ICML 2026 Tutorial

Diffusion and Flow Matching

From Memorization to Generalization & Beyond

Quentin Bertrand & Mathurin Massias

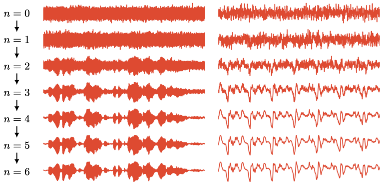
https://memorization-generalization.github.io

Inria | ENS Lyon | Laboratoire Hubert Curien | Mila Affiliated Member | CIFAR Global Scholar

Diffusion and Flow matching are everywhere

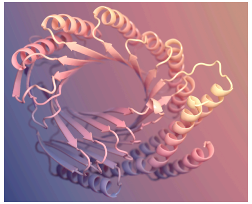
- Diffusion introduced in 2015[1], took off in 2020[2,3]
- Flow matching introduced in 2023[4,5,6]
- Current SOTA techniques for generation of images, audio, videos, proteins...

Stable Diffusion



WaveGrad

Sora



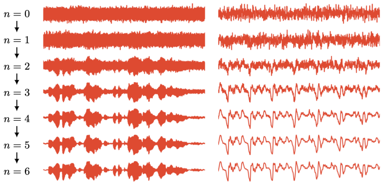
RF Diffusion

[5] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.
[1] J. Sohl-Dickstein et al., **Deep unsupervised learning using nonequilibrium thermodynamics,** In: ICML, 2015.
[2] J. Ho et al., **Denoising diffusion probabilistic models,** In: NeurIPS, 2020.
[3] Y. Song et al., **Score-based generative modeling through stochastic differential equations,** In: ICLR, 2021.
[4] Y. Lipman et al., **Flow matching for generative modeling,** In: ICLR, 2023.
[6] X. Liu et al., **Flow straight and fast: Learning to generate and transfer data with rectified flow,** In: ICLR, 2023.

Diffusion and Flow matching are everywhere

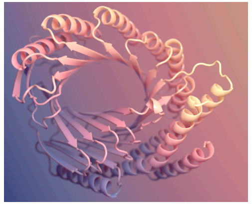
- Diffusion introduced in 2015[1], took off in 2020[2,3]
- Flow matching introduced in 2023[4,5,6]
- Current SOTA techniques for generation of images, audio, videos, proteins...
- And even for text now

Stable Diffusion



WaveGrad

Sora



RF Diffusion

Diffusion Gemma

[5] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.
[1] J. Sohl-Dickstein et al., **Deep unsupervised learning using nonequilibrium thermodynamics,** In: ICML, 2015.
[2] J. Ho et al., **Denoising diffusion probabilistic models,** In: NeurIPS, 2020.
[3] Y. Song et al., **Score-based generative modeling through stochastic differential equations,** In: ICLR, 2021.
[4] Y. Lipman et al., **Flow matching for generative modeling,** In: ICLR, 2023.
[6] X. Liu et al., **Flow straight and fast: Learning to generate and transfer data with rectified flow,** In: ICLR, 2023.

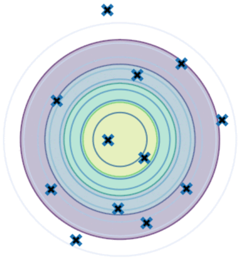
These models have stellar performances but…

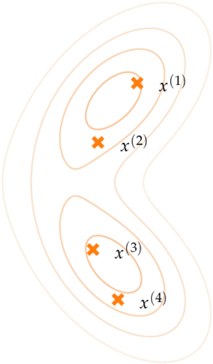
They are mostly black boxes with many puzzles:

These models have stellar performances but…

They are mostly black boxes with many puzzles:

- Optimal models should only generate training data?[7]



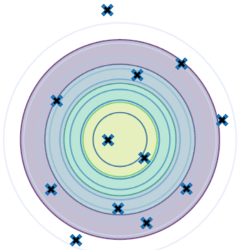


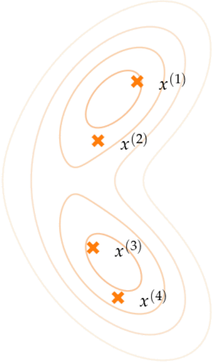
[7] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[8] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[9] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

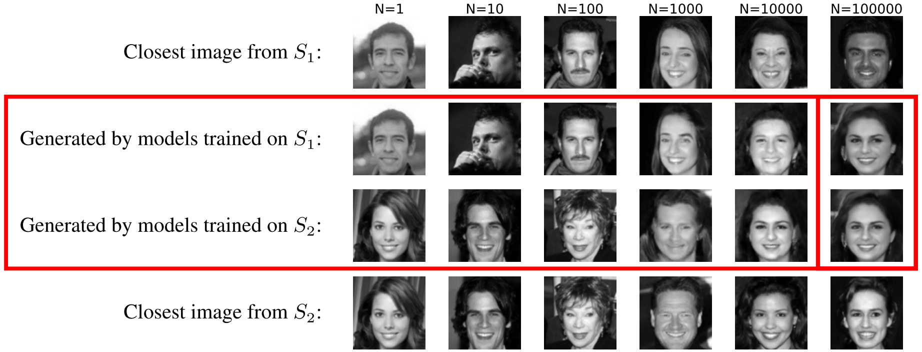
These models have stellar performances but…

They are mostly black boxes with many puzzles:

- Optimal models should only generate training data?[7]
- Models trained on different datasets generate the same images?[8]





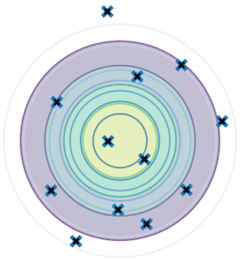


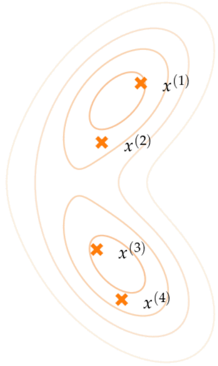
[7] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[8] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[9] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

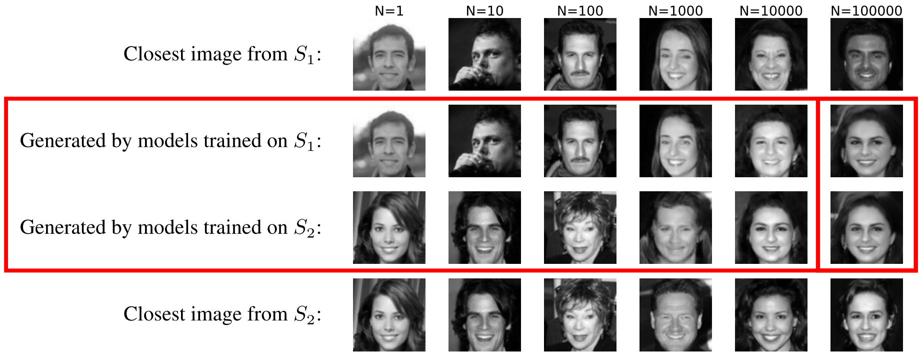
These models have stellar performances but…

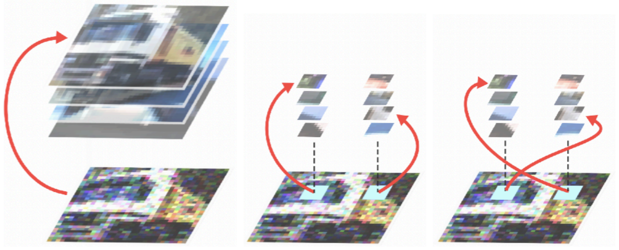
They are mostly black boxes with many puzzles:

- Optimal models should only generate training data?[7]
- Models trained on different datasets generate the same images?[8]
- Generated images are patchworks of training images?[9]









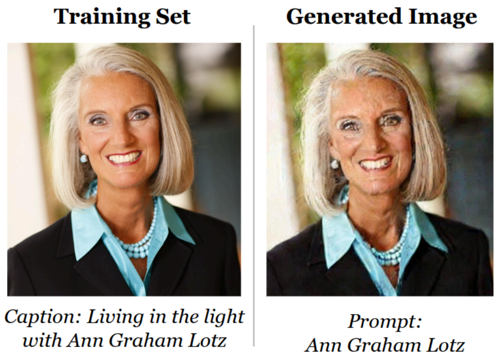
[7] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[8] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[9] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Tutorial goal

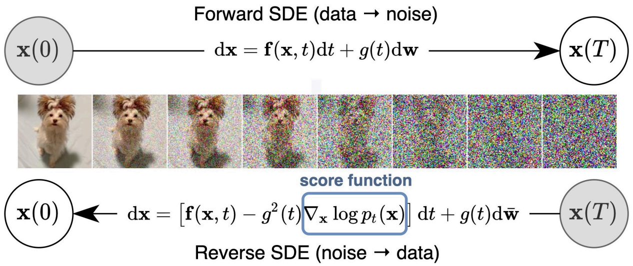
When, how, and why do Diffusion & Flow matching create new data?

