2026.9
9
Wed

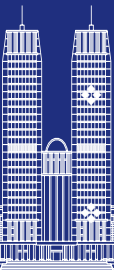
数学科学学院、大数据学院
School of Mathematical Sciences & School of Data Science
高卫国 Weiguo Gao

Algorithmic Analysis of Generative Models from a Numerical Analysis Perspective

Joint work with 厉茗 Ming Li (Fudan)

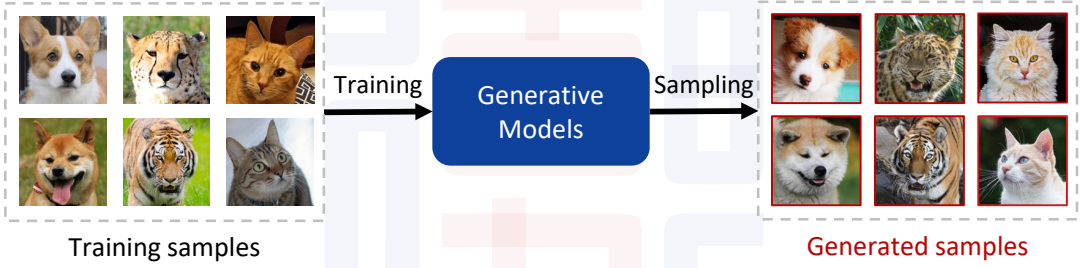
Outline

- Generative Models
  - 1.1 GANs and Flow-Based Models
- Algorithmic Mechanisms and Acceleration
  - 2.1 Diffusion Distillation
- Evolution of Discriminator and Generator Gradients in GAN Training
  - 3.1 What drives the particles to capture all the modes?
  - 3.2 Why does GAN collapse?
- Subspace Fidelity of Flow Matching Models
  - 4.1 How do flow matching models memorize?
  - 4.2 How do flow matching models generalize?
- Diffusion Trajectory Distillation
  - 5.1 Linear Gaussian Regime
  - 5.2 Nonlinear Gaussian Mixture Mechanism
- Conlcuding Remarks



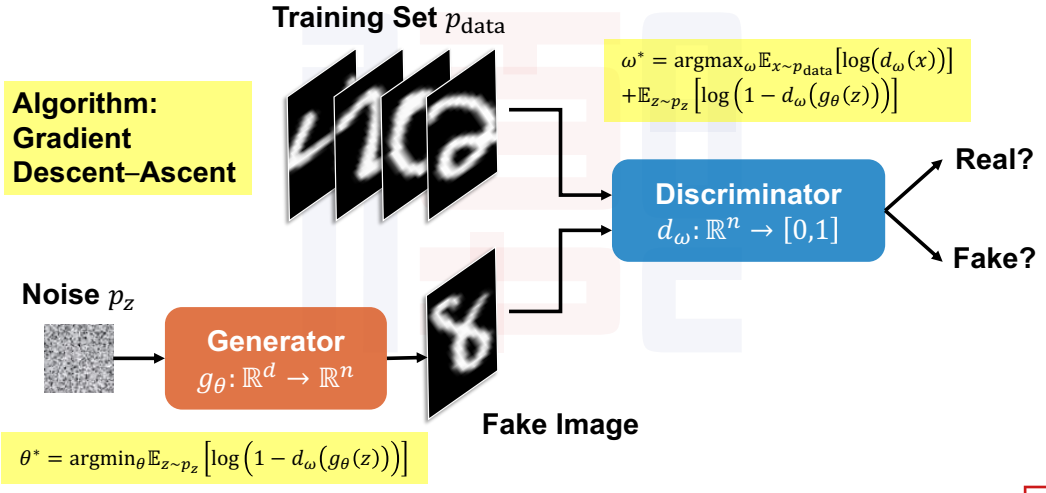
Goal: Learn data distribution to generate new samples similar to the training samples.

E.g. image generation (DALL · E), video generation (Sora), text generation (ChatGPT).



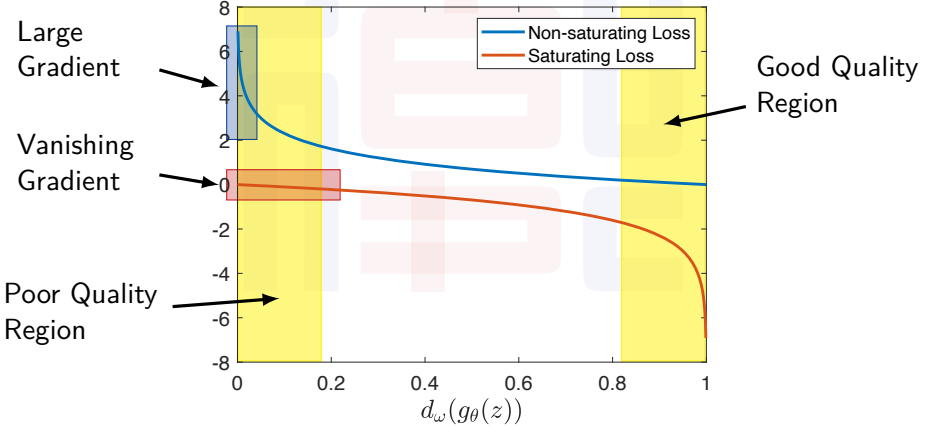
Model	Training stability	Sample quality	Sampling speed
GANs [Goodfellow et al. 2014]	Low	High	Fast
VAEs [Kingma & Welling 2014]	High	Low	Fast
Diffusion Models [Song et al. 2021]	High	High	Slow

❖ Vanilla GAN loss function: $\min_{\theta}\max_{\omega}\mathbb{E}_{x\sim p_{\text{data}}}[\log(d_\omega(x))]+\mathbb{E}_{z\sim p_z}[\log(1-d_\omega(g_\theta(z))]$.



❖ NSGAN only modifies the generator loss function:

$$\theta^* = \text{argmin}_\theta \, \mathbb{E}_{z\sim p_z}[\underbrace{\log(1 - d_\omega(g_\theta(z)))}_{\text{Saturating loss}}] \quad \Rightarrow \quad \theta^* = \text{argmin}_\theta \, \mathbb{E}_{z\sim p_z}[\underbrace{-\log d_\omega(g_\theta(z))}_{\text{Non-saturating loss}}]$$



Key idea: Learn to reverse a forward process that gradually corrupts data with noise.

Forward process: Gradually adds Gaussian noise to data $\boldsymbol{z}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.

Sample	$\boldsymbol{x}_0$	$\boldsymbol{z}_1$	$\boldsymbol{z}_2$	$\boldsymbol{z}_3$	$\boldsymbol{z}_4$	$\boldsymbol{z}_5$	$\boldsymbol{z}_6$	$\boldsymbol{z}_7$
Data	$\alpha_0 \boldsymbol{x}_0$	$\alpha_1 \boldsymbol{x}_0$	$\alpha_2 \boldsymbol{x}_0$	$\alpha_3 \boldsymbol{x}_0$	$\alpha_4 \boldsymbol{x}_0$	$\alpha_5 \boldsymbol{x}_0$	$\alpha_6 \boldsymbol{x}_0$	$\alpha_7 \boldsymbol{x}_0$
Noise	$\sigma_0 \boldsymbol{\varepsilon}$	$\sigma_1 \boldsymbol{\varepsilon}$	$\sigma_2 \boldsymbol{\varepsilon}$	$\sigma_3 \boldsymbol{\varepsilon}$	$\sigma_4 \boldsymbol{\varepsilon}$	$\sigma_5 \boldsymbol{\varepsilon}$	$\sigma_6 \boldsymbol{\varepsilon}$	$\sigma_7 \boldsymbol{\varepsilon}$

Noise schedule: $\{\alpha_t\}_{t=0}^T$ and $\{\sigma_t\}_{t=0}^T$ control the balance between signal and noise,

$$\alpha_0 = 1, \quad \alpha_T = 0, \quad \alpha_t^2 + \sigma_t^2 = 1.$$

Objective: Train a neural network $\hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t)$ to predict clean $\boldsymbol{x}_0$ from its noisy observation $\boldsymbol{z}_t$,

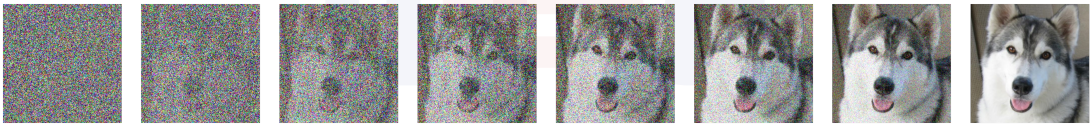
$$\mathcal{L}_{\text{denoise}}(\boldsymbol{\eta}) = \mathbb{E}_{t \sim \mathcal{U}(0,T)} \left[w_t \cdot \mathbb{E}_{\boldsymbol{x}_0 \sim p_0, \boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \| \hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t) - \boldsymbol{x}_0 \|_2^2 \right].$$

Recall: Forward process $\boldsymbol{z}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\varepsilon}$, $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. **How to reverse it?**

Sampling process[1]**:** Start from $\boldsymbol{z}_T \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, then iteratively estimate the previous state

$$\boldsymbol{z}_{t-1} = \alpha_{t-1} \underbrace{\hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t)}_{\text{estimated data}} + \sigma_{t-1} \cdot \underbrace{\left(\frac{\boldsymbol{z}_t - \alpha_t \hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t)}{\sigma_t} \right)}_{\text{estimated noise}}, \quad t = T, \ldots, 1.$$

- ❖ Estimation is only locally valid at each step.
- ❖ Requires many steps (e.g., $1{,}000$) to recover data from noise $\rightsquigarrow$ slow and costly sampling.



$\boldsymbol{z}_7$ $\boldsymbol{z}_6$ $\boldsymbol{z}_5$ $\boldsymbol{z}_4$ $\boldsymbol{z}_3$ $\boldsymbol{z}_2$ $\boldsymbol{z}_1$ $\boldsymbol{z}_0$

[1] DDIM (deterministic): [Song et al. 2021]. See also DDPM (stochastic): [Ho et al. 2020].

- Source distribution: p_0 (e.g., standard Gaussian); Target distribution: p_1.
- Velocity field $\boldsymbol{u}_t$ transforms p_0 to p_1 via

$$\frac{\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x})}{\mathrm{d}t} = \boldsymbol{u}_t(\boldsymbol{\phi}_t(\boldsymbol{x})), \quad t \in [0,1], \quad \boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x} \sim p_0, \quad \boldsymbol{\phi}_1(\boldsymbol{x}) \sim p_1.$$

- The solution $\boldsymbol{\phi}_t$ is called **flow**. The distribution of $\boldsymbol{\phi}_t$ is p_t, satisfying the continuity equation

$$\frac{\partial p_t(\boldsymbol{x})}{\partial t} + \nabla \cdot \big(p_t(\boldsymbol{x})\boldsymbol{u}_t(\boldsymbol{x})\big) = \mathbf{0}, \quad t \in [0,1].$$

- Approximate $\boldsymbol{u}_t$ with a neural network $\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta})$.
- Flow matching objective:

$$\mathcal{L}_{\mathsf{FM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}[0,1],\boldsymbol{x}\sim p_t(\boldsymbol{x})} \big[\|\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{u}_t(\boldsymbol{x})\|_2^2\big].$$

Recall: Flow ODE & continuity equation

$$\text{(Flow ODE)} \quad \frac{\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x})}{\mathrm{d}t} = \boldsymbol{u}_t(\boldsymbol{\phi}_t(\boldsymbol{x})); \qquad \text{(Continuity equation)} \quad \frac{\partial p_t(\boldsymbol{x})}{\partial t} + \nabla \cdot \big(p_t(\boldsymbol{x})\boldsymbol{u}_t(\boldsymbol{x})\big) = \mathbf{0}.$$

- **Addressing unavailability of p_1 and non-uniqueness of p_t**: For $\boldsymbol{y} \sim p_1$, define

$$p_t(\boldsymbol{x}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}; t\boldsymbol{y}, (1-t)^2\boldsymbol{I}).$$

- **Addressing non-uniqueness of $\boldsymbol{u}_t$**: Define the flow as Optimal Transport paths

$$\boldsymbol{\phi}_t(\boldsymbol{x}|\boldsymbol{y}) = (1-t)\boldsymbol{x} + t\boldsymbol{y}.$$

- The conditional velocity field:

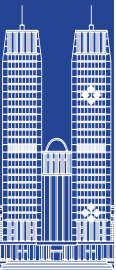
$$\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{y}) = \frac{\boldsymbol{y} - \boldsymbol{x}}{1-t}.$$

- The conditional flow matching objective:

