (2023). "Debias coarsely, sample conditionally: Statistical downscaling through optimal transport and probabilistic diffusion models". In: *Advances in Neural Information Processing Systems* 36, pp. 47749–47763.

Similar Behaviour with Conditioning

Consider variance exploding process $d\mathbf{x}_t = \sqrt{g(t)}d\mathbf{w}_t,\ \mathbf{x}_t|\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0; \sigma^2(t))$
Note that diffusion is in $\mathbf{x}$ with $\mathbf{y}$ fixed

Optimal empirical score for conditional distributions $p(\mathbf{x}_0|\mathbf{y})$ (Gu et al., 2023)

$$s^* \in \arg\min_s \int_0^T \mathbb{E}|s(\mathbf{x}, \mathbf{y}, t) - \nabla_{\mathbf{x}} \log p(\mathbf{x}, t|\mathbf{x}_0)|^2 dt$$

For $p_0 = \frac{1}{N}\sum_{i=1}^N \delta_{(\mathbf{x}_0^i, \mathbf{y}^i)}$ with paired samples $\{\mathbf{x}_0^i, \mathbf{y}^i\} \sim p(\mathbf{x}_0, \mathbf{y})$, the minimizer is

$$s^*(\mathbf{x}, \mathbf{y}^*, t) = -\frac{1}{\sigma^2(t)} \sum_{i:\mathbf{y}_i=\mathbf{y}^*} (\mathbf{x} - \mathbf{x}_0^i) w_i(\mathbf{x}, t),$$

with normalized weights $w_i(\mathbf{x}, t) \propto \exp\left(-\frac{|\mathbf{x}-\mathbf{x}_0^i|^2}{2\sigma^2(t)}\right)$

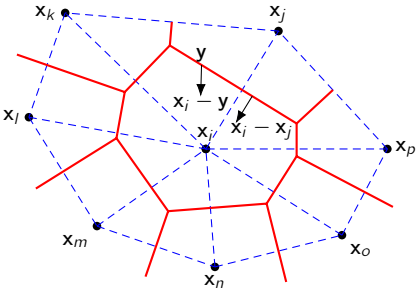
Takeaway:

- Empirical score has the same form as for unconditioned settings
- We will focus on the unconditioned setting today

Main Idea Behind Set Invariance

1. For points on the boundary $\mathbf{y} \in \partial V^{\delta}(\mathbf{x}_0^i) \cap B(0, R)$ and neighboring points $\mathbf{x}_0^j$,

$$\left\langle \mathbf{x}_0^i - \mathbf{y}, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle \geq \alpha > 0$$



2. After sufficient time, the weights for $\mathbf{y} \in V^{\delta}(\mathbf{x}_0^i)$ are

$$w_i(\mathbf{y}, s) \approx 1, \qquad w_\ell(\mathbf{y}, s) \approx 0$$

3. The nonlinear part of dynamics behave similar to $\mathbf{x}_0^i$

$$\begin{aligned} \left|\mathbf{x}_0^i - \mathrm{y}_N(\mathbf{y}, s)\right| &\leq |1 - w_i(\mathbf{y}, s)||\mathbf{x}_0^i| + \sum_{\ell \neq i} |w_\ell(\mathbf{y}, s)||\mathbf{x}_0^\ell| \\ &\leq \frac{\alpha}{2 \max_{j,k} |\mathbf{x}_0^j - \mathbf{x}_k|} \end{aligned}$$

4. Inner product of the dynamics with boundary faces $\mathbf{x}_0^i - \mathbf{x}_0^j$ is bounded from below

$$\begin{aligned} \left\langle \frac{d\mathbf{y}}{ds}, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle &= \left\langle \mathbf{x}_0^i - \mathbf{y}, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle + \left\langle \mathrm{y}_N(\mathbf{y}, s) - \mathbf{x}_0^i, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle \\ &\geq \left\langle \mathbf{x}_0^i - \mathbf{y}, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle - |\left\langle \mathrm{y}_N(\mathbf{y}, s) - \mathbf{x}_0^i, \mathbf{x}_0^i - \mathbf{x}_0^j \right\rangle| \geq \alpha/2 > 0 \end{aligned}$$