Plan

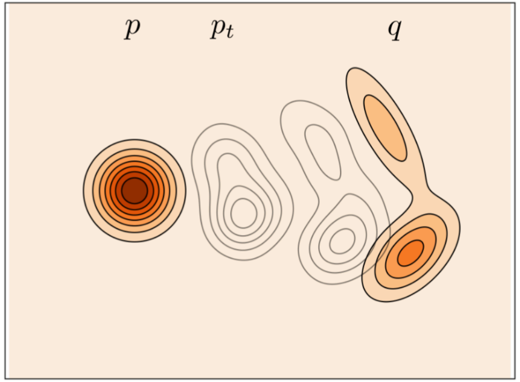
- Introduction to Flow matching and Diffusion: why should they memorize? (40 min)
- Discussion #1 (10 min)



Extracting Training Data from Diffusion Models Carlini et al. (2023)



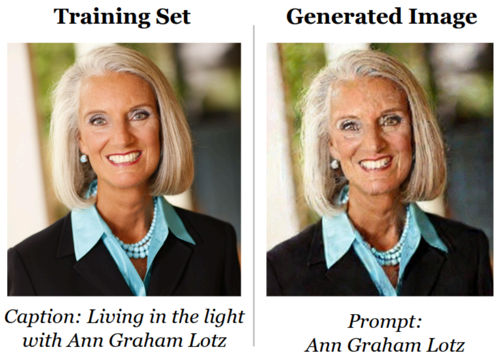
https://yang-song.net/blog/2021/score/



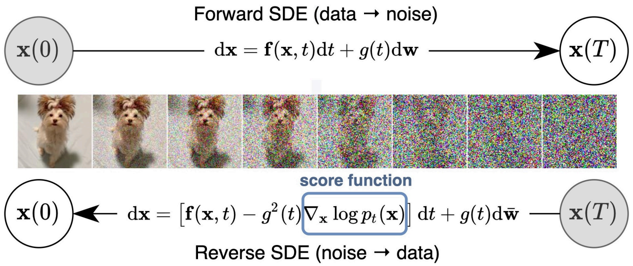
Flow Matching Guide and Code, Lipman et al. (2025)

Plan

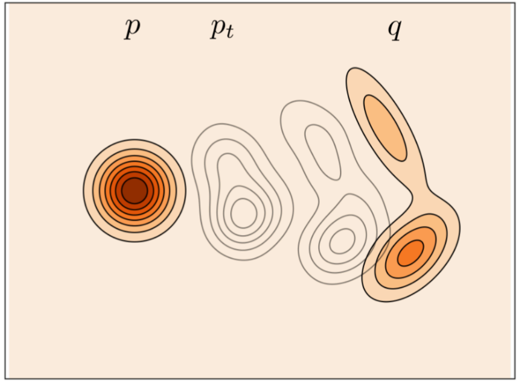
- Introduction to Flow matching and Diffusion: why should they memorize? (40 min)
- Discussion #1 (10 min)
- Toward understanding the generalization of Flow matching and Diffusion (30 min)
- Discussion #2 (10 min)



Extracting Training Data from Diffusion Models Carlini et al. (2023)



https://yang-song.net/blog/2021/score/



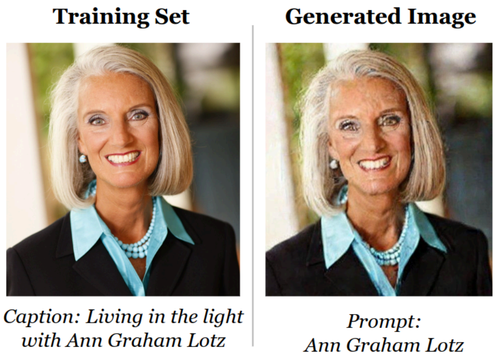
Flow Matching Guide and Code, Lipman et al. (2025)

Plan

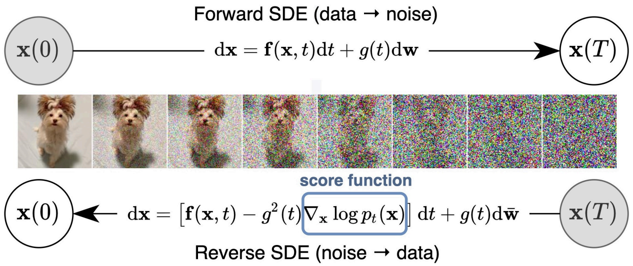
- Introduction to Flow matching and Diffusion: why should they memorize? (40 min)
- Discussion #1 (10 min)

- Toward understanding the generalization of Flow matching and Diffusion (30 min)
- Discussion #2 (10 min)

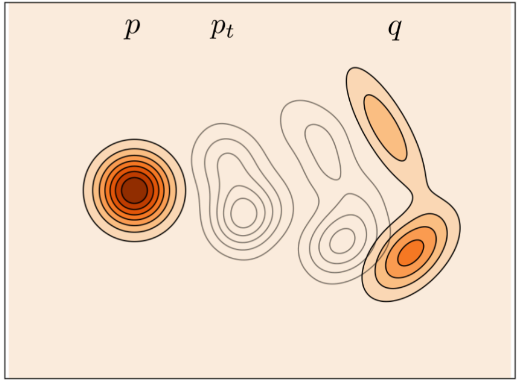
- Beyond generalization (30 min)
- Current open problems (15 min)
- Discussion #3 (15 min)



Extracting Training Data from Diffusion Models Carlini et al. (2023)



https://yang-song.net/blog/2021/score/



Flow Matching Guide and Code, Lipman et al. (2025)

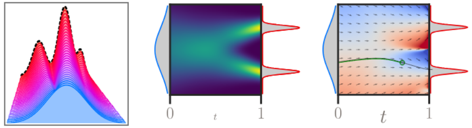
Who we are

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

- Researchers at Inria (France)
- Blog post[10] and contributions in memorization[11,12]

A Visual Dive into Conditional Flow Matching



On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity

Quentin Bertrand[15*], Anne Gagneux[2*], Mathurin Massias[3*], Rémi Emonet[14*]

TRAINING FLOW MATCHING: THE ROLE OF WEIGHTING AND PARAMETERIZATION

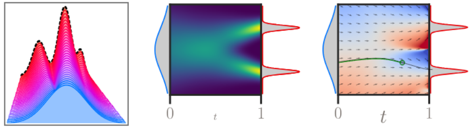
Anne Gagneux[*1], Ségolène Martin[*23], Rémi Gribonval[3] & Mathurin Massias[3]

[10] A. Gagneux et al., **A Visual Dive into Conditional Flow Matching,** In: ICLR Blogposts, 2025.
[11] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.
[12] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: ICLR Delta workshop, 2026.

Who we are

- Researchers at Inria (France)
- Blog post[10] and contributions in memorization[11,12]

A Visual Dive into Conditional Flow Matching



On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity

Quentin Bertrand[15*], Anne Gagneux[2*], Mathurin Massias[3*], Rémi Emonet[14*]

TRAINING FLOW MATCHING: THE ROLE OF WEIGHTING AND PARAMETERIZATION

Anne Gagneux[*1], Ségolène Martin[*23], Rémi Gribonval[3] & Mathurin Massias[3]

- Many thanks to Rémi Emonet, Ségolène Martin, Anne Gagneux, Georges Le Bellier and Eloi Tanguy who contributed to the material





[10] A. Gagneux et al., **A Visual Dive into Conditional Flow Matching,** In: ICLR Blogposts, 2025.
[11] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.
[12] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: ICLR Delta workshop, 2026.

A few disclaimers

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations
- For completeness we include some bonus slides that we will not cover live

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations
- For completeness we include some bonus slides that we will not cover live
- We limit ourselves to unconditional image generation

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations
- For completeness we include some bonus slides that we will not cover live
- We limit ourselves to unconditional image generation
- We could not include all the literature on the topic[1]

[1] If you feel we should have included something, please send us an email!

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations
- For completeness we include some bonus slides that we will not cover live
- We limit ourselves to unconditional image generation
- We could not include all the literature on the topic[1]
- Presentation available at https://memorization-generalization.github.io

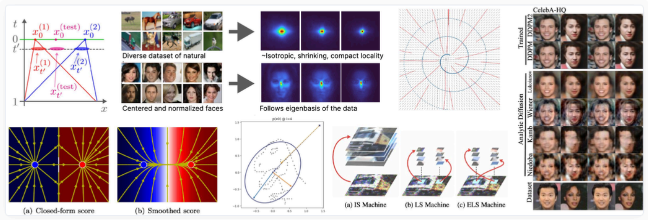
[1] If you feel we should have included something, please send us an email!

A few disclaimers

- We sometimes trade rigor for pedagogy; there will be approximations
- For completeness we include some bonus slides that we will not cover live
- We limit ourselves to unconditional image generation
- We could not include all the literature on the topic[1]
- Presentation available at https://memorization-generalization.github.io
- Other tutorials!

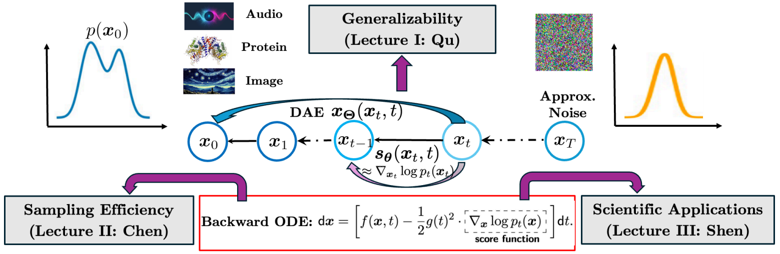
Analytic Understanding of Diffusion Models

A deep dive into the mathematical foundations and analytic perspectives behind modern diffusion-based generative models.



https://analytic-diffusion.github.io/ CVPR 2026

Harnessing Low Dimensionality in Diffusion Models: From Theory to Practice



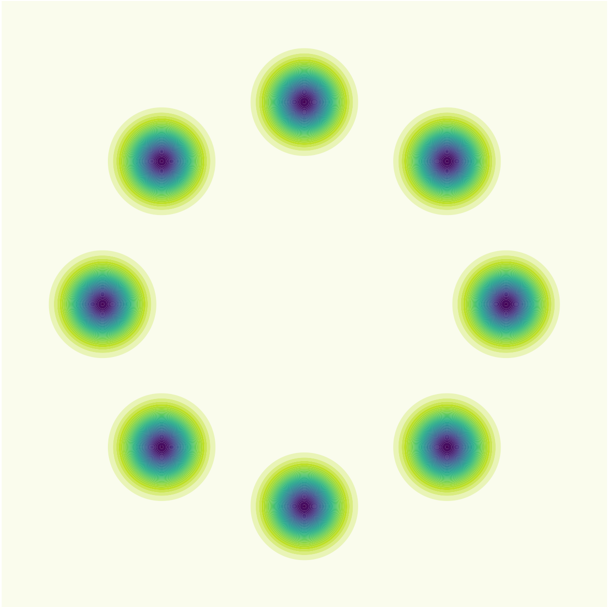
https://icml-2025-tutorial.github.io/ ICML 2025

[1] If you feel we should have included something, please send us an email!