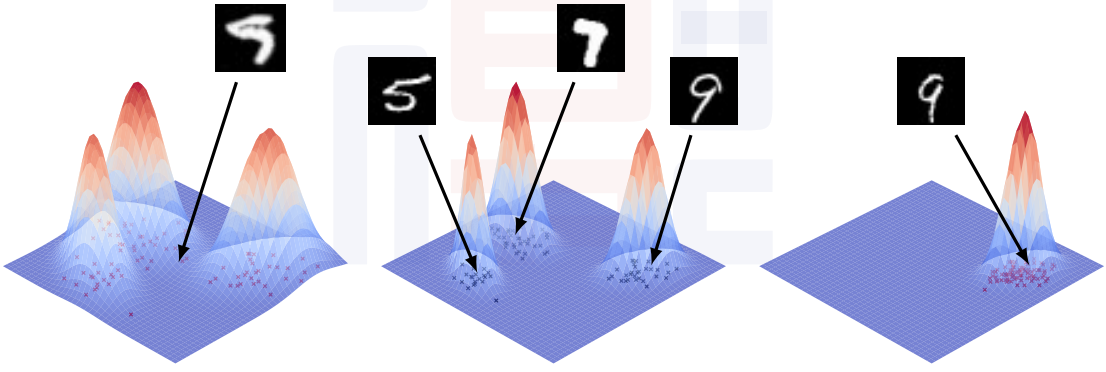
$$\mathcal{L}_{\mathsf{CFM}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{t\sim\mathcal{U}[0,1], \boldsymbol{y}\sim p_1(\boldsymbol{y}), \boldsymbol{x}\sim p_t(\boldsymbol{x}|\boldsymbol{y})} \left[\|\boldsymbol{v}_t(\boldsymbol{x};\boldsymbol{\theta}) - \boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{y})\|_2^2\right].$$

Outline

- Generative Models
  - 1.1 GANs and Flow-Based Models
- Algorithmic Mechanisms and Acceleration
  - 2.1 Diffusion Distillation
- Evolution of Discriminator and Generator Gradients in GAN Training
  - 3.1 What drives the particles to capture all the modes?
  - 3.2 Why does GAN collapse?
- Subspace Fidelity of Flow Matching Models
  - 4.1 How do flow matching models memorize?
  - 4.2 How do flow matching models generalize?
- Diffusion Trajectory Distillation
  - 5.1 Linear Gaussian Regime
  - 5.2 Nonlinear Gaussian Mixture Mechanism
- Conlcuding Remarks



- **Mode mixture** [An et al. 2020]: Samples blend different modes, producing unrealistic data.
- **Mode collapse** [Goodfellow 2017]: Samples are limited in variety compared to the training set.



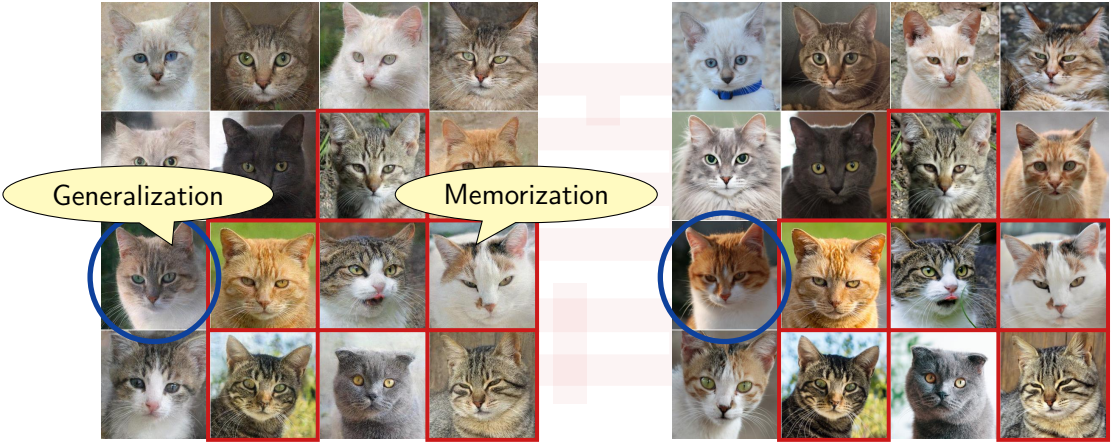
Mode mixture | Training set | Mode collapse

- Mode collapse and mode mixture are treated as **separate, independent** issues.
- Mode collapse is viewed as a **complete training failure**.

	Practical considerations	Theoretical frameworks
Mode mixture	Optimal transport [An et al. 2020; Gu et al. 2021; Lei et al. 2019];	Singularity theory from optimal transport theory [An et al. 2020; Gu et al. 2021; Lei et al. 2019].
	Rejection sampling [Tanielian et al. 2020].	
Mode collapse, nonconvergence, relation to particle models	Different metrics [Arjovsky et al. 2017; Li et al. 2017; Mao et al. 2017; Nowozin et al. 2016];	Static landscape analysis [Sun et al. 2020];
	Joint architecture [Larsen et al. 2016];	Infinite-width generator and discriminator [Becker et al. 2022; No et al. 2021];
	Multiple generators/discriminators [Ghosh et al. 2018; Nguyen et al. 2017];	Simple assumptions on data [Feizi et al. 2017; Mescheder et al. 2018];
	Normalization [Ba et al. 2016; Ioffe & Szegedy 2015; Miyato et al. 2018].	Relation to particle models [Franceschi et al. 2023; Huang & Zhang 2023].

One group of cats is real, while the other is generated by a generative model. Which is which?

One group of cats is real, while the other is generated by a generative model. Which is which?



Left: Generated cats. Right: Real cats.

Red frames : Generated cats that replicate real ones.

	Diffusion models	Flow matching models
Manifold learning	Linear spaces [Chen et al. 2023b];	Ours: Generated samples lie near or in sample data subspaces.
	Low-rank Gaussian mixtures [Wang et al. 2024a].	
Memorization	Empirical investigations [Yoon et al. 2023];	Ours: The optimal velocity field leads to memorization.
	Theoretical investigations [Li et al. 2024b].	
Generation paths	Feature emergence [Wang & Vastola 2023a; Biroli et al. 2024];	Path straightness [Lee et al. 2023; Wang et al. 2024b];
	Path curveness [Wang et al. 2024b].	Ours: Rigorous proof of path straightness and hierarchy emergence.
Generalization	Implicit bias [Kadkhodaie et al. 2024];	Ours: The suboptimal velocity field leads to generalization.
	Statistical bounds [Li et al. 2024a].	
Error bounds	Stochastic sampling [Chen et al. 2023a; Benton et al. 2024b].	Deterministic ones [Benton et al. 2024a];
		Ours: Sample-specific error bound.

Two mainstream approaches:

- **Solver-based:** Solve sampling process (ODE/SDE) using advanced numerical solvers.
- **Distillation-based:** Train a fast student model to mimic the pretrained teacher model.

	Solver-based	**Distillation-based**
Sampling steps	Adaptive (20–50)	Fixed (1–8)
Training cost	None	Additional training required
Flexibility	Tunable at test time	Tuned at training time
Model type	Reuses original model	Trains a separate student model
Representatives	Heun's method [Karras et al. 2022]	Diffusion trajectory distillation [Gu et al. 2023; Luhman & Luhman 2021; Salimans & Ho 2022; Song et al. 2023]
	Pseudo linear multistep [Liu et al. 2022]	Distribution matching distillation [Yin et al. 2024a; Yin et al. 2024b]
	DPM-Solver & DPM-Solver++ [Lu et al. 2022a; Lu et al. 2022b]	Adversarial distillation [Sauer et al. 2024]
	Unified predictor-corrector [Zhao et al. 2023]	

Key idea: Train a student model to approximate the multiple teacher steps in a single step.

Definition (Composite operator of k steps)

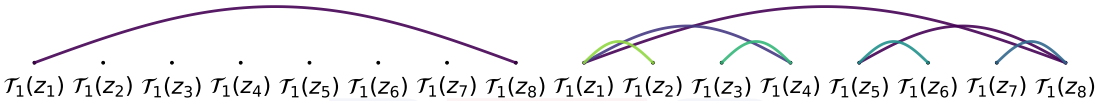
Assume the teacher performs the following DDIM update at time t:

$$\boldsymbol{f_\eta}(\boldsymbol{z}_t, t) := \alpha_{t-1} \cdot \hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t) + \sigma_{t-1} \cdot \left(\frac{\boldsymbol{z}_t - \alpha_t \hat{\boldsymbol{x}}_{\boldsymbol{\eta}}(\boldsymbol{z}_t, t)}{\sigma_t} \right).$$

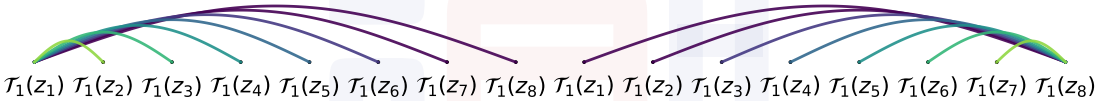
Then the composite operator over k teacher steps is defined as

$$\mathcal{T}_k(\boldsymbol{z}_t) := \boldsymbol{f_\eta}(\boldsymbol{f_\eta}(\cdots \boldsymbol{f_\eta}(\boldsymbol{z}_t, t) \cdots, t-k+2), t-k+1).$$

1. **Initialize** student model $\boldsymbol{A}_t^{\text{st}}(\boldsymbol{z}_t)$ to match the teacher model's one-step operator $\mathcal{T}_1(\boldsymbol{z}_t)$.
2. **Train** the student $\boldsymbol{A}_{t-k+1:t}^{\text{st}}(\boldsymbol{z}_t)$ to approximate $\mathcal{T}_k(\boldsymbol{z}_t)$.
3. **Repeat** distillation by reassigning student as teacher until only step $\boldsymbol{A}_{1:T}^{\text{st}}(\boldsymbol{z}_T)$ remains.



Vanilla distillation [Luhman & Luhman 2021] Progressive distillation [Salimans & Ho 2022]



Sequential consistency [Song et al. 2023] Sequential BOOT [Gu et al. 2023]

Method	**Target**	**Description**
Vanilla distillation	$\boldsymbol{A}^{\text{st}}_{1:T}(\boldsymbol{z}_T) \approx \mathcal{T}_T(\boldsymbol{z}_T)$	Approximates the entire trajectory in one step
Progressive distillation	$\boldsymbol{A}^{\text{st}}_{t-1:t}(\boldsymbol{z}_t) \approx \mathcal{T}_2(\boldsymbol{z}_t)$	Iteratively merges pairs to reduce the total steps
Sequential consistency	$\boldsymbol{A}^{\text{st}}_{1:t}(\boldsymbol{z}_t) \approx \mathcal{T}_t(\boldsymbol{z}_t)$	Sequentially maps any timestep $\boldsymbol{z}_t$ directly to $\boldsymbol{z}_0$
Sequential BOOT	$\boldsymbol{A}^{\text{st}}_{T-k+1:T}(\boldsymbol{z}_T) \approx \mathcal{T}_k(\boldsymbol{z}_T)$	Fixes input $\boldsymbol{z}_T$ and sequentially extends target to $\boldsymbol{z}_0$

Question: Which strategy might be the "best"?

Outline

- Generative Models
  - 1.1 GANs and Flow-Based Models
- Algorithmic Mechanisms and Acceleration
  - 2.1 Diffusion Distillation
- Evolution of Discriminator and Generator Gradients in GAN Training
  - 3.1 What drives the particles to capture all the modes?
  - 3.2 Why does GAN collapse?
- Subspace Fidelity of Flow Matching Models
  - 4.1 How do flow matching models memorize?
  - 4.2 How do flow matching models generalize?
- Diffusion Trajectory Distillation
  - 5.1 Linear Gaussian Regime
  - 5.2 Nonlinear Gaussian Mixture Mechanism
- Conlcuding Remarks

Recall: Non-saturating generator loss $\mathcal{L}(\theta) = \mathbb{E}_{z\sim p_z}[-\log d_\omega(g_\theta(z))]$.