Introduction to
Flow matching and Diffusion

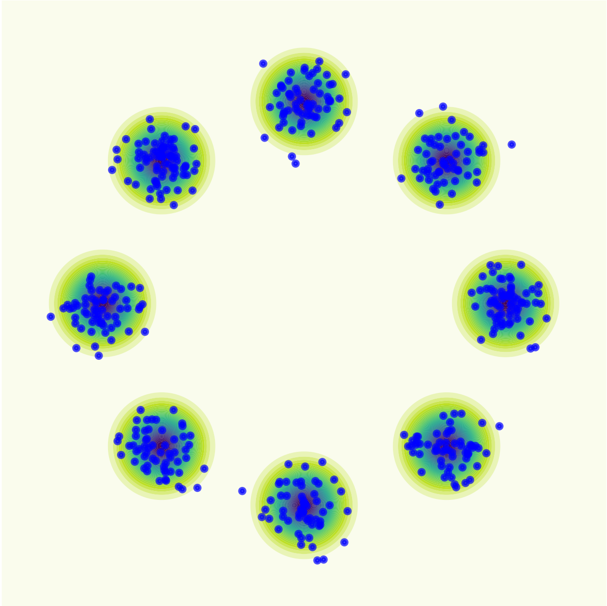
A step back: what is generative modeling?

A step back: what is generative modeling?



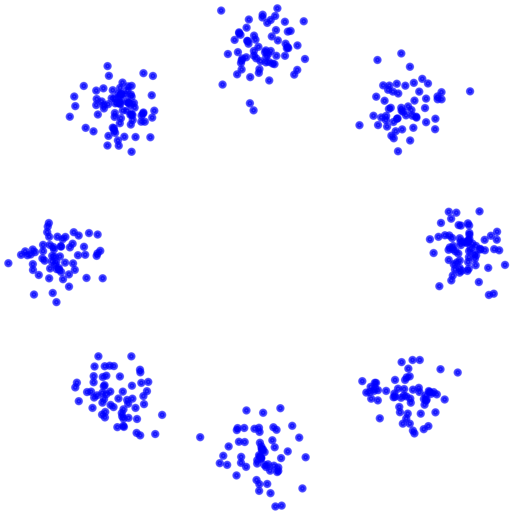
A step back: what is generative modeling?

density p_{data} + samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



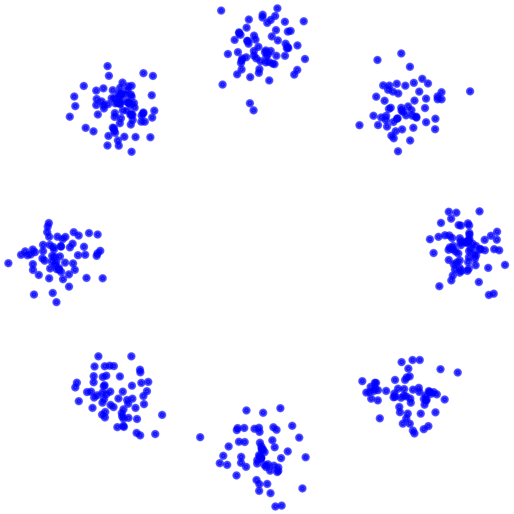
A step back: what is generative modeling?

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



A step back: what is generative modeling?

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$

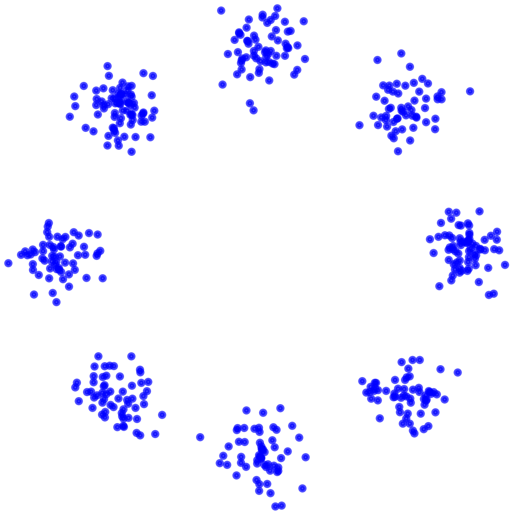


Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

A step back: what is generative modeling?

samples $x^{(1)}, \dots, x^{(n)} \sim p_{\text{data}}$



Problem setting:

- Access to n samples $x^{(1)}, \dots, x^{(n)}$
- $\underbrace{x^{(1)}, \dots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

Goal:

- Draw new samples $\underbrace{x_{\text{new}}^{(1)}, x_{\text{new}}^{(2)}, \dots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

A step back: what is generative modeling?

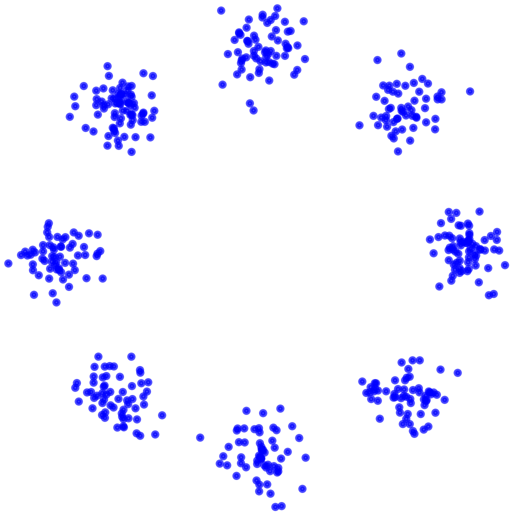
Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

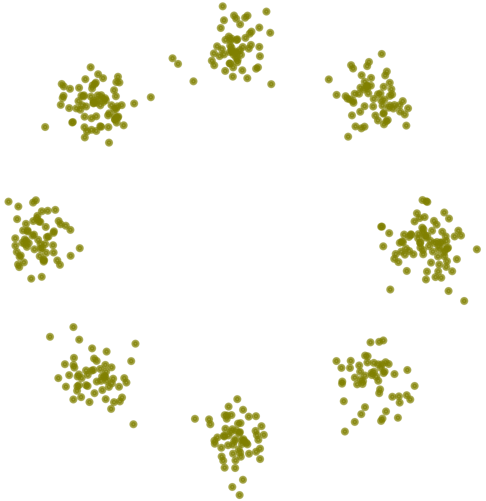
Goal:

- Draw new samples $\underbrace{x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Goal: generate samples $x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots \sim p_{\text{data}}$



A step back: what is generative modeling?

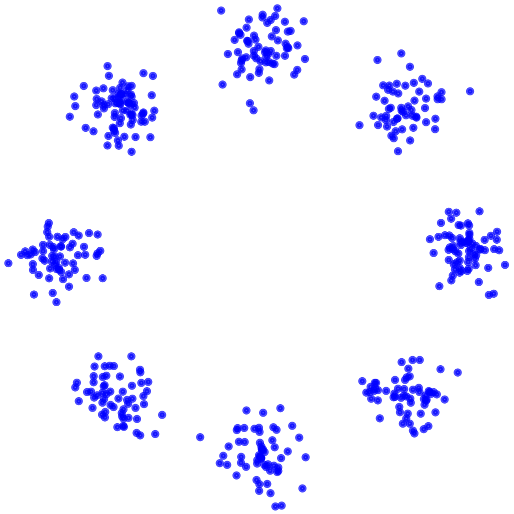
Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

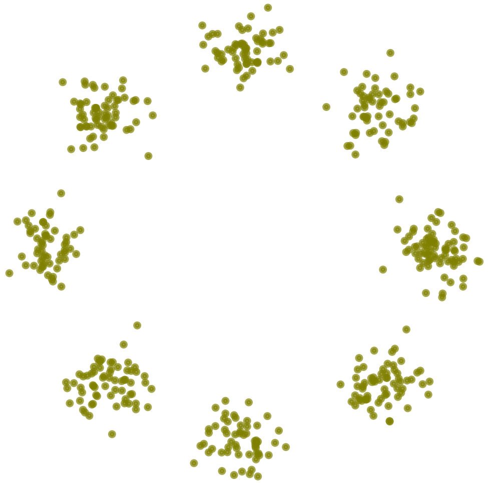
Goal:

- Draw new samples $\underbrace{x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Goal: generate samples $x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots \sim p_{\text{data}}$



A step back: what is generative modeling?

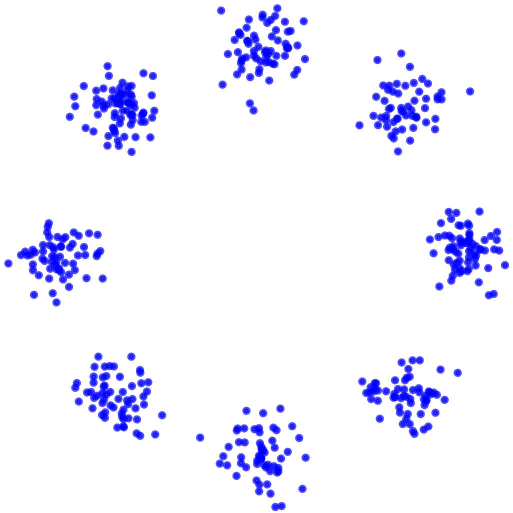
Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

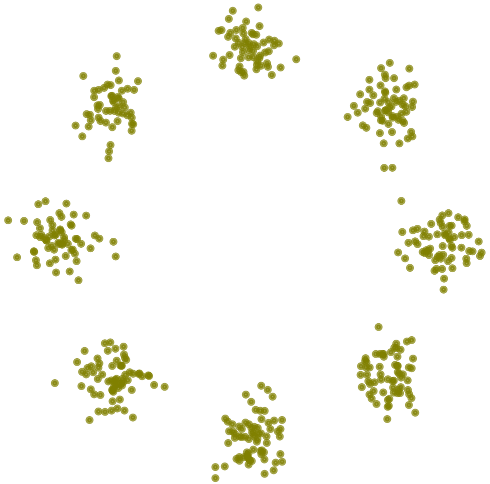
Goal:

- Draw new samples $\underbrace{x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Goal: generate samples $x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots \sim p_{\text{data}}$



A step back: what is generative modeling?