- Original NSGAN training procedure: Sample $z_i \sim p_z$ $(1 \le i \le m)$,

$$\theta' \leftarrow \theta + \nabla_\theta \frac{1}{m}\sum_{i=1}^{m} \log d_\omega\big(g_\theta(z_i)\big).$$

- Particle model interpretation of NSGAN:
  1. Generate particles $Z_i = g_\theta(z_i)$ $(1 \le i \le m)$;
  2. **Update the particles** (s: stepsize)

$$\hat{Z}_i = Z_i + s \cdot \nabla d_\omega(Z_i)/d_\omega(Z_i) \quad (1 \le i \le m);$$

  3. Apply the stop gradient operator to $\hat{Z}_i$ and update θ (MSE loss)

$$\theta' \leftarrow \theta - \nabla_\theta \frac{1}{m}\sum_{i=1}^{m} \big\|g_\theta(z_i) - \hat{Z}_i\big\|_2^2.$$

$\rightsquigarrow g_{\theta'}(z_i) \approx \hat{Z}_i.$

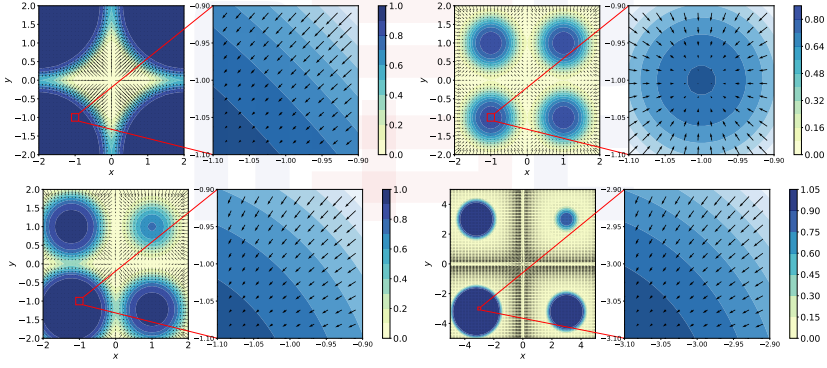
Theorem (The Limiting Scenarios)

Assume that the discriminator is optimal, i.e., $d^*_\omega(x) = p_{\text{data}}(x)/(p_{\text{data}}(x)+p_g(x))$. Denote $r(x) = p_{\text{data}}(x)/p_g(x)$. Where $r(x) \approx 0$, x is updated following approximately $\nabla \log\big(r(x)\big)$. Conversely, when $r(x) \gg 1$, x is updated following approximately $\nabla\big(-1/r(x)\big)$.

$$\frac{\nabla d^*_\omega(x)}{d^*_\omega(x)} = \nabla r(x) \cdot \frac{1}{r(x)(1+r(x))}.$$

- The value of $\log\big(r(x)\big)$ changes dramatically as $r(x)$ decreases from 1 to 0, leading to correspondingly large magnitudes of $\|\nabla \log\big(r(x)\big)\|_2$ when $r(x) \approx 0$.
  - When $p_g(x) \gg p_{\text{data}}(x)$, x is pushed towards distant points.
- Conversely, $\nabla\big(-1/r(x)\big)$ changes more gradually with increasing x, resulting in smaller magnitudes of $\|\nabla\big(-1/r(x)\big)\|_2$ when $r(x) \gg 1$.
  - When $p_g(x) \ll p_{\text{data}}(x)$, x tend to remain relatively stationary.

	Description	p_{data}	p_g
Case 1	Initialization with concentrated particles	$\mathcal{N}([\pm 1, \pm 1], 0.1\mathbf{I}_2)$	$\mathcal{N}([0,0], 0.2\mathbf{I}_2)$
Case 2	Particles covering all modes	$\mathcal{N}([\pm 1, \pm 1], 0.1\mathbf{I}_2)$	$\mathcal{U}([-2,2]\times[-2,2])$
Case 3	Particles concentrated near a single mode	$\mathcal{N}([\pm 1, \pm 1], 0.1\mathbf{I}_2)$	$\mathcal{N}([1,1], \mathbf{I}_2)$
Case 4	Globally separated modes	$\mathcal{N}([\pm 3, \pm 3], 0.1\mathbf{I}_2)$	$\mathcal{N}([3,3], 3\mathbf{I}_2)$



Let $r(x) = p_{\text{data}}(x)/p_g(x)$ be the density ratio and let f be a scalar function.

$$d^*_\omega(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}$$

The optimal discriminator

$$\hat{d}_\omega(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + f(r(x)) \cdot p_g(x)}$$

A class of suboptimal discriminators

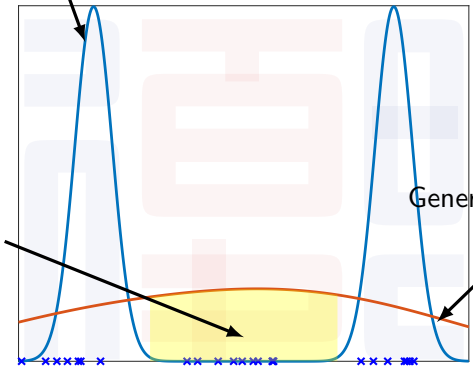
Proposition (Vector Field)

Under some regularity conditions,

$$\cos\theta = \frac{\left\langle \nabla \hat{d}_\omega(x), \nabla d^*_\omega(x) \right\rangle}{\left\| \nabla \hat{d}_\omega(x) \right\|_2 \left\| \nabla d^*_\omega(x) \right\|_2} = \text{sign}\left(\frac{f(r(x))}{r(x)} - f'(r(x)) \right).$$

Implication: There exists $\delta > 0$ such that whenever $r(x) < \delta$, the two vectors point in the same direction (the same configuration as in the fitting phase).

Real distribution p_{data}



Generated distribution p_g

The larger the area, the severer the mode mixture.

Probably mode mixture

$\rightsquigarrow$ What determines the area?

Notations:

- ν: Truncated Gaussian distribution $\mathcal{N}_r(0, I_n)$ in n-dimensional ball $\mathcal{B}_r(\mathbf{0})$.
- $\hat{\mu}$: Probability measure with PDF

$$p_{\text{data}}(x) = \frac{1}{N}\sum_{i=1}^{N} K_\sigma(x, x_i) := \frac{1}{N}\sum_{i=1}^{N} \frac{1}{(2\pi\sigma^2)^{n/2}} \cdot \exp\left(-\frac{\|x - x_i\|_2^2}{2\sigma^2}\right),$$

- g: A function that satisfies $g_{\#}\nu = \hat{\mu}$.

Theorem (Steepness is Provably Large, Informal)

We have $\mathcal{S}_g(x_*) > M$, where

$$M = \delta \cdot \sigma \cdot \sqrt{2\pi} \max_{0 \le \lambda \le 2} \min_{1 \le i \le N} \exp\left(\frac{\|\lambda \bar{x} - x_i\|_2^2}{2n\sigma^2}\right).$$

Here, $\bar{x} = \sum_{i=1}^{N} x_i/n$, and $\delta = \exp\left(-r^2/2\right)/\sqrt{2\pi}$.

Theorem (Large Steepness Indicates Less Severe Mode Mixture, Informal)

Assume that $p_{\text{data}} \sim \sum_{i=1}^{N} \mathcal{N}(\mu_i, \sigma^2)/N$. Let $\Phi(x)$ denote the cumulative distribution function (CDF) of the standard normal distribution $\mathcal{N}(0,1)$. And let $\Psi(x)$ be the CDF of the distribution $p_{\text{data}}(x)$. Furthermore, assume that the generator function g is increasing and satisfies $\mathcal{S}_g(x) \leq k$. Then, under some regularity conditions, the probability that the generated samples fall into the interval

$$\bigcup_{i=1}^{N} [\mu_i + 3\sigma, \mu_{i+1} - 3\sigma],$$

which indicates mode mixture, is at least

$$\sum_{i=1}^{N} \left(\Phi\left(\Phi^{-1}\left(\Psi\left(\frac{\mu_i + \mu_{i+1}}{2} \right) \right) + \frac{\mu_{i+1} - \mu_i - 3\sigma}{2k} \right) - \Phi\left(\Phi^{-1}\left(\Psi\left(\frac{\mu_i + \mu_{i+1}}{2} \right) \right) - \frac{\mu_{i+1} - \mu_i - 3\sigma}{2k} \right) \right).$$

Theorem (Intrinsic Connection of the Gradients, Informal))

Assume that there are infinitely many particles and the step size $s > 0$ is sufficiently small. Then, the Jacobian $J_{g_{\theta'}}(z)$ of the updated generator $g_{\theta'}$ satisfies

$$J_{g_{\theta'}}(z) = J_{g_\theta}(z) + s \cdot \nabla_x \Big(\frac{\nabla d_\omega}{d_\omega} \Big) \big(g_\theta(z) \big) \cdot J_{g_\theta}(z).$$

Theorem (Local Analysis of Steepness, Informal)

Assume that the separation condition parameter $\Delta = 8\sigma$ and the discriminator $d(x) = 1/2 - \|x - x_i\|_2/(8\sigma)$ for all $x \in B_{4\sigma}(x_i)$. Then, the steepness of the updated generator $g_{\theta'}$ satisfies

$$\mathcal{S}_{g_{\theta'}}(z) \le \Big(1 - \frac{s}{(4\sigma - r)^2} \Big) \cdot \mathcal{S}_{g_\theta}(z), \quad \text{for all } g_\theta(z) \in B_{2\sigma}(x_i),$$

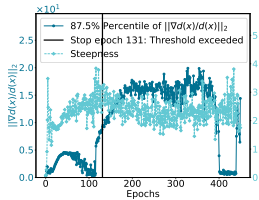
where $r = \|g_\theta(z) - x_i\|_2$, provided that the step size $s < (4\sigma - r)^2$ is sufficiently small.

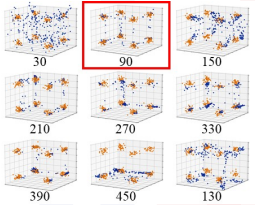
Input: An NSGAN model including a generator g_θ and a discriminator d_ω, thresholds $k_d > 0$ and $k_g < 0$, the number of modes $m \geq 1$, the number of warm-up iterations N_w.

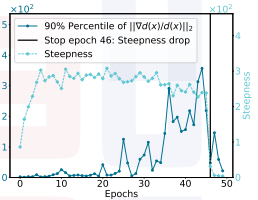
1: **for** number of training iterations **do**
2: Train the discriminator d_ω and the generator g_θ by the standard procedure.
3: Compute the $(1 - 1/m)$-th quantile of $\|\nabla d_\omega / d_\omega\|_2$ in a batch. Let the value be q.
4: Compute the mean steepness $\mathcal{S}_g^{\text{current}}$ across the batch and calculate the proportional drop compared to the previous iteration as $\Delta\mathcal{S}_g = (\mathcal{S}_g^{\text{current}} - \mathcal{S}_g^{\text{prev}})/\mathcal{S}_g^{\text{prev}}$.
5: **if** $(q > k_s \cdot \sqrt{2}/\sigma$ **or** $\Delta\mathcal{S}_g < k_g)$ **and** current iteration $> N_w$ **then**
6: **break**
7: **end if**
8: Update $\mathcal{S}_g^{\text{prev}} = \mathcal{S}_g^{\text{current}}$.
9: **end for**
10: **return** The best-performing model from earlier checkpoints.

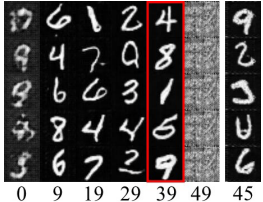
$\rightsquigarrow$ Inherent to GAN training, no need for extrinsic criteria such as the FID score.

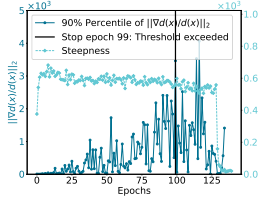
- Blue: Discriminator gradients $\|\nabla d(x)/d(x)\|_2$.
- Light blue: Steepness $\mathcal{S}_g$.

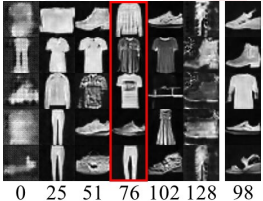


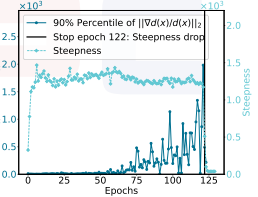


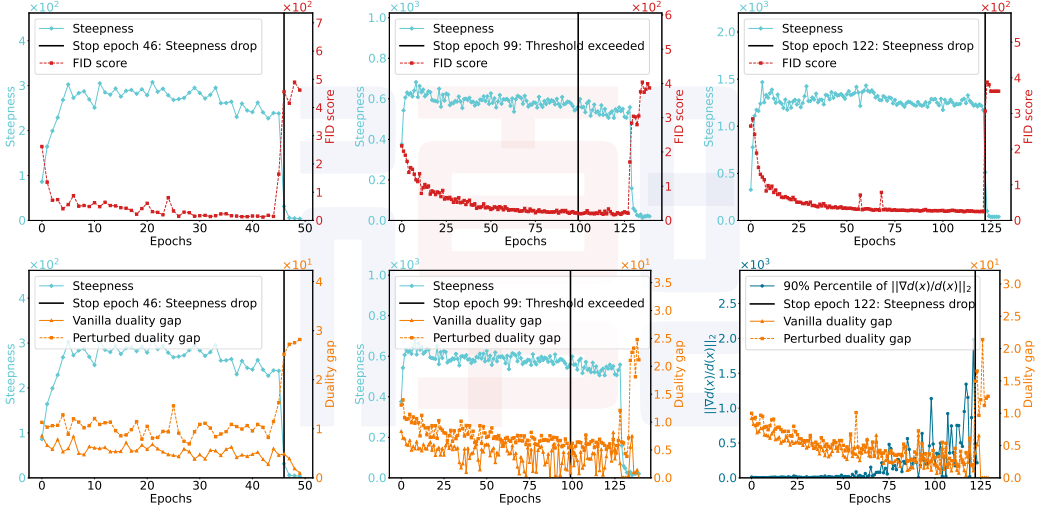








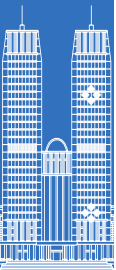


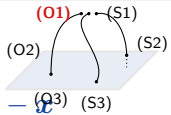


MNIST Fashion MNIST CIFAR-10

Outline

- Generative Models
  - 1.1 GANs and Flow-Based Models
- Algorithmic Mechanisms and Acceleration
  - 2.1 Diffusion Distillation
- Evolution of Discriminator and Generator Gradients in GAN Training
  - 3.1 What drives the particles to capture all the modes?
  - 3.2 Why does GAN collapse?
- Subspace Fidelity of Flow Matching Models
  - 4.1 How do flow matching models memorize?
  - 4.2 How do flow matching models generalize?
- Diffusion Trajectory Distillation
  - 5.1 Linear Gaussian Regime
  - 5.2 Nonlinear Gaussian Mixture Mechanism
- Conlcuding Remarks





The optimal velocity field for conditional flow matching

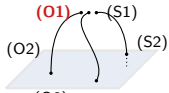
$$\text{(Optimal velocity field)} \quad \boldsymbol{v}_t^*(\boldsymbol{x}) = \mathop{\mathbb{E}}_{\boldsymbol{y}\sim p_1(\boldsymbol{y})}[\boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{y})|\boldsymbol{x},t], \text{ where } \boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{y}) = \frac{\boldsymbol{y}-\boldsymbol{x}}{1-t}.$$

Proposition (Optimal velocity field for discrete target distribution and OT paths)

Let $p_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_d)$, and suppose p_1 is a discrete distribution $p_1 \sim \left(\sum_{i=1}^N \delta_{\boldsymbol{y}^i}\right)/N$. Then

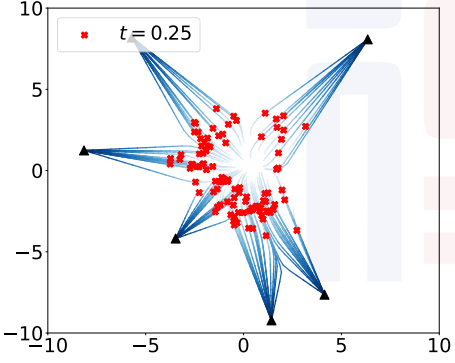
$$\boldsymbol{v}_t^*(\boldsymbol{x}) = \sum_{i=1}^N \frac{\boldsymbol{y}^i - \boldsymbol{x}}{1-t} \cdot \frac{\exp\left(-\|(\boldsymbol{x}-t\boldsymbol{y}^i)/(1-t)\|_2^2/2\right)}{\sum_{j=1}^N \exp\left(-\|(\boldsymbol{x}-t\boldsymbol{y}^j)/(1-t)\|_2^2/2\right)}.$$

- The optimal velocity field $\boldsymbol{v}_t^*(\boldsymbol{x})$ is a weighted sum of the vectors $(\boldsymbol{y}^i - \boldsymbol{x})/(1-t)$.
- The weights are given by a **softmax** function depending on $\|\boldsymbol{x} - t\boldsymbol{y}^i\|_2$'s and $1/(1-t)$.

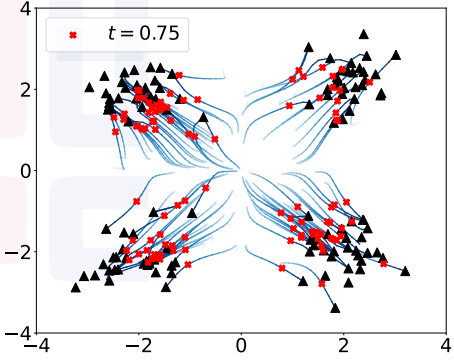


Recall: Flow ODE

$$\frac{\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x})}{\mathrm{d}t} = \sum_{i=1}^{N} \frac{\boldsymbol{y}^i - \boldsymbol{\phi}_t(\boldsymbol{x})}{1-t} \cdot \frac{\exp\left(-\|(\boldsymbol{\phi}_t(\boldsymbol{x}) - t\boldsymbol{y}^i)/(1-t)\|_2^2/2\right)}{\sum_{j=1}^{N} \exp\left(-\|(\boldsymbol{\phi}_t(\boldsymbol{x}) - t\boldsymbol{y}^j)/(1-t)\|_2^2/2\right)}, \quad \boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad t \in [0, 1].$$



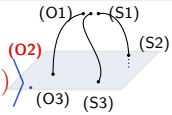
Sparse and well-separated dataset



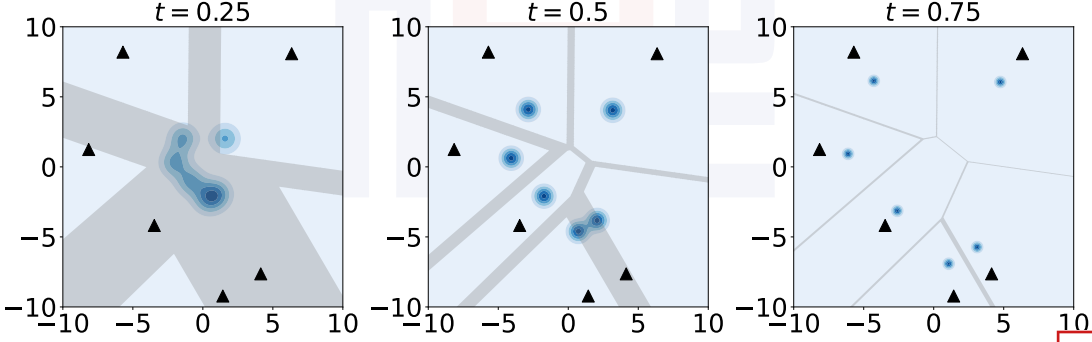
Abundant and hierarchical dataset

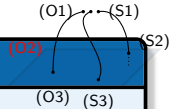
Recall: The optimal velocity field

$$\boldsymbol{v}_t^*(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{\boldsymbol{y}^i - \boldsymbol{x}}{1-t} \cdot \frac{\exp\left(-\|(\boldsymbol{x} - t\boldsymbol{y}^i)/(1-t)\|_2^2/2\right)}{\sum_{j=1}^{N} \exp\left(-\|(\boldsymbol{x} - t\boldsymbol{y}^j)/(1-t)\|_2^2/2\right)} := \left\langle \frac{\boldsymbol{y}^i - \boldsymbol{x}}{1-t}, \boldsymbol{w}_t(\boldsymbol{x}) \right\rangle.$$



- Straight generation path toward $\boldsymbol{y}^i$ $\iff$ Softmax weight $\boldsymbol{w}_t(\boldsymbol{x})$ close to a one-hot vector
- ▲: $\boldsymbol{y}^i$'s; Gray region: $\boldsymbol{w}_t(\boldsymbol{x})$ **not** close to a one-hot vector; Blue heatmap: density of p_t.



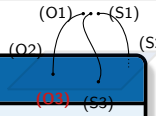


Theorem (Probability bound for softmax weight concentration)

Assume that $\|\boldsymbol{y}^i - \boldsymbol{y}^j\|_2 \geq M$ for all $1 \leq i \neq j \leq N$. Then the probability that $\boldsymbol{x}$ falls into the region $\mathcal{R} = \{\boldsymbol{x} \colon \max(\boldsymbol{w}_t(\boldsymbol{x})) \leq \tau < 1\}$, i.e., where the softmax weight vector is not close to a one-hot vector in the τ-sense, is bounded by

$$\mathbb{P}(\boldsymbol{x} \in \mathcal{R}) \leq \frac{1}{\sqrt{2\pi}} \cdot \frac{1-t}{t} \cdot \frac{1}{M} \cdot \log \frac{\tau(N-1)}{1-\tau} \cdot N(N-1).$$

- As $t \to 1$, the softmax weights concentrate on one component.
- Larger τ increases the bound logarithmically.
- Greater separation M lowers the bound inversely.
- More real samples N increase the bound quadratically.
- The bound is independent of d, applicable in high dimensions.



Theorem (Memorization phenomenon under the optimal velocity field)

Consider the ODE

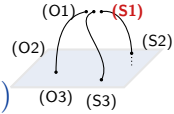
$$\frac{\mathrm{d}\boldsymbol{\phi}_t(\boldsymbol{x})}{\mathrm{d}t} = \boldsymbol{v}_t^*(\boldsymbol{\phi}_t(\boldsymbol{x})), \quad \boldsymbol{\phi}_0(\boldsymbol{x}) = \boldsymbol{x},$$

where

$$\boldsymbol{v}_t^*(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{\boldsymbol{y}^i - \boldsymbol{x}}{1-t} \cdot \frac{\exp\big(-\|(\boldsymbol{x} - t\boldsymbol{y}^i)/(1-t)\|_2^2/2\big)}{\sum_{j=1}^{N} \exp\big(-\|(\boldsymbol{x} - t\boldsymbol{y}^j)/(1-t)\|_2^2/2\big)}.$$

Then the set $\{\boldsymbol{\phi}_1(\boldsymbol{x}) \colon \boldsymbol{x} \in \mathbb{R}^d\}$ equals $\{\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^N\}$ up to a finite set.

- Can be proved using the variation of constants method.
- The phrase "up to a finite set" is essential. E.g., with two real data points, a noise sample starting at their midpoint stays stationary under $\boldsymbol{v}_t^*$.

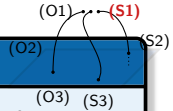


Recall: The optimal velocity field

$$\text{(Optimal velocity field)}\quad \boldsymbol{v}_t^*(\boldsymbol{x}) = \sum_{i=1}^{N} \frac{\boldsymbol{y}^i - \boldsymbol{x}}{1-t} \cdot \frac{\exp\left(-\|(\boldsymbol{x} - t\boldsymbol{y}^i)/(1-t)\|_2^2/2\right)}{\sum_{j=1}^{N} \exp\left(-\|(\boldsymbol{x} - t\boldsymbol{y}^j)/(1-t)\|_2^2/2\right)}.$$

Modeling the suboptimal velocity field $\hat{\boldsymbol{v}}_t(\boldsymbol{x})$:

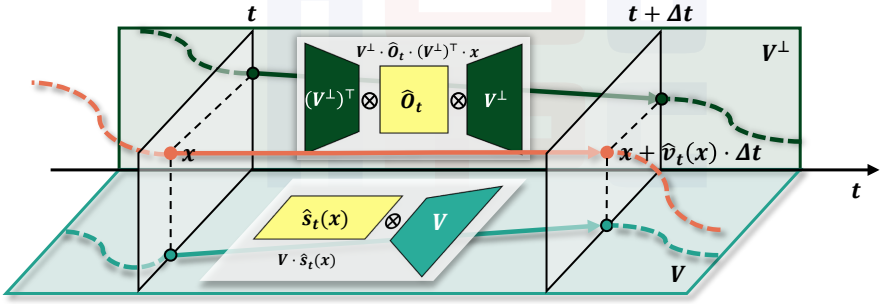
- Using a fixed network architecture (e.g., two-layer neural network, ReLU network, etc.).
  - May not fully capture the actual network used in practice.
  - Involves complex and potentially intractable analysis.
  - Treats spatial information $\boldsymbol{x}$ and temporal information t equally.
- Leaving the network architecture unspecified.
  - May fail to uncover the structure within $\boldsymbol{v}_t^*(\boldsymbol{x})$.
- **Partly using a predefined network architecture, while keeping other components flexible.**

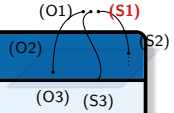


Definition (Orthogonal Subspace Decomposition Network, OSDNet)

Let $\boldsymbol{Y} = [\boldsymbol{y}^1, \boldsymbol{y}^2, \ldots, \boldsymbol{y}^N]$. Let $\boldsymbol{Y} = \boldsymbol{V}\boldsymbol{R}$, where $\boldsymbol{V}$ is orthonormal and spans the same space as range$(\boldsymbol{Y})$. The suboptimal velocity field, as defined by OSDNet, is given by

$$\hat{\boldsymbol{v}}_t\big(\boldsymbol{x}; \hat{\boldsymbol{O}}_t, \hat{\boldsymbol{s}}_t(\boldsymbol{x})\big) = \boldsymbol{V}^{\perp} \cdot \hat{\boldsymbol{O}}_t \cdot (\boldsymbol{V}^{\perp})^{\top} \cdot \boldsymbol{x} + \boldsymbol{V} \cdot \hat{\boldsymbol{s}}_t(\boldsymbol{x}).$$





Proposition (Optimal velocity field as a specific instance of OSDNet)

The optimal velocity field $\boldsymbol{v}_t^*(\boldsymbol{x})$ is a specific instance of the OSDNet, where

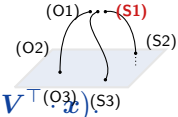
$$\hat{\boldsymbol{O}}_t^* = -\frac{1}{1-t}\cdot \boldsymbol{I}, \quad \hat{\boldsymbol{s}}_t^*(\boldsymbol{x}) = \frac{1}{1-t}\cdot\left(\boldsymbol{R}\cdot \boldsymbol{w}_t(\boldsymbol{x}) - \boldsymbol{V}^\top\cdot \boldsymbol{x}\right).$$