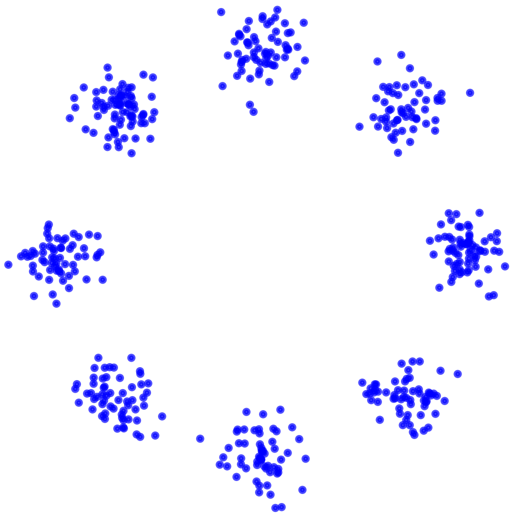
Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

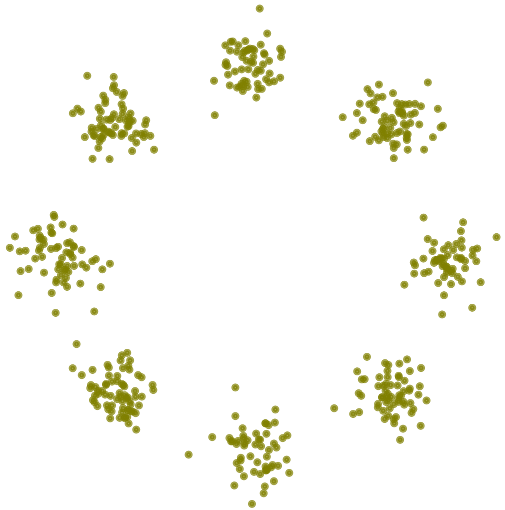
Goal:

- Draw new samples $\underbrace{x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$

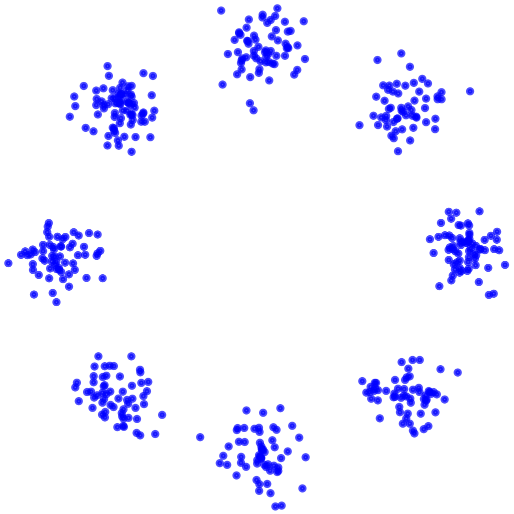


Goal: generate samples $x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots \sim p_{\text{data}}$

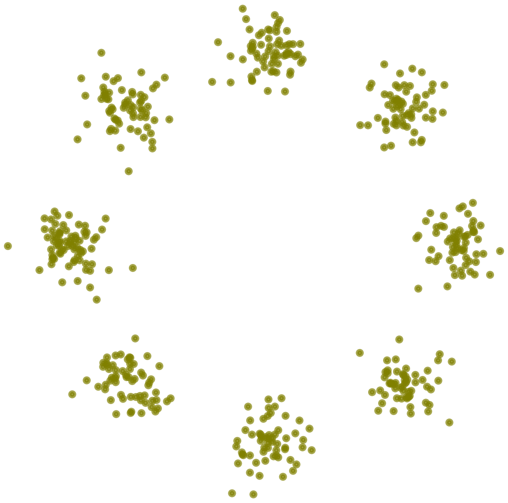


A step back: what is generative modeling?

samples $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Goal: generate samples $x_{\text{new}}^{(1)}, x_{\text{new}}^{(2)}, \ldots \sim p_{\text{data}}$



Problem setting:

- Access to n samples $x^{(1)}, \ldots, x^{(n)}$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

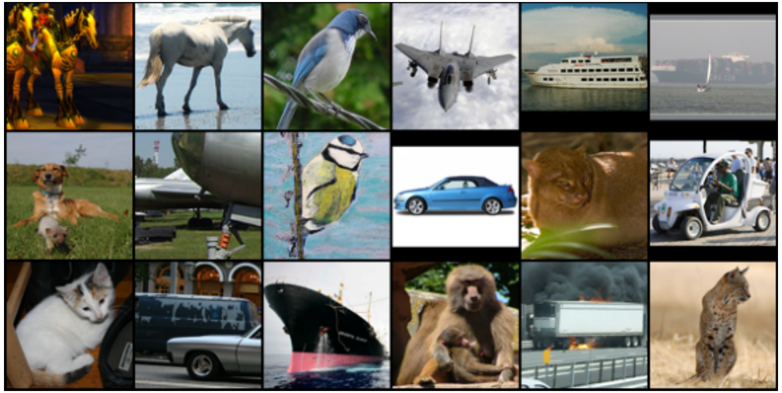
Goal:

- Draw new samples $\underbrace{x_{\text{new}}^{(1)}, x_{\text{new}}^{(2)}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

Generative modeling: an image example

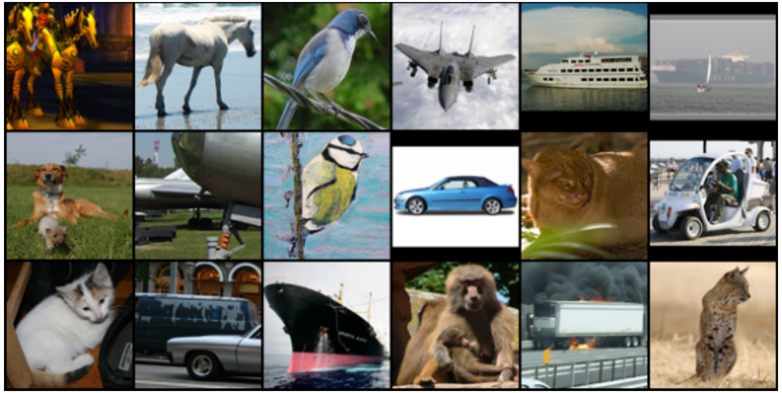
Generative modeling: an image example

data $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Generative modeling: an image example

data $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$

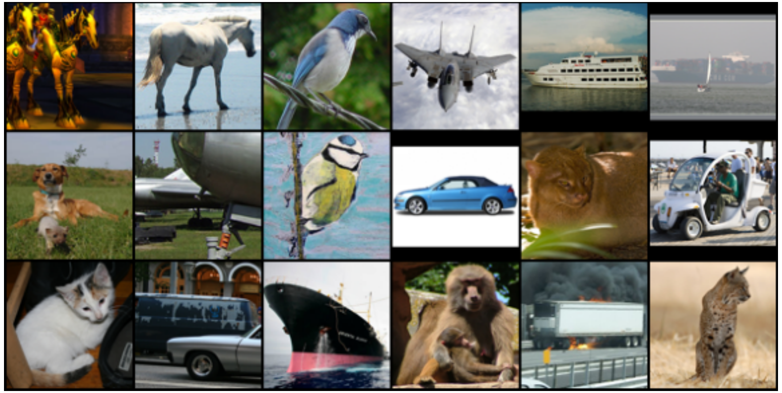


Data:

- Access to samples $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$
- Drawn from p_{data} : $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$
- Could also be audio, texts, proteins, etc

Generative modeling: an image example

data $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



Data:

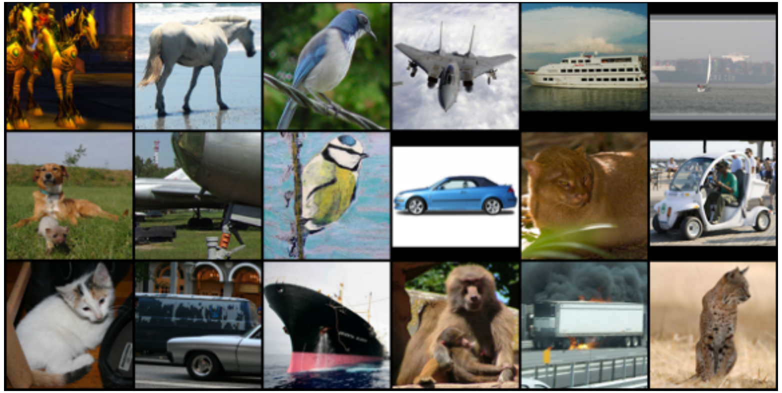
- Access to samples $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$
- Drawn from p_{data} : $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$
- Could also be audio, texts, proteins, etc

Goal:

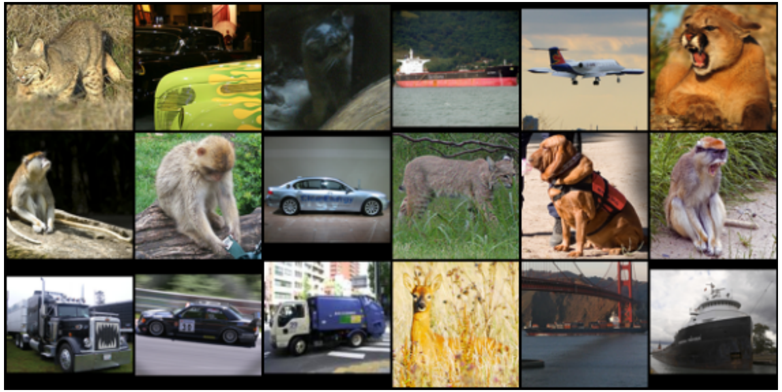
- Create new samples $\underbrace{x^{(1)}_{\text{new}}, x^{(2)}_{\text{new}}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