Using the variation of constants method and **time-ordered matrix exponentials**,

$$\boldsymbol{\phi}_t(\boldsymbol{x}) = \mathcal{T}\exp\left(\int_0^t \boldsymbol{V}^\perp\cdot\hat{\boldsymbol{O}}_s\cdot(\boldsymbol{V}^\perp)^\top \mathrm{d}s\right) \cdot\left(\boldsymbol{x} + \int_0^t \mathcal{T}\exp\left(\int_0^s \boldsymbol{V}^\perp\cdot\hat{\boldsymbol{O}}_\tau\cdot(\boldsymbol{V}^\perp)^\top \mathrm{d}\tau\right)^{-1}\cdot\boldsymbol{V}\cdot\hat{\boldsymbol{s}}_s(\boldsymbol{\phi}_s(\boldsymbol{x}))\mathrm{d}s\right).$$

Assuming the $\hat{\boldsymbol{O}}_t$'s are diagonal,

$$\boldsymbol{\phi}_1(\boldsymbol{x}) = \underbrace{\boldsymbol{V}\cdot\left(\boldsymbol{V}^\top\cdot\boldsymbol{x} + \int_0^1 \hat{\boldsymbol{s}}_s(\boldsymbol{\phi}_s(\boldsymbol{x}))\mathrm{d}s\right)}_{\text{within the span of } \boldsymbol{V}} + \underbrace{\boldsymbol{V}^\perp\cdot\exp\left(\int_0^1 \hat{\boldsymbol{O}}_s\mathrm{d}s\right)\cdot(\boldsymbol{V}^\perp)^\top\cdot\boldsymbol{x}}_{\text{orthogonal to the span of } \boldsymbol{V}}.$$



Recall: The optimal counterpart of OSDNet

$$\text{(Optimal } \hat{\boldsymbol{O}}_t\text{)} \quad \hat{\boldsymbol{O}}_t^* = -\frac{1}{1-t}\cdot \boldsymbol{I}; \quad \text{(Optimal } \hat{\boldsymbol{s}}_t(\boldsymbol{x})\text{)} \quad \hat{\boldsymbol{s}}_t^*(\boldsymbol{x}) = \frac{1}{1-t}\cdot\left(\boldsymbol{R}\cdot\boldsymbol{w}_t(\boldsymbol{x}) - \boldsymbol{V}^\top\cdot\boldsymbol{x}\right)$$

Proposition (Teacher-student training of OSDNet)

Minimizing the CFM loss

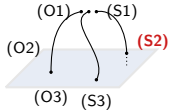
$$\mathcal{L}_{\text{CFM}}\left(\hat{\boldsymbol{O}}_t, \hat{\boldsymbol{s}}_t(\boldsymbol{x})\right) = \mathbb{E}_{t\sim\mathcal{U}[0,1],\boldsymbol{y}\sim p_1(\boldsymbol{y}),\boldsymbol{x}\sim p_t(\boldsymbol{x}|\boldsymbol{y})}\left[\left\|\hat{\boldsymbol{v}}_t\left(\boldsymbol{x};\hat{\boldsymbol{O}}_t,\hat{\boldsymbol{s}}_t(\boldsymbol{x})\right) - \boldsymbol{u}_t(\boldsymbol{x}|\boldsymbol{y})\right\|_2^2\right],$$

is equivalent to solving the two independent minimization problems below:

$$\min_{\hat{\boldsymbol{O}}_t} \mathbb{E}_{t\sim\mathcal{U}[0,1],\boldsymbol{x}\sim p_t(\boldsymbol{x})}\left[\left\|\boldsymbol{V}^\perp\cdot\left(\hat{\boldsymbol{O}}_t + \frac{1}{1-t}\cdot\boldsymbol{I}\right)\cdot(\boldsymbol{V}^\perp)^\top\cdot\boldsymbol{x}\right\|_2^2\right]$$

and

$$\min_{\hat{\boldsymbol{s}}_t(\boldsymbol{x})} \mathbb{E}_{t\sim\mathcal{U}[0,1],\boldsymbol{x}\sim p_t(\boldsymbol{x})}\left[\left\|\boldsymbol{V}\cdot\left(\hat{\boldsymbol{s}}_t(\boldsymbol{x}) - \frac{1}{1-t}\cdot(\boldsymbol{R}\cdot\boldsymbol{w}_t(\boldsymbol{x}) - \boldsymbol{V}^\top\cdot\boldsymbol{x})\right)\right\|_2^2\right].$$



① Transforming the objective function into

$$\int_0^1 \left\|(1-t)\cdot\hat{\boldsymbol{O}}_t+\boldsymbol{I}\right\|_{\mathrm{F}}^2\mathrm{d}t=\sum_{\hat{o}_t\in\mathrm{diag}(\hat{\boldsymbol{O}}_t)}\int_0^1\left((1-t)\hat{o}_t+1\right)^2\mathrm{d}t.$$

② Let the sinusoidal positional encoding be

$$\mathrm{emb}(t)=\left(\sin\left(\frac{s\cdot t}{\ell^{\frac{0}{\mathrm{dim}}}}\right),\cos\left(\frac{s\cdot t}{\ell^{\frac{0}{\mathrm{dim}}}}\right),\sin\left(\frac{s\cdot t}{\ell^{\frac{2}{\mathrm{dim}}}}\right),\cos\left(\frac{s\cdot t}{\ell^{\frac{2}{\mathrm{dim}}}}\right),\ldots,\sin\left(\frac{s\cdot t}{\ell^{\frac{\mathrm{dim}-2}{\mathrm{dim}}}}\right),\cos\left(\frac{s\cdot t}{\ell^{\frac{\mathrm{dim}-2}{\mathrm{dim}}}}\right)\right)^\top,$$

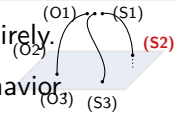
where s is the scaling factor, ℓ is the wavelength and dim is the embedding dimension. Assume that $\hat{o}_t=\boldsymbol{\kappa}^\top\mathrm{emb}(t)$. Optimize $\boldsymbol{\kappa}$ with respect to the objective

$$\int_0^1\left((1-t)\hat{o}_t+1\right)^2\mathrm{d}t=\int_0^1\left((1-t)\boldsymbol{\kappa}^\top\mathrm{emb}(t)+1\right)^2\mathrm{d}t:=\boldsymbol{\kappa}^\top\boldsymbol{A}\boldsymbol{\kappa}+2\boldsymbol{\kappa}^\top\boldsymbol{b}+1,$$

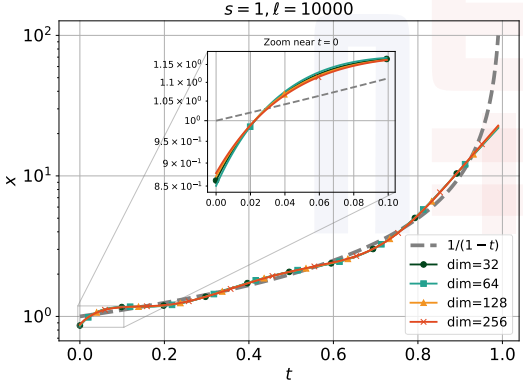
where $\boldsymbol{A}=\int_0^1(1-t)^2\,\mathrm{emb}(t)\,\mathrm{emb}(t)^\top\mathrm{d}t$ and $\boldsymbol{b}=\int_0^1(1-t)\,\mathrm{emb}(t)\mathrm{d}t$.

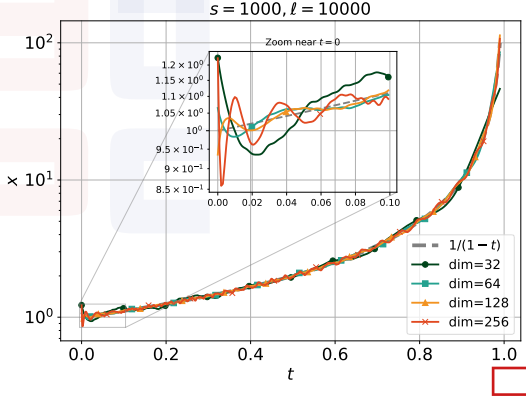
③ The off-subspace component of $\boldsymbol{\phi}_1(\boldsymbol{x})$ is

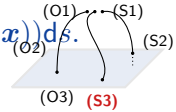
$$\exp\left(-\boldsymbol{b}^\top\boldsymbol{A}^{-\top}\boldsymbol{e}\right)\cdot\boldsymbol{V}^\perp\cdot(\boldsymbol{V}^\perp)^\top\cdot\boldsymbol{x},\quad\boldsymbol{e}=\int_0^1\mathrm{emb}(t)\mathrm{d}t.$$



- The contribution of the off-space component persists and does not vanish entirely.
- When scaling factor s is small, different dim exhibit similar approximation behavior.
- When scaling factor s is large, increasing dim yields an oscillatory yet better approximation.







① Assume $\boldsymbol{x} \in \text{range}(\boldsymbol{V})$. The subspace component becomes $\boldsymbol{x} + \int_0^1 \boldsymbol{V} \cdot \hat{\boldsymbol{s}}_s(\boldsymbol{\phi}_s(\boldsymbol{x}))\mathrm{d}s$.

② Reformulate it as the solution of

$$\mathrm{d}\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}) = \boldsymbol{V} \cdot \hat{\boldsymbol{s}}_t(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}))\,\mathrm{d}t, \quad \hat{\boldsymbol{\phi}}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad t \in [0,1].$$

③ Add noise to the ODE to account for perturbations:

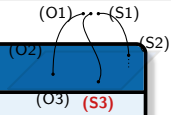
$$\mathrm{d}\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}) = \boldsymbol{V} \cdot \hat{\boldsymbol{s}}_t(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}))\,\mathrm{d}t + \sigma\,\mathrm{d}\boldsymbol{W}_t, \quad \hat{\boldsymbol{\phi}}_0(\boldsymbol{x}) = \boldsymbol{x}, \quad t \in [0,1].$$

Theorem (Generalization of subspace components, informal)

The difference between $\boldsymbol{\phi}^*_{1-\varepsilon}(\boldsymbol{x})$ and $\hat{\boldsymbol{\phi}}_{1-\varepsilon}(\boldsymbol{x})$ is given by

$$\boldsymbol{\phi}^*_{1-\varepsilon}(\boldsymbol{x}) - \hat{\boldsymbol{\phi}}_{1-\varepsilon}(\boldsymbol{x}) = \int_0^{1-\varepsilon} \nabla_{\boldsymbol{x}} \boldsymbol{\phi}^*_{t,1-\varepsilon}(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x})) \cdot \boldsymbol{V} \cdot \big(\boldsymbol{s}^*_t(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x})) - \hat{\boldsymbol{s}}_t(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}))\big)\mathrm{d}t$$

$$- \sigma \cdot \int_0^{1-\varepsilon} \nabla_{\boldsymbol{x}} \boldsymbol{\phi}^*_{t,1-\varepsilon}(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}))\mathrm{d}\boldsymbol{W}_t - \frac{\sigma^2}{2} \cdot \int_0^{1-\varepsilon} \sum_{i,j=1}^{d} \big(\nabla^2_{\boldsymbol{x}} \boldsymbol{\phi}^*_{t,1-\varepsilon}(\hat{\boldsymbol{\phi}}_t(\boldsymbol{x}))\big)_{i,j,:}\,\mathrm{d}t.$$



Theorem (Error bound of subspace components, informal)

Specifically, for $\sigma = 0$, we have

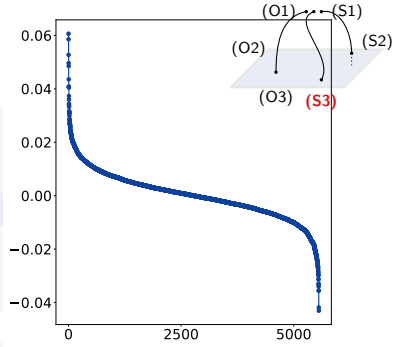
$$\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_d)}[\|\boldsymbol{\phi}^*_{1-\varepsilon}(\boldsymbol{x}) - \hat{\boldsymbol{\phi}}_{1-\varepsilon}(\boldsymbol{x})\|_2^2] \le C \cdot \mathop{\mathbb{E}}_{t\sim\mathcal{U}(0,1),\boldsymbol{x}\sim p_t}[\|\boldsymbol{s}^*_t(\boldsymbol{x}) - \hat{\boldsymbol{s}}_t(\boldsymbol{x})\|_2^2],$$

where p_t is the distribution of $\boldsymbol{\phi}^*_t(\boldsymbol{x})$.