Generative modeling: an image example

data $x^{(1)}, \ldots, x^{(n)} \sim p_{\text{data}}$



new samples $x_{\text{new}}^{(1)}, x_{\text{new}}^{(2)}, \ldots \sim p_{\text{data}}$



Data:

- Access to samples $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$
- Drawn from p_{data} : $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$
- Could also be audio, texts, proteins, etc

Goal:

- Create new samples $\underbrace{x_{\text{new}}^{(1)}, x_{\text{new}}^{(2)}, \ldots}_{\text{goal}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

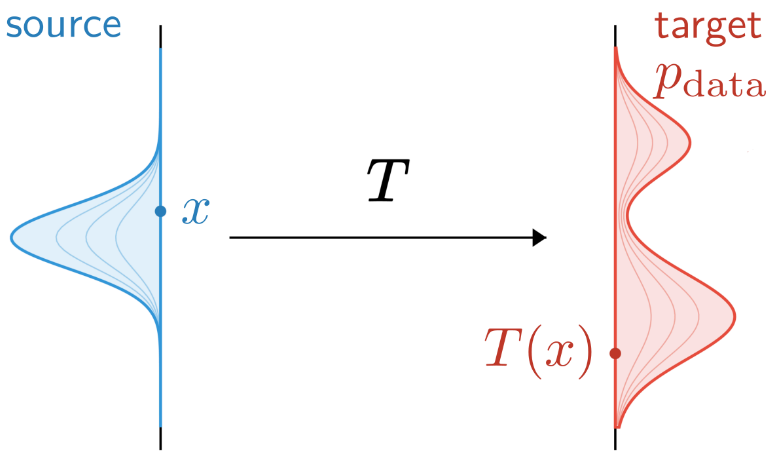
Diffusion and Flow matching: the big picture

Common idea:

Diffusion and Flow matching: the big picture

Common idea:

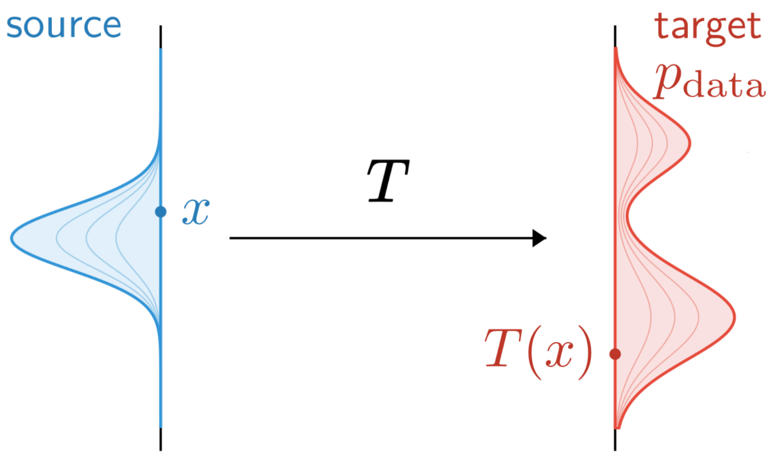
- Learn mapping T sending samples from source distribution to samples from target distribution p_{data}

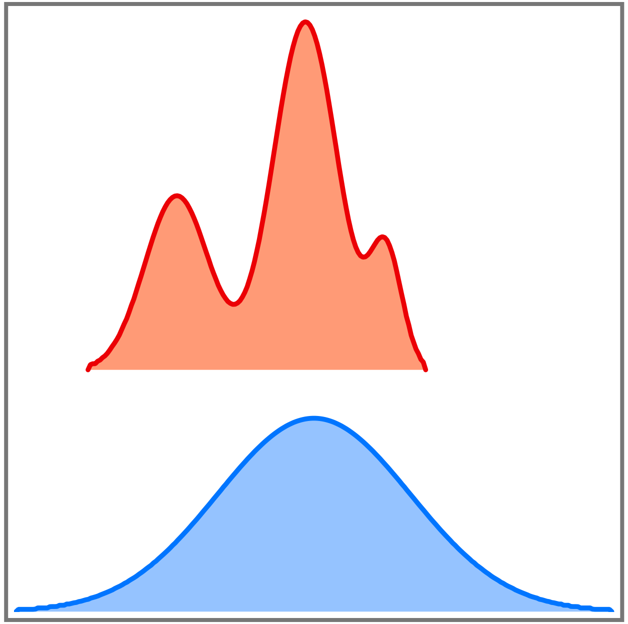


Diffusion and Flow matching: the big picture

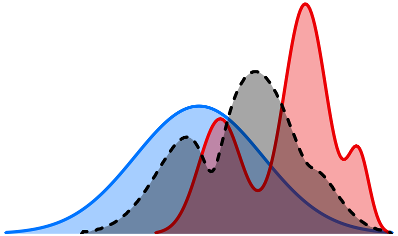
Common idea:

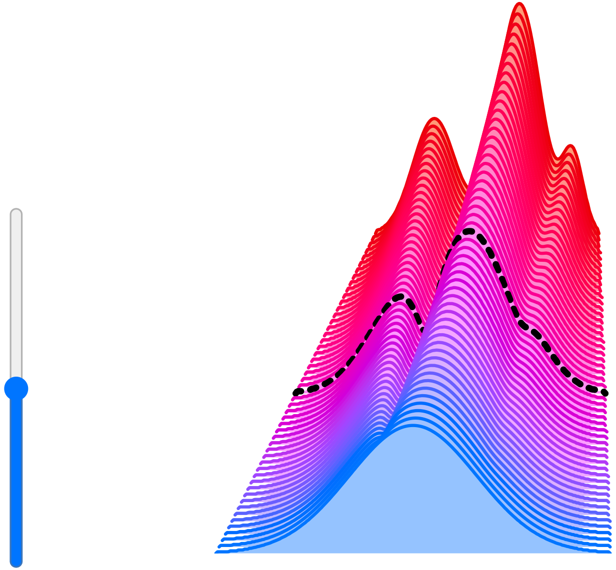
- Learn mapping T sending samples from source distribution to samples from target distribution p_{data}
- Rely on a continuous interpolation between the source and target distributions





Play with paths from to source and target distributions!





NB: Diffusion only supports Gaussian source distribution

Flow matching

Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)

Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$

Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

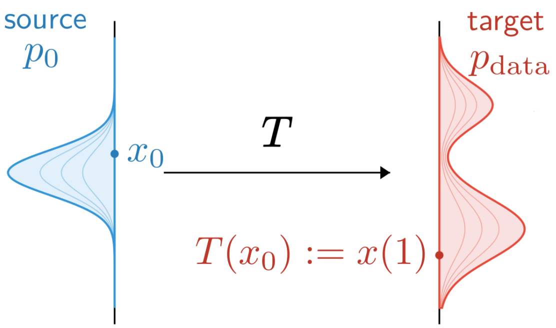
$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$

- The generated sample is defined as the solution at time 1, $x(1) \in \mathbb{R}^d$

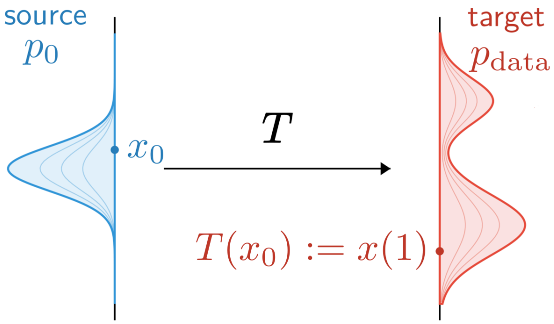


Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0,1] \end{cases}$$

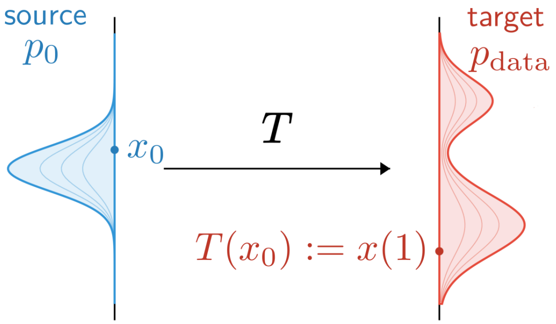
- The generated sample is defined as the solution at time 1, $x(1) \in \mathbb{R}^d$
- Key object: **velocity field** $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$



Flow matching

- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0, 1] \end{cases}$$



- The generated sample is defined as the solution at time 1, $x(1) \in \mathbb{R}^d$
- Key object: **velocity field** $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$
- If velocity $u(x, t)$ is good, $x_0 \sim p_0 \implies x(1) \sim p_{\mathrm{data}}$

$x(1)$ is random because x_0 is

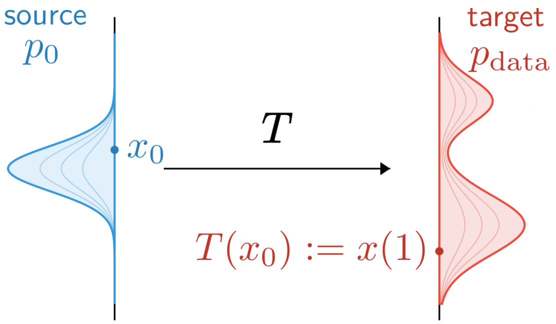
Flow matching

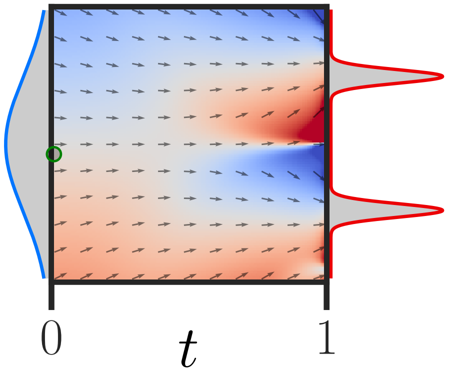
- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0,1] \end{cases}$$