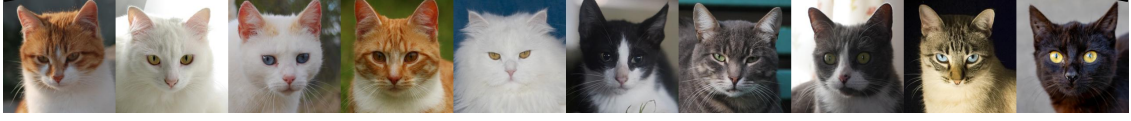
- The difference between generated and real samples consists of three parts:
  - The gap between optimal and suboptimal vector fields;
  - The noise from the stochastic process;
  - The squared noise term error.
- The proximity of generated samples to $\boldsymbol{y}^i$ positively correlates with how well $\hat{\boldsymbol{s}}_t(\boldsymbol{x})$ is trained.
  - **Flow matching models generalize when they cannot memorize**.

Generated image

Projected image



Projection coefficients



Dataset images of top 10 coefficients

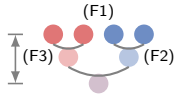
Outline

- Generative Models
  - 1.1 GANs and Flow-Based Models
- Algorithmic Mechanisms and Acceleration
  - 2.1 Diffusion Distillation
- Evolution of Discriminator and Generator Gradients in GAN Training
  - 3.1 What drives the particles to capture all the modes?
  - 3.2 Why does GAN collapse?
- Subspace Fidelity of Flow Matching Models
  - 4.1 How do flow matching models memorize?
  - 4.2 How do flow matching models generalize?

Diffusion Trajectory Distillation

- 5.1 Linear Gaussian Regime
- 5.2 Nonlinear Gaussian Mixture Mechanism

Conlcuding Remarks



(F1) Derivation of the teacher operators

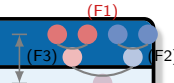
$\mathcal{T}_T(\boldsymbol{z}_T)$ A_1 $\times$ A_2 $\cdots$ A_{T-1} $\times$ A_T $\boldsymbol{z}_T$

(F3) Evaluation of distillation strategy

(F2) Operator shrinkage in merging

$A^{\mathrm{st}}_{1:2}$ $\cdots$ $A^{\mathrm{st}}_{T-1:T}$

$\boldsymbol{A}_{1:T}(\boldsymbol{z}_T)$ $A^{\mathrm{st}}_{1:T}$ $\boldsymbol{z}_T$



Assumption (Real data distribution is Gaussian)

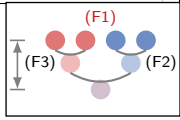
Assume that the real data distribution p_0 is a d-dimensional centered Gaussian distribution with diagonal covariance matrix, i.e., $p_0 = \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, where $\mathbf{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_d)$, and the eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$.

- Real data: $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{\Lambda})$, diagonal $\mathbf{\Lambda}$;
- Forward process: $\boldsymbol{z}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;
- Optimal denoising estimator: $\hat{\boldsymbol{x}}_0^\star(\boldsymbol{z}_t, t) = \mathbb{E}[\boldsymbol{x}_0 | \boldsymbol{z}_t] = \alpha_t \mathbf{\Lambda}(\alpha_t^2 \mathbf{\Lambda} + \sigma_t^2 \boldsymbol{I})^{-1} \boldsymbol{z}_t$;
- Sampling step: $\boldsymbol{z}_{t-1} = \alpha_{t-1} \hat{\boldsymbol{x}}_0^\star(\boldsymbol{z}_t, t) + \sigma_{t-1} \cdot \left(\frac{\boldsymbol{z}_t - \alpha_t \hat{\boldsymbol{x}}_0^\star(\boldsymbol{z}_t, t)}{\sigma_t} \right)$.

Why this assumption?

Implicit bias: Under limited capacity, diffusion models learn the Gaussian structure in real data [Li et al. 2024c].

Analytical tractability: A diagonal Gaussian makes each DDIM step a coordinate-wise linear map [Li et al. 2024c; Wang et al. 2023; Wang & Vastola 2023b].



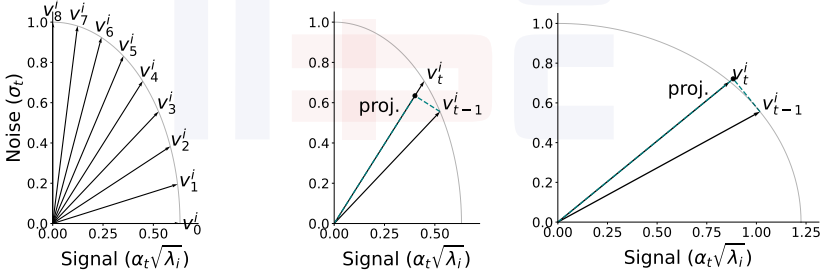
Recall: Teacher sampling step under Gaussian data is

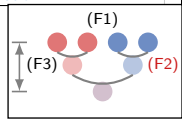
$$\boldsymbol{z}_{t-1} = \underbrace{(\alpha_{t-1}\alpha_t \boldsymbol{\Lambda} + \sigma_{t-1}\sigma_t \boldsymbol{I})(\alpha_t^2 \boldsymbol{\Lambda} + \sigma_t^2 \boldsymbol{I})^{-1}}_{\text{linear operator}} \cdot \boldsymbol{z}_t.$$

Observation: The operator is diagonal $\rightsquigarrow$ each coordinate evolves independently.

Notation: Define signal-noise vector $\boldsymbol{v}_t^i := (\alpha_t\sqrt{\lambda_i}, \sigma_t)^\top \in \mathbb{R}^2$, then (we will revisit this later)

$$(\boldsymbol{z}_{t-1})_i = \frac{\langle \boldsymbol{v}_{t-1}^i, \boldsymbol{v}_t^i \rangle}{\|\boldsymbol{v}_t^i\|_2^2} \cdot (\boldsymbol{z}_t)_i, \quad i = 1, \ldots, d.$$





Student model: A diagonal linear operator at each step

$$\boldsymbol{z}_t \mapsto \boldsymbol{A}_t^{\mathrm{st}} \boldsymbol{z}_t, \quad \text{where } \boldsymbol{A}_t^{\mathrm{st}} = \mathrm{Diag}(a_t^1, \ldots, a_t^d),$$

initialized from the teacher model's closed-form DDIM step: $\boldsymbol{z}_t \mapsto \boldsymbol{A}_t^\star \boldsymbol{z}_t$.

Training objective: Match teacher model's k-step update in one step,

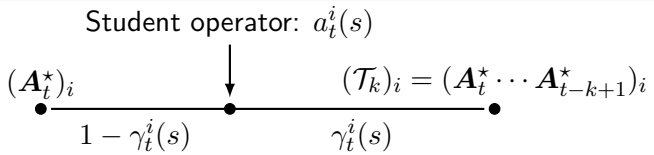
$$\mathcal{L}_t(\boldsymbol{A}_t^{\mathrm{st}}) = \mathbb{E}_{\boldsymbol{z}_t \sim p_t}\left[\|\boldsymbol{A}_t^{\mathrm{st}} \boldsymbol{z}_t - \mathcal{T}_k(\boldsymbol{z}_t)\|_2^2\right].$$

Proposition (Student operator interpolates between teacher and target)

Under the gradient flow ODE, the student update $a_t^i(s)$ evolves as

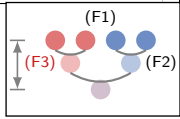
$$a_t^i(s) = (1 - \gamma_t^i(s)) \cdot (\mathcal{T}_k)_i + \gamma_t^i(s) \cdot (\boldsymbol{A}_t^\star)_i,$$

where the interpolation weight $\gamma_t^i(s) = \exp(-2s\|\boldsymbol{v}_t^i\|_2^2)$, with $\boldsymbol{v}_t^i = (\alpha_t \sqrt{\lambda_i}, \sigma_t)^\top \in \mathbb{R}^2$.



How to evaluate a distillation strategy?

By comparing the operator it learns against the teacher operator?
— A surrogate target is better!



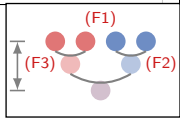
Proposition (Teacher model contracts the covariance)

Each output coordinate satisfies $|(\mathcal{T}_T(\boldsymbol{z}_T))_i| \le \sqrt{\lambda_i} \cdot |(\boldsymbol{z}_T)_i|$, with equality if and only if $(\boldsymbol{z}_T)_i = 0$. Consequently, if $\boldsymbol{z}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, then the teacher model produces samples with a contracted covariance matrix, i.e., $\mathrm{Cov}\big(\mathcal{T}_T(\boldsymbol{z}_T)\big) \preceq \boldsymbol{\Lambda}$, where the inequality is strict if $\boldsymbol{\Lambda} \succ \boldsymbol{0}$.

Definition (Surrogate composite operator)

For each $i \in \{1, \dots, d\}$, define the surrogate composite operator coordinate-wise as:

$$(\widetilde{\mathcal{T}}_T)_i = \prod_{t=1}^{T} (\widetilde{\boldsymbol{A}}_t^\star)_i, \quad \text{where} \quad (\widetilde{\boldsymbol{A}}_t^\star)_i := \begin{cases} (\boldsymbol{A}_t^\star)_i, & \text{if } \lambda_i \le 1 \text{ or } (\boldsymbol{A}_t^\star)_i \ge 1, \\ 1, & \text{if } \lambda_i > 1 \text{ and } (\boldsymbol{A}_t^\star)_i < 1. \end{cases}$$



Initialization ($k = 0$): Start from teacher operator:

$$(\boldsymbol{A}_t^{(0)})_i = (\boldsymbol{A}_t^{\star})_i = \frac{\langle \boldsymbol{v}_{t-1}^i, \boldsymbol{v}_t^i \rangle}{\|\boldsymbol{v}_t^i\|_2^2}, \quad t = 1, \ldots, T.$$

The signal-noise vector is defined per-coordinate as

$$\boldsymbol{v}_t^i := (\alpha_t \sqrt{\lambda_i}, \sigma_t)^\top \in \mathbb{R}^2.$$

Operator merging ($k \geq 1$): Replace $\boldsymbol{A}_{t_1}^{(k-1)}, \ldots, \boldsymbol{A}_{t_2}^{(k-1)}$ with

$$(\boldsymbol{A}_{t_1:t_2}^{(k)})_i = (1 - \gamma_{t_2}^i) \cdot \prod_{t=t_1}^{t_2} (\boldsymbol{A}_t^{(k-1)})_i + \gamma_{t_2}^i \cdot (\boldsymbol{A}_{t_2}^{(k-1)})_i.$$

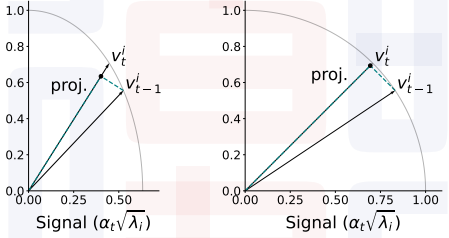
The shrinkage factor is defined per-coordinate as

$$\gamma_{t_2}^i = \exp(-2s\|\boldsymbol{v}_{t_2}^i\|_2^2), \quad s > 0.$$

Repeat: Continue merging to construct final trajectory $\boldsymbol{A}_{1:T}^{(K)}$.

Proposition (Teacher operators $(\boldsymbol{A}_t^\star)_i \leq 1$)

When $\lambda_i \leq 1$, we have $(\boldsymbol{A}_t^\star)_i = (\alpha_{t-1}\alpha_t\lambda_i + \sigma_{t-1}\sigma_t)/(\alpha_t^2\lambda_i + \sigma_t^2) \leq 1$.

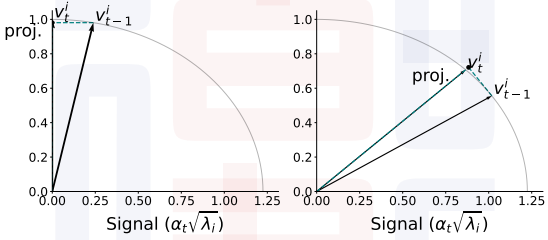


Theorem (Sequential BOOT is optimal when $\lambda \leq 1$)

Assume that $\boldsymbol{\Lambda} = \lambda \leq 1$. Then the optimal merge strategy that minimizes the squared error to the surrogate target $\prod_{t=1}^{T} A_t^\star$ (which coincides with the original composite operator when $\lambda \leq 1$) is the sequential BOOT distillation.

Proposition (Teacher operators $(\boldsymbol{A}_t^\star)_i > 1$ for $t \neq T$)

When $\lambda_i \gg 1$, we have $A_T^\star < 1$ and $A_1^\star, A_2^\star, \ldots, A_{T-1}^\star > 1$.



Theorem (Vanilla distillation is optimal when $\lambda \gg 1$)

Assume that $\boldsymbol{\Lambda} = \lambda \gg 1$. Then the optimal merge strategy that minimizes the squared error to the surrogate target $\prod_{t=1}^{T} \widetilde{A}_t^\star$ is the vanilla trajectory distillation.