- The generated sample is defined as the solution at time 1, $x(1) \in \mathbb{R}^d$
- Key object: **velocity field** $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$
- If velocity $u(x,t)$ is good, $x_0 \sim p_0 \implies x(1) \sim p_{\text{data}}$

$x(1)$ is random because x_0 is





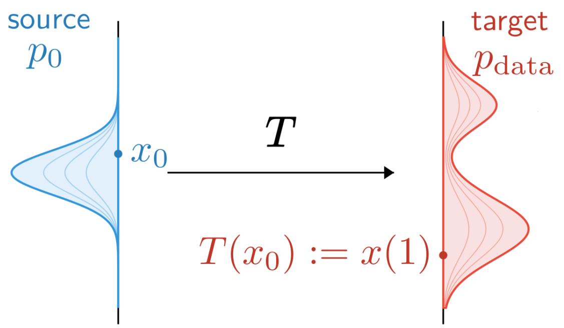
Flow matching

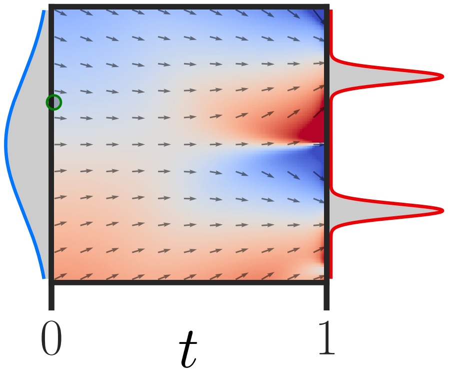
- Convention: go from time $t = 0$ (source) to time $t = 1$ (target)
- The source distribution is p_0 ; we'll take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- Generate samples by solving an Ordinary Differential Equation (ODE):

$$\begin{cases} x(0) = x_0 \sim p_0 \\ \partial_t x(t) = u(x(t), t) \quad \forall t \in [0,1] \end{cases}$$

- The generated sample is defined as the solution at time 1, $x(1) \in \mathbb{R}^d$
- Key object: **velocity field** $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$
- If velocity $u(x,t)$ is good, $x_0 \sim p_0 \implies x(1) \sim p_{\text{data}}$

$x(1)$ is random because x_0 is



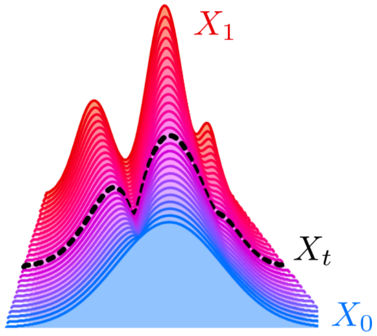


But how to find a *good* velocity $u(x,t)$?

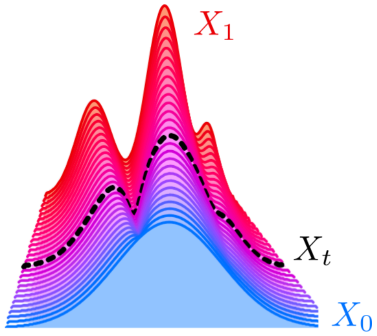
How to find a good velocity?

Setup: define three random variables

- $X_0 \sim p_0$ source distribution: we will take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- $X_1 \sim p_{\mathrm{data}}$ target distribution: clean image
- $X_t := (1-t)X_0 + tX_1$ interpolation: noised image

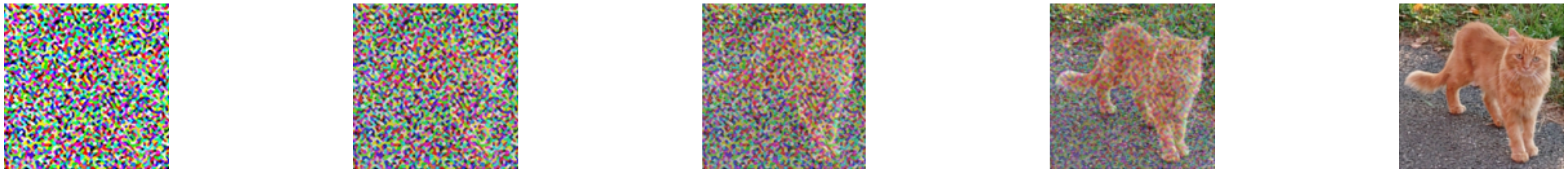


How to find a good velocity?



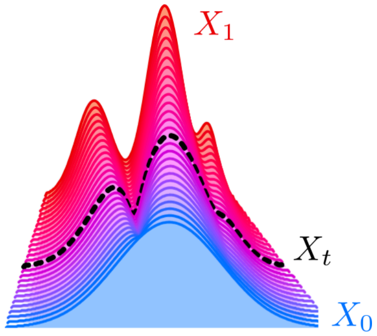
Setup: define three random variables

- $X_0 \sim p_0$ source distribution: we will take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- $X_1 \sim p_{\mathrm{data}}$ target distribution: clean image
- $X_t := (1-t)X_0 + tX_1$ interpolation: noised image



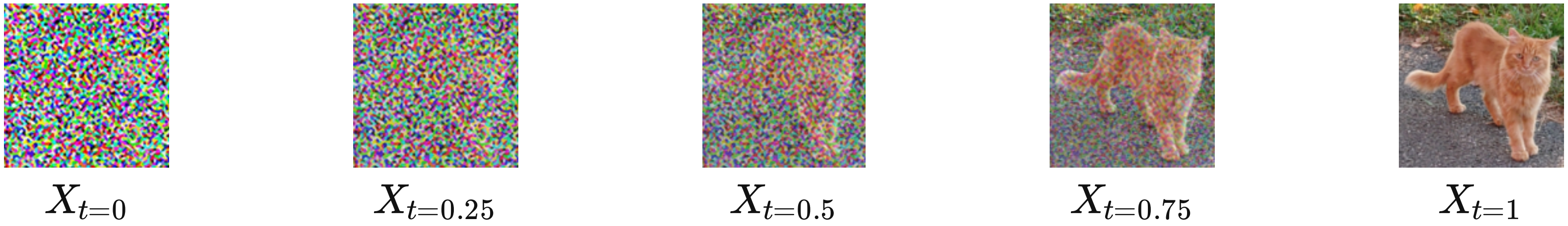
$X_{t=0}$ $X_{t=0.25}$ $X_{t=0.5}$ $X_{t=0.75}$ $X_{t=1}$

How to find a good velocity?



Setup: define three random variables

- $X_0 \sim p_0$ source distribution: we will take $p_0 = \mathcal{N}(0, \mathrm{Id}_d)$
- $X_1 \sim p_{\text{data}}$ target distribution: clean image
- $X_t := (1-t)X_0 + tX_1$ interpolation: noised image



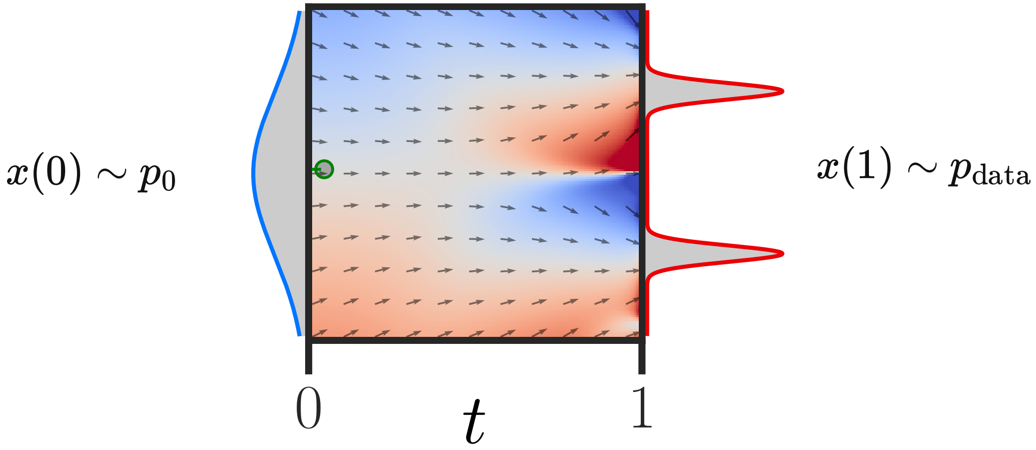
The FM magic[5,13,14] : $u(x,t) := \mathbb{E}[X_1 - X_0 | X_t = x]$ is a *good* velocity:

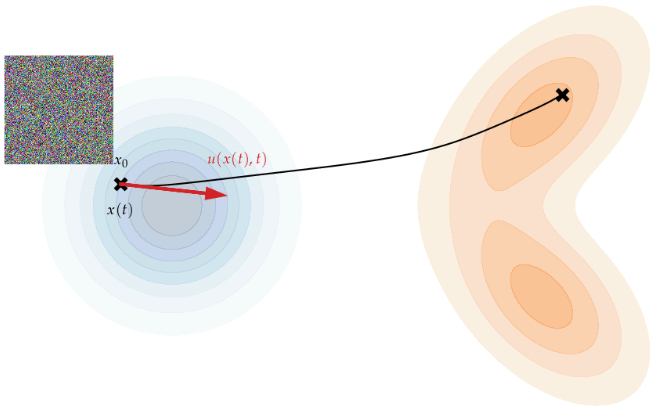
$$x(0) \sim p_0 \implies x(1) \sim p_{\text{data}}$$

[5] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.

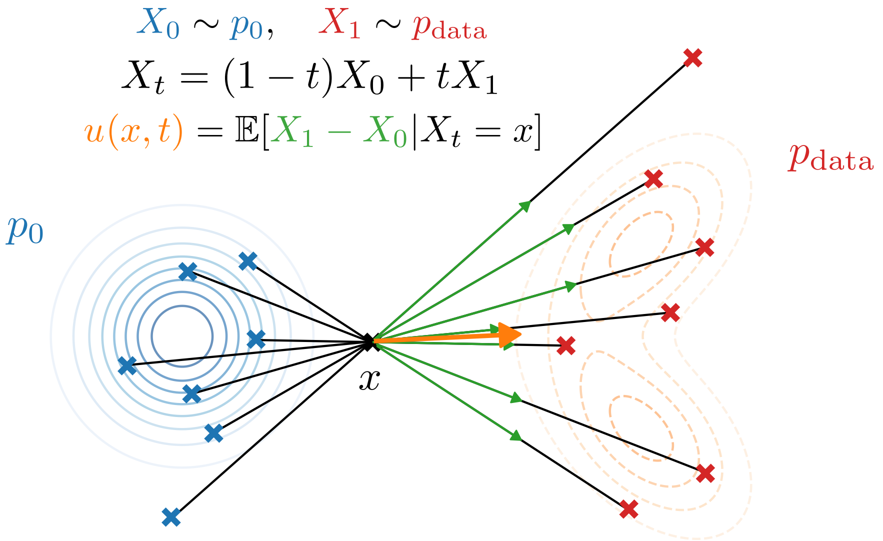
[13] Y. Lipman et al., **Flow Matching for Generative Modeling,** In: ICLR, 2023.

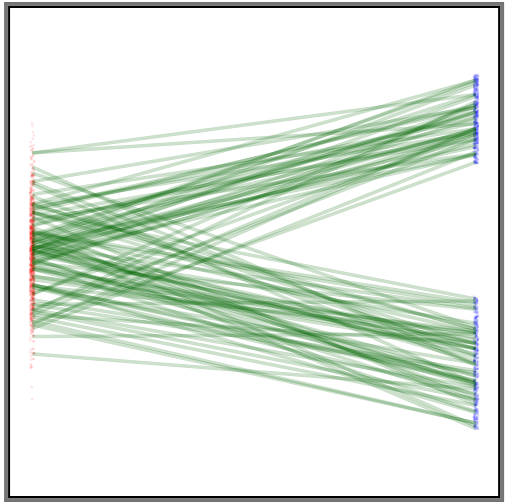
[14] X. Liu et al., **Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow,** In: ICLR, 2023.


$x(0) \sim p_0$
$x(1) \sim p_{\text{data}}$
0
t
1


x_0
$u(x(t), t)$
$x(t)$

Why the conditional expectation is "good"





More in our blog post: **A Visual dive into conditional flow matching**[10]: https://dl.heeere.com/cfm/

[10] A. Gagneux et al., **A Visual Dive into Conditional Flow Matching,** In: ICLR Blogposts, 2025.

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \left\| u_\theta(\underbrace{(1-t)x_0 + tx_1}_{=x_t}, t) - (x_1 - x_0) \right\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$

$$x_1 \sim p_{\text{data}}$$

$$t \sim \text{Uniform}([0,1])$$

$$x_t = (1-t)x_0 + tx_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \left\| u_\theta(\underbrace{(1-t)x_0 + t x_1}_{=x_t}, t) - (x_1 - x_0) \right\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \mathrm{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + t x_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta(\underbrace{(1-t)x_0 + tx_1}_{=x_t}, t) - (x_1 - x_0)\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

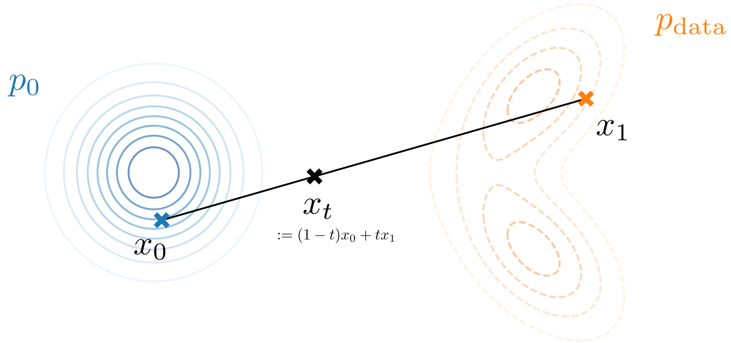
SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta(\underbrace{(1-t)x_0 + tx_1}_{=x_t}, t) - (x_1 - x_0)\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$



SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

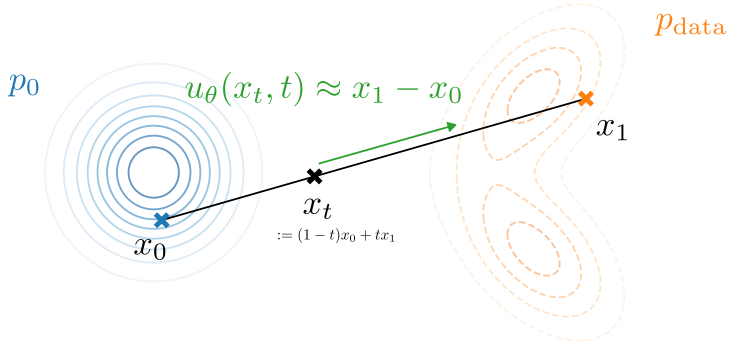
How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \left\| u_\theta(\underbrace{(1-t)x_0 + tx_1}_{=x_t}, t) - (x_1 - x_0) \right\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$



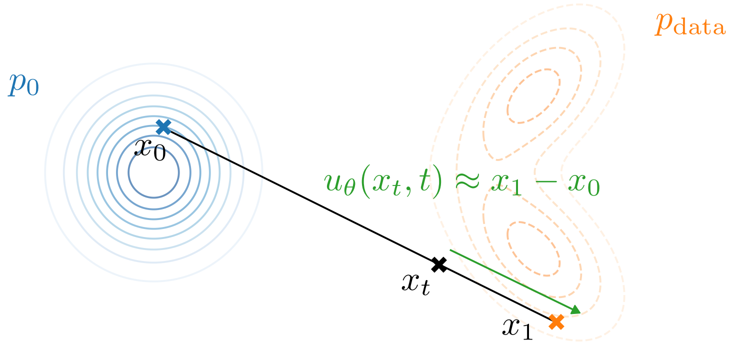
How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta(\underbrace{(1-t)x_0 + tx_1}_{=x_t}, t) - (x_1 - x_0)\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

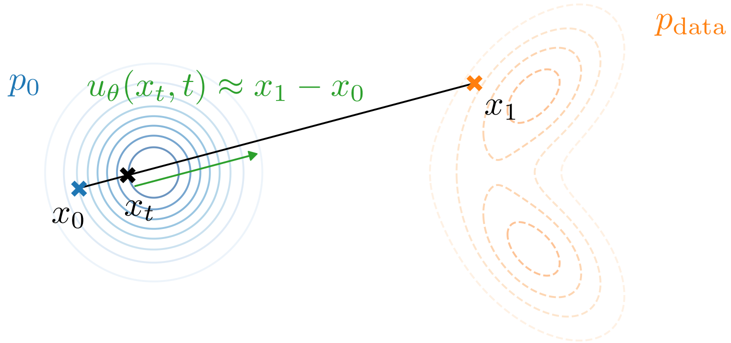


How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \left\| u_\theta(\underbrace{(1-t)x_0 + t x_1}_{=x_t}, t) - (x_1 - x_0) \right\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \text{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + t x_1$$



SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \left\| u_\theta(\underbrace{(1-t)x_0 + t x_1}_{=x_t}, t) - (x_1 - x_0) \right\|^2$$

Train practically with:

$$x_0 \sim p_0 = \mathcal{N}(0, \text{Id}_d)$$

$$x_1 \sim p_{\text{data}}$$

$$t \sim \text{Uniform}([0,1])$$

$$x_t = (1-t)x_0 + t x_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

few days of training and you're done

How to train a Flow matching model?

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \text{unif}([0,1])}} \| u_\theta(\underbrace{(1-t)x_0 + t x_1}_{= x_t}, t) - (x_1 - x_0) \|^2$$

Train practically with:

$$\begin{aligned} x_0 &\sim p_0 = \mathcal{N}(0, \mathrm{Id}_d) \\ x_1 &\sim p_{\text{data}} \\ t &\sim \text{Uniform}([0,1]) \\ x_t &= (1-t)x_0 + t x_1 \end{aligned}$$