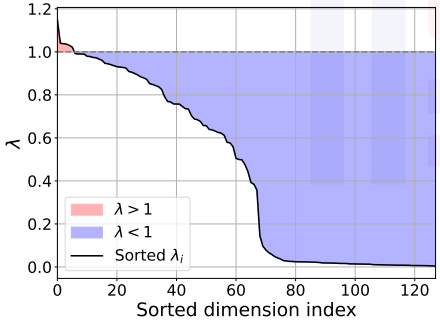
Settings:

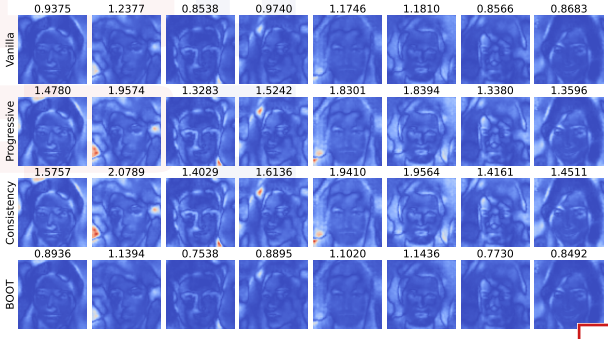
- Real data distribution $p_0 = \mathcal{N}(0, \lambda)$; teacher steps $T = 512$; student training time $s = 6.4$.
- Reported metric: signed error, defined as $\mathbb{E}[\text{Student output} - \text{Surrogate teacher output}]$.
- Underlined entries indicate the lowest error in each row.
- Results largely match theory: BOOT performs best when $\lambda \leq 1$; vanilla is optimal when $\lambda \gg 1$.

λ	Vanilla	Progressive	BOOT	Consistency
0.20	7.0E-4 $\pm$ 1.6E-4	7.9E-2 $\pm$ 4.9E-4	5.0E-4 $\pm$ 1.0E-4	4.0E-1 $\pm$ 1.3E-2
0.50	4.3E-4 $\pm$ 8.7E-5	1.0E-2 $\pm$ 9.9E-5	3.4E-4 $\pm$ 6.0E-5	1.7E-1 $\pm$ 1.4E-2
1.00	3.0E-6 $\pm$ 4.4E-7	1.5E-5 $\pm$ 3.2E-7	3.4E-6 $\pm$ 6.2E-7	7.1E-4 $\pm$ 1.1E-4
1.02	-1.3E-4 $\pm$ 2.4E-6	-1.6E-4 $\pm$ 1.2E-6	-1.3E-4 $\pm$ 1.7E-6	-2.2E-3 $\pm$ 2.9E-4
2.00	-5.9E-4 $\pm$ 8.5E-5	-1.1E-3 $\pm$ 7.1E-5	-6.7E-4 $\pm$ 1.1E-4	-5.2E-2 $\pm$ 6.2E-3
5.00	-1.8E-3 $\pm$ 2.8E-4	-2.9E-3 $\pm$ 3.7E-4	-2.4E-3 $\pm$ 6.4E-4	-9.2E-2 $\pm$ 1.3E-2

Settings:

- Train a Multiscale Structural Similarity VAE (MSSIMVAE) [Snell et al. 2017] on the CelebA [Liu et al. 2015] dataset.
- The latent space is approximately diagonal Gaussian with most eigenvalues smaller than 1.
- Empirical results align with theory: BOOT achieves the lowest error.





Goal: Find a merging strategy $\mathcal{S}$ that produces a final merged student operator $\boldsymbol{A}_{1:T}(\mathcal{S})$ minimizing

$$\mathcal{L}_{W_2}(\mathcal{S}) = \sum_{i=1}^{d} \left((\widetilde{\mathcal{T}}_T)_i - (\boldsymbol{A}_{1:T}(\mathcal{S}))_i \right)^2.$$

Brute-force complexity: A merge plan is built recursively, either by a single direct merge on the whole interval, or by splitting at some $1 \le k \le T-1$ and independently merging the left and right subintervals. Let N_T be the number of valid merge plans on T steps. Then

$$N_T = \Theta\left(\frac{5^T}{T^{3/2}} \right).$$

Pareto dominance: Define the preference direction

$$\text{pref}_i := \begin{cases} \text{higher is better}, & \lambda_i > 1, \\ \text{lower is better}, & \lambda_i \le 1. \end{cases}$$

For two candidates $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^d$, say $\boldsymbol{A}$ dominates $\boldsymbol{B}$ if $\boldsymbol{A}$ is at least as good as $\boldsymbol{B}$ in every coordinate under pref_i, and it is strictly better in at least one coordinate.

DP table: For every interval $[t_1, t_2]$, $\mathrm{dp}[t_1][t_2]$ stores a Pareto frontier of candidate merged operators on that interval.

Initialization: For a single step, no merge is needed

$$\mathrm{dp}[t][t] = \{\boldsymbol{A}_t^\star\}.$$

Recurrence: Fix $[t_1, t_2]$ with $t_1 < t_2$. Build a candidate pool $\mathcal{C}_{t_1:t_2}$ from

- Direct merge with shrinkage applied at time t_2

$$(\boldsymbol{A}^{\mathsf{direct}}_{t_1:t_2})_i = (1-\gamma^i_{t_2}) \prod_{t=t_1}^{t_2} (\boldsymbol{A}^\star_t)_i + \gamma^i_{t_2} (\boldsymbol{A}^\star_{t_2})_i.$$

- All binary splits at $m \in \{t_1, \ldots, t_2 - 1\}$. For every $\boldsymbol{L} \in \mathrm{dp}[t_1][m]$ and $\boldsymbol{R} \in \mathrm{dp}[m+1][t_2]$,

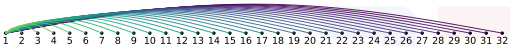
$$(\boldsymbol{A}^{\mathsf{split}}_{t_1:t_2})_i = (1-\gamma^i_{t_2})(\boldsymbol{L})_i (\boldsymbol{R})_i + \gamma^i_{t_2} (\boldsymbol{R})_i.$$

Complexity: Brute force is exponential in T. The DP reuses interval solutions and avoids enumerating all merge trees. With Pareto frontiers, runtime depends on the typical frontier size

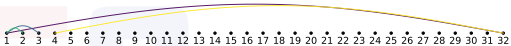
$$\mathcal{O}\big(T^3 \cdot d \cdot \#\{\text{Typical Pareto frontier}\}\big) \ll \Theta\left(\frac{5^T}{T^{3/2}}\right).$$

Observation: Hybrid and intricate merging strategies in the transitional regime ($\Lambda \approx I$).

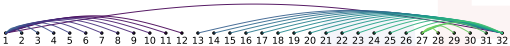
Λ = [0.95, 1.05]



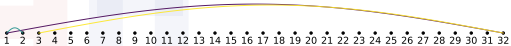
Λ = [0.95, 1.25]



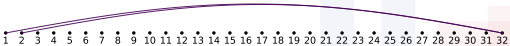
Λ = [0.97, 1.10]



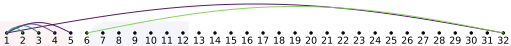
Λ = [0.98, 1.30]



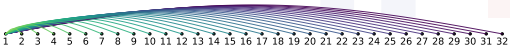
Λ = [0.99, 1.40]



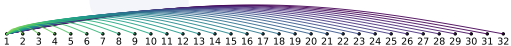
Λ = [1.00, 1.20]



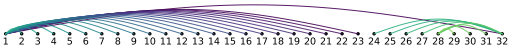
Λ = [1.02, 1.35]



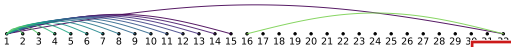
Λ = [1.02, 1.80]

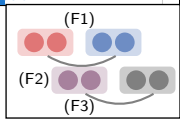


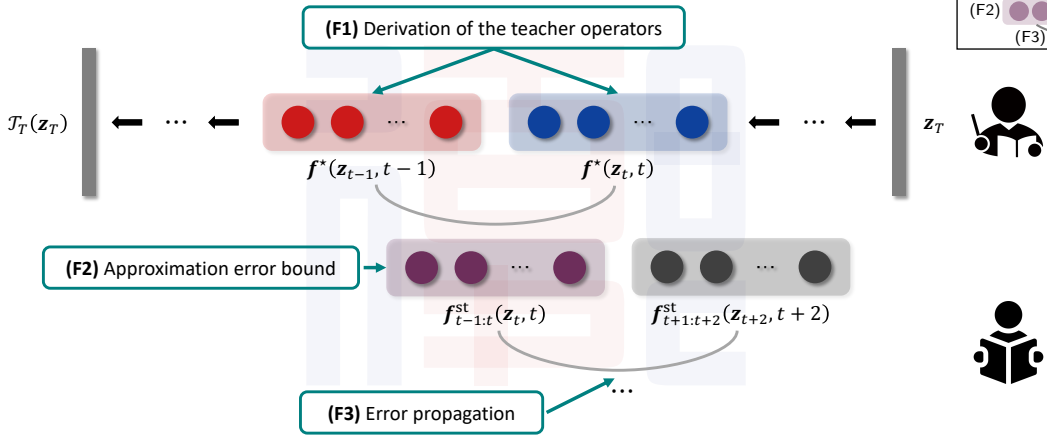
Λ = [1.05, 1.60]

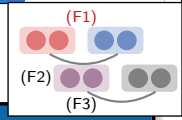


Λ = [1.08, 1.60]




(F1)
(F2)
(F3)


(F1) Derivation of the teacher operators
$\mathcal{T}_T(\boldsymbol{z}_T)$
$\boldsymbol{z}_T$
$\boldsymbol{f}^\star(\boldsymbol{z}_{t-1}, t-1)$
$\boldsymbol{f}^\star(\boldsymbol{z}_t, t)$
(F2) Approximation error bound
$\boldsymbol{f}^{\mathrm{st}}_{t-1:t}(\boldsymbol{z}_t, t)$
$\boldsymbol{f}^{\mathrm{st}}_{t+1:t+2}(\boldsymbol{z}_{t+2}, t+2)$
(F3) Error propagation



Real data: $\boldsymbol{x}_0 \sim \sum_{k=1}^{K} \pi_k \mathcal{N}\big(\boldsymbol{x}_0; \boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k\big)$, where $\pi_k > 0$ and $\sum_{k=1}^{K} \pi_k = 1$.

Proposition (Teacher sampling step is a K-component affine MoE)

Under the assumption above, each sampling step is a K-component affine MoE operator:

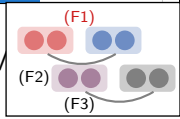
$$\boldsymbol{z}_{t-1} = \boldsymbol{f}^\star(\boldsymbol{z}_t, t) := \sum_{k=1}^{K} \gamma_{k,t}^\star(\boldsymbol{z}_t)\big(\boldsymbol{A}_{t,k}^\star \boldsymbol{z}_t + \boldsymbol{b}_{k,t}^\star\big),$$

where, for each component $1 \le k \le K$,

$$\gamma_{k,t}(\boldsymbol{z}_t) := \frac{\pi_k \mathcal{N}\big(\boldsymbol{z}_t; \alpha_t \boldsymbol{\mu}_k, \alpha_t^2 \boldsymbol{\Lambda}_k + \sigma_t^2 \boldsymbol{I}\big)}{\sum_{\ell=1}^{K} \pi_\ell \mathcal{N}\big(\boldsymbol{z}_t; \alpha_t \boldsymbol{\mu}_\ell, \alpha_t^2 \boldsymbol{\Lambda}_\ell + \sigma_t^2 \boldsymbol{I}\big)},$$

$$\boldsymbol{A}_{t,k}^\star := (\alpha_{t-1}\alpha_t \boldsymbol{\Lambda}_k + \sigma_{t-1}\sigma_t \boldsymbol{I})(\alpha_t^2 \boldsymbol{\Lambda}_k + \sigma_t^2 \boldsymbol{I})^{-1},$$

$$\boldsymbol{b}_{k,t}^\star := (\alpha_{t-1}\boldsymbol{I} - \alpha_t \boldsymbol{A}_{t,k})\boldsymbol{\mu}_k.$$



Parameterization: We parametrize the student model as a K-component affine M

$$\boldsymbol{f}^{\mathrm{st}}(\boldsymbol{z}_t, t) = \sum_{k=1}^{K} \gamma_{k,t}^{\mathrm{st}}(\boldsymbol{z}_t)\big(\boldsymbol{A}_{k,t}^{\mathrm{st}}\boldsymbol{z}_t + \boldsymbol{b}_{k,t}^{\mathrm{st}}\big), \quad \sum_{k=1}^{K} \gamma_{k,t}^{\mathrm{st}}(\boldsymbol{z}_t) = 1.$$

Lemma (Two-step teacher composition yields K^2 components)

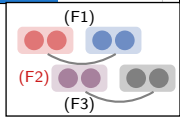
The two-step composite teacher operator $\mathcal{T}_2(\boldsymbol{z}_t) = \boldsymbol{f}^{\star}(\boldsymbol{f}^{\star}(\boldsymbol{z}_t, t), t-1)$ can be expanded as

$$\mathcal{T}_2(\boldsymbol{z}_t) = \sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\big(\boldsymbol{A}_{ij,t}^{(2)}\boldsymbol{z}_t + \boldsymbol{b}_{ij,t}^{(2)}\big),$$

where

$$w_{ij,t}^{(2)}(\boldsymbol{z}_t) = \gamma_{i,t}^{\star}(\boldsymbol{z}_t)\gamma_{j,t-1}^{\star}\big(\boldsymbol{f}^{\star}(\boldsymbol{z}_t, t)\big), \quad \boldsymbol{A}_{ij,t}^{(2)} = \boldsymbol{A}_{j,t-1}^{\star}\boldsymbol{A}_{i,t}^{\star}, \quad \boldsymbol{b}_{ij,t}^{(2)} = \boldsymbol{A}_{j,t-1}^{\star}\boldsymbol{b}_{i,t}^{\star} + \boldsymbol{b}_{j,t-1}^{\star}.$$

Complexity growth: Composition increases components $K \to K^2$ (and $K \to K^k$ for k steps).



Recall: The student operator is a K-component affine MoE

$$\boldsymbol{f}^{\mathsf{st}}(\boldsymbol{z}_t, t) = \sum_{k=1}^{K} \gamma_{k,t}^{\mathsf{st}}(\boldsymbol{z}_t)\big(\boldsymbol{A}_{k,t}^{\mathsf{st}}\boldsymbol{z}_t + \boldsymbol{b}_{k,t}^{\mathsf{st}}\big).$$

And the two-step composite teacher operator is a K^2-component affine MoE

$$\mathcal{T}_2(\boldsymbol{z}_t) = \sum_{i=1}^{K}\sum_{j=1}^{K} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\boldsymbol{g}_{ij,t}(\boldsymbol{z}_t), \quad \boldsymbol{g}_{ij,t}(\boldsymbol{z}_t) = \boldsymbol{A}_{ij,t}^{(2)}\boldsymbol{z}_t + \boldsymbol{b}_{ij,t}^{(2)}.$$

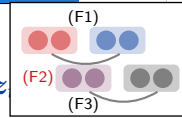
Theorem (Two-step approximation error upper bound)

Define the best-achievable K-component student approximation error

$$\varepsilon_{t-1:t} := \inf_{\gamma_{k,t}^{\mathsf{st}}, \boldsymbol{A}_{k,t}^{\mathsf{st}}, \boldsymbol{b}_{k,t}^{\mathsf{st}}} \mathbb{E}_{\boldsymbol{z}_t\sim p_t}\Big[\big\|\boldsymbol{f}^{\mathsf{st}}(\boldsymbol{z}_t, t) - \mathcal{T}_2(\boldsymbol{z}_t)\big\|_2^2\Big].$$

Then, for any partition $\Pi = \{S_1, \ldots, S_K\}$ of the K^2 index pairs (i, j), we have

$$\varepsilon_{t-1:t} \le \sum_{k=1}^{K} \inf_{\boldsymbol{A},\boldsymbol{b}} \mathbb{E}_{\boldsymbol{z}_t\sim p_t}\Big[\sum_{(i,j)\in S_k} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\big\|\boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{b} - \boldsymbol{g}_{ij,t}(\boldsymbol{z}_t)\big\|_2^2\Big].$$



Fix a cluster $S_k \subset \{(i,j)\colon 1 \le i,j \le K\}$ and define

$$W_{k,t}(\boldsymbol{z}_t) = \sum_{(i,j)\in S_k} w_{ij,t}^{(2)}(\boldsymbol{z}_t), \quad \bar{\boldsymbol{g}}_{k,t}(\boldsymbol{z}_t) = \frac{1}{W_{k,t}(\boldsymbol{z}_t)} \sum_{(i,j)\in S_k} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\boldsymbol{g}_{ij,t}(\boldsymbol{z}$$

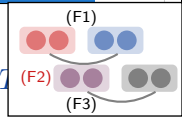
Then for any affine candidate $\boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{b}$,

$$\mathbb{E}_{\boldsymbol{z}_t\sim p_t}\Big[\sum_{(i,j)\in S_k} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\big\|\boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{b} - \boldsymbol{g}_{ij,t}(\boldsymbol{z}_t)\big\|_2^2\Big]$$

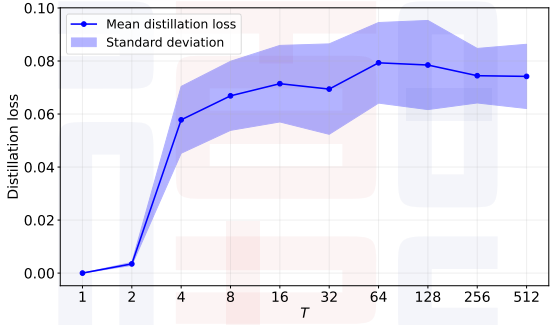
$$= \underbrace{\mathbb{E}_{\boldsymbol{z}_t\sim p_t}\Big[W_{k,t}(\boldsymbol{z}_t)\big\|\boldsymbol{A}\boldsymbol{z}_t + \boldsymbol{b} - \bar{\boldsymbol{g}}_{k,t}(\boldsymbol{z}_t)\big\|_2^2\Big]}_{\text{bias term}} + \underbrace{\mathbb{E}_{\boldsymbol{z}_t\sim p_t}\Big[\sum_{(i,j)\in S_k} w_{ij,t}^{(2)}(\boldsymbol{z}_t)\big\|\bar{\boldsymbol{g}}_{k,t}(\boldsymbol{z}_t) - \boldsymbol{g}_{ij,t}(\boldsymbol{z}_t)\big\|_2^2\Big]}_{\text{variance term}}.$$

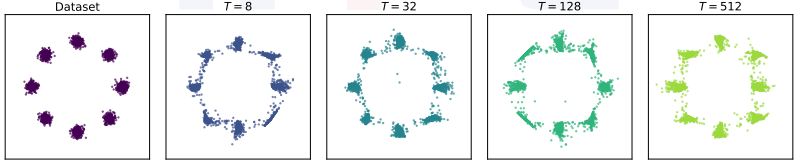
- **Bias term:** Reducible by choosing proper $(\boldsymbol{A}, \boldsymbol{b})$ (i.e., best affine fit to the cluster centroid).
- **Variance term:** Irreducible, reflecting within-cluster heterogeneity of the K^2 affine experts.

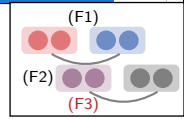
Setup: Student is a $K = 8$ affine MoE. Target is $\mathcal{T}_T$ (vanilla distillation).



- Distillation loss becomes nonzero as $T > 1$ and increases monotonically with T



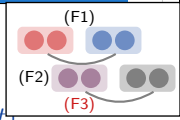




Two-stage distillation: Fix t, choose $k_1, k_2 \in \mathbb{N}$, and set $k = k_1 + k_2$. We distill two stagewise students and then merge them into one operator:

$$z_t \xrightarrow{f^{\mathrm{st}}_{t-k_1+1:t}(\cdot,t)} \widehat{z}_{t-k_1} \xrightarrow{f^{\mathrm{st}}_{t-k+1:t-k_1}(\cdot,t-k_1)} \widehat{z}_{t-k} \quad \Longrightarrow \quad f^{\mathrm{st}}_{t-k+1:t}(\cdot,t).$$

- Stage 1: $f^{\mathrm{st}}_{t-k_1+1:t}(z_t,t)$ approximates $\mathcal{T}_{k_1}(z_t)$, incurring an approximation error
$$\varepsilon_{t-k_1+1:t} := \mathbb{E}_{z_t\sim p_t}\left[\|f^{\mathrm{st}}_{t-k_1+1:t}(z_t) - \mathcal{T}_{k_1}(z_t)\|_2^2\right].$$
- Stage 2: $f^{\mathrm{st}}_{t-k+1:t-k_1}(z_{t-k_1}, t-k_1)$ approximates $\mathcal{T}_{k_2}(z_{t-k_1})$, incurring a distribution shift error
$$\Delta_{t-k_1} := \mathbb{E}_{z_t\sim p_t}\left[\left\|f^{\mathrm{st}}_{t-k+1:t-k_1}\left(f^{\mathrm{st}}_{t-k_1+1:t}(z_t)\right) - \mathcal{T}_{k_2}\left(f^{\mathrm{st}}_{t-k_1+1:t}(z_t)\right)\right\|_2^2\right].$$
- Re-merge: $f^{\mathrm{st}}_{t-k+1:t}(z_t,t)$ approximates $f^{\mathrm{st}}_{t-k+1:t-k_1} \circ f^{\mathrm{st}}_{t-k_1+1:t}$, incurring a merging error
$$\varepsilon^{\mathrm{merge}}_{t-k+1:t} := \mathbb{E}_{z_t\sim p_t}\left[\left\|f^{\mathrm{st}}_{t-k+1:t}(z_t) - (f^{\mathrm{st}}_{t-k+1:t-k_1} \circ f^{\mathrm{st}}_{t-k_1+1:t})(z_t)\right\|_2^2\right].$$



Recall: The two-stage distillation process is

$$z_t \xrightarrow[\varepsilon_{t-k_1+1:t}]{\boldsymbol{f}^{\mathrm{st}}_{t-k_1+1:t}(\cdot,t)} \widehat{z}_{t-k_1} \xrightarrow[\Delta_{t-k_1}]{\boldsymbol{f}^{\mathrm{st}}_{t-k+1:t-k_1}(\cdot,t-k_1)} \widehat{z}_{t-k} \Longrightarrow \boldsymbol{f}^{\mathrm{st}}_{t-k+1:t}(\cdot,t).$$

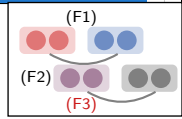
$\varepsilon^{\mathrm{merge}}_{t-k+1:t}$

Theorem (Error propagation across a two-stage merge)

Assume $\mathcal{T}_{k_2}$ is $L^\star$-Lipschitz. Then

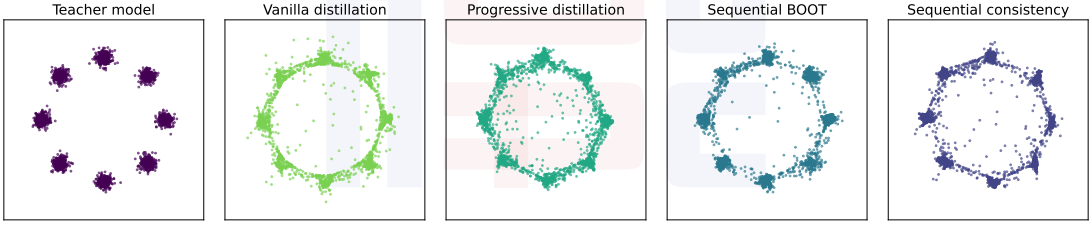
$$\mathbb{E}_{z_t\sim p_t}\left[\|\boldsymbol{f}^{\mathrm{st}}_{t-k+1:t}(z_t)-\mathcal{T}_k(z_t)\|_2^2\right] \le 2\varepsilon^{\mathrm{merge}}_{t-k+1:t} + 4\Delta_{t-k_1} + 4(L^\star)^2\varepsilon_{t-k_1+1:t}.$$

- $\varepsilon^{\mathrm{merge}}_{t-k+1:t}$: Error from compressing the two-stage student composition into one operator.
- Δ_{t-k_1}: Stage-2 error evaluated on $\boldsymbol{f}^{\mathrm{st}}_{t-k_1+1:t}(z_t)$ (distribution shift from the ideal marginal).
- $(L^\star)^2\varepsilon_{t-k_1+1:t}$: Lipschitz amplification of the stage-1 error through $\mathcal{T}_{k_2}$.



Setup: Fix $T = 32$ and $K = 8$. Train student models using canonical strategies.

- Distilled student models interpolate between adjacent modes.
- This behavior aligns with approximation error accumulating across merges.



- We propose a two-phase progression of GAN training: **fitting** and **collapse**.
  - Particles driven by **discriminator gradients** $\nabla d(Z)/d(Z)$ to capture modes. Used **generator steepness** $\mathcal{S}_g(x)$ to measure and analyze mode mixture severity.
  - **Joint gradient** analysis explains collapse; developed early stopping algorithm.
- We decompose the (sub)optimall dynamics into **memorization** and **generalization**.
  - Generated samples **memorize** real data points under the **optimal velocity field**, leveraging geometric insights into the generation paths.
  - In **suboptimal scenarios**, off-subspace components decay while subspace components **generalize within the data-defined subspace**.
- We recast diffusion trajectory distillation as **an operator merging problem**.
  - Merging induces signal shrinkage and yields a **variance-driven phase transition** (sequential BOOT for $\lambda \leq 1$, vanilla distillation for $\lambda \gg 1$).
  - Merging increases target complexity, causing **intrinsic approximation error**, and errors propagate across merges.

- **GAN**: https://openreview.net/forum?id=58gPkcVbFL
- **Flow Matching**: https://www.nature.com/articles/s44387-026-00149-4
- **Operator Merging**: https://www.sciencedirect.com/science/article/pii/S0893608026004831
- Quantitative Approximation: https://arxiv.org/abs/2606.03820v1
- Model Collapse: https://arxiv.org/abs/2503.08117

Thanks for your attention!