SGD step on θ with loss $\| u_\theta(x_t, t) - (x_1 - x_0) \|^2$

few days of training and you're done

the minimizer is $u(x,t) = \mathbb{E}[X_1 - X_0 | X_t = x]$

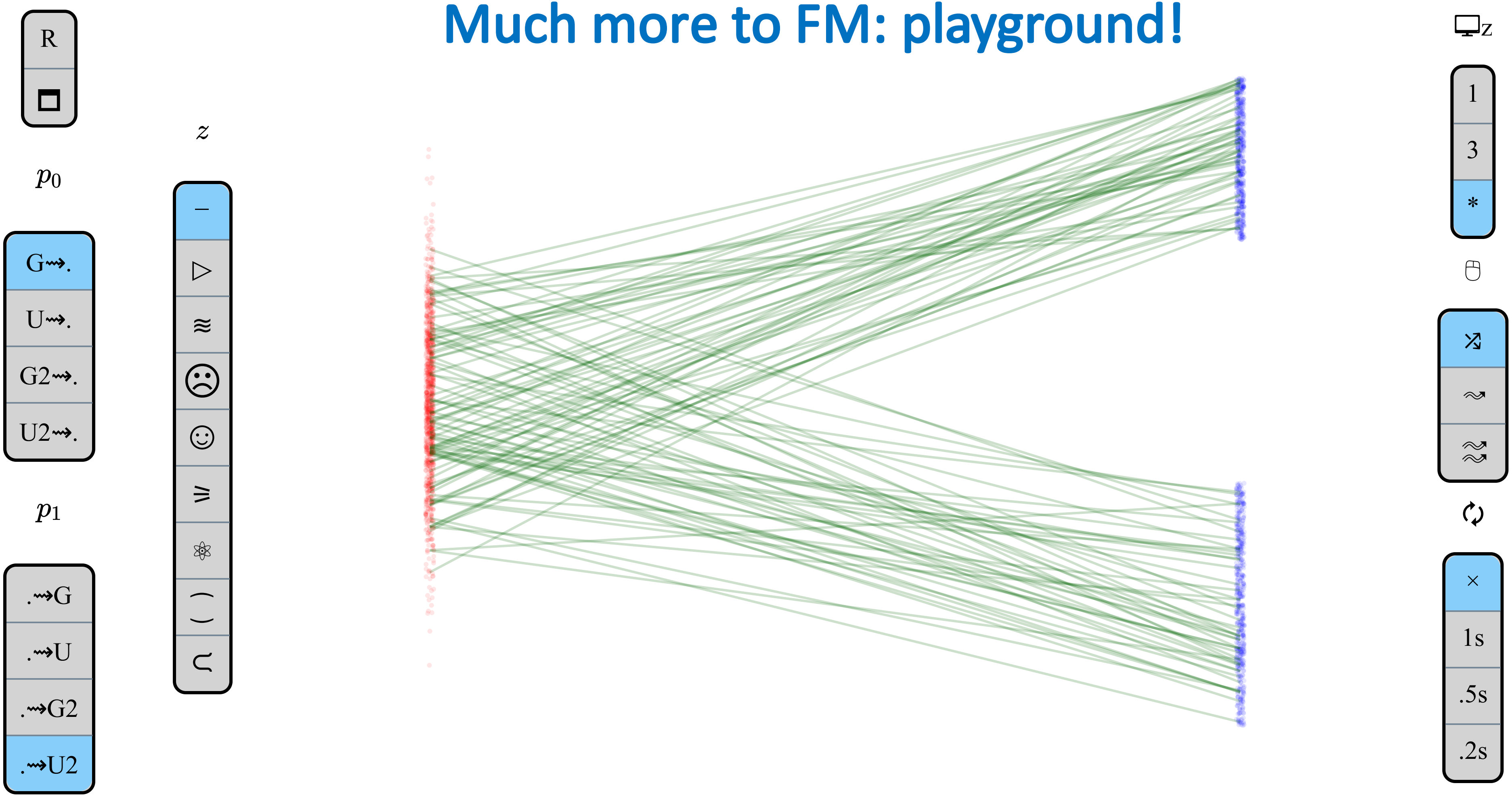
the minimizer (over all measurable functions)

Bonus: much more to FM[15,16]

- Source distribution p_0 can be arbitrary
- Time sampling need not be uniform
- In training, x_0 and x_1 need not be sampled independently[17,18]
- X_t need not equal $(1-t)X_0 + tX_1$
- In practice guidance/conditioning is used a lot

[15] E. Pierret et al., **Flow Matching for Applied Mathematicians,** In: , 2026.
[16] Y. Lipman et al., **Flow matching guide and code,** In: arXiv preprint arXiv:2412.06264, 2024.
[17] A. Pooladian et al., **Multisample flow matching: Straightening flows with minibatch couplings,** In: ICML, 2023.
[18] A. Tong et al., **Improving and generalizing flow-based generative models with minibatch optimal transport,** In: TMLR, 2023.

Much more to FM: playground!



Diffusion

Only thing to know: it's equivalent to Flow matching[19]!

Diffusion Meets Flow Matching: Two Sides of the Same Coin

https://diffusionflow.github.io/

[19] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: ICLR Blogposts, 2025.

Diffusion

Diffusion

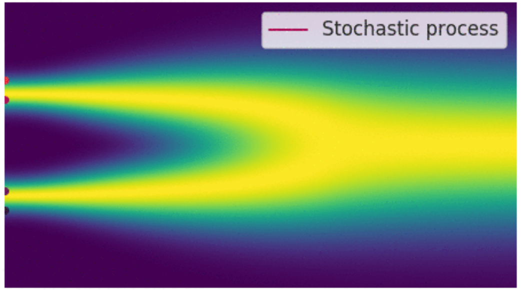
- Convention differs from Flow matching: $t = 0$ corresponds to clean data

Diffusion

- Convention differs from Flow matching: $t = 0$ corresponds to clean data
- Similar idea: progressively corrupt image by adding Gaussian noise

Diffusion

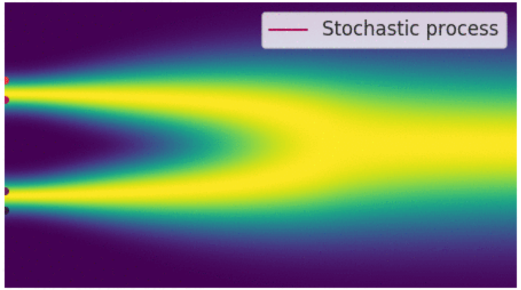
- Convention differs from Flow matching: $t = 0$ corresponds to clean data
- Similar idea: progressively corrupt image by adding Gaussian noise
- Trajectories are stochastic

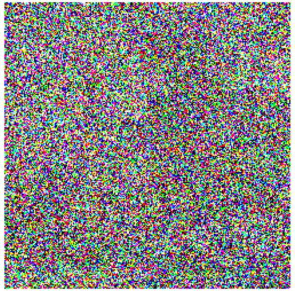


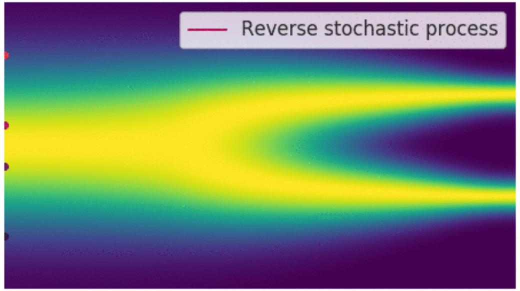
https://yang-song.net/blog/2021/score/

Diffusion

- Convention differs from Flow matching: $t = 0$ corresponds to clean data
- Similar idea: progressively corrupt image by adding Gaussian noise
- Trajectories are stochastic
- One learns to reverse the noising process



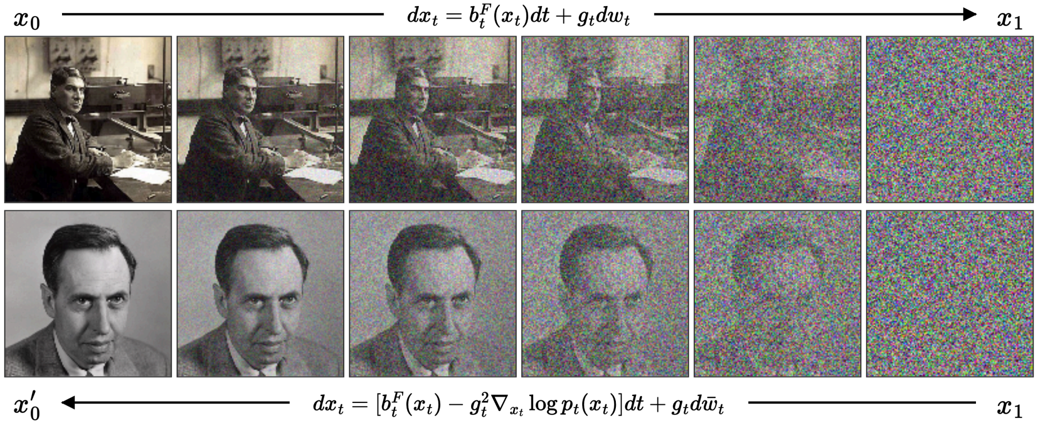




https://yang-song.net/blog/2021/score/

Learning Diffusion

- Instead of learning the velocity u, diffusion learns to reverse the corruption process
- It is driven by a Stochastic Differential Equation (SDE)

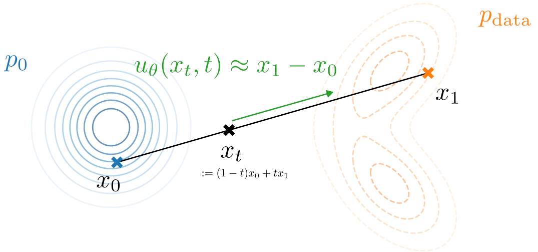


- A key quantity to learn is the *score* $\nabla \log p_t(x_t) \in \mathbb{R}^d$ (p_t is the density of X_t)

  It is theoretically equivalent to learn to predict the score, the added noise or the original image from the corrupted image x_t
- Training procedure is similar to Flow matching: sample clean image, noise it, learn to denoise it

Memorization in Diffusion & Flow Matching

A small caveat in training



Flow matching training:

$$x_0 \sim p_0 = \mathcal{N}(0, \mathrm{Id}_d)$$
$$x_1 \sim p_{\mathrm{data}}$$
$$t \sim \mathrm{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$

Flow matching code

Flow Matching

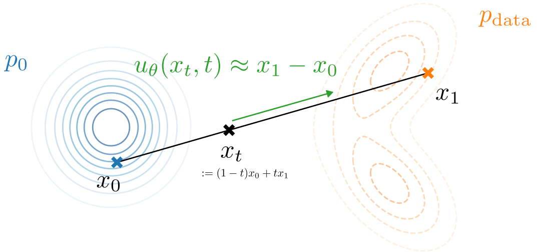
Yaron Lipman[1], Marton Havas
Ricky T. Q. Chen[1], David Lope
[1]FAIR at Meta, [2]MIT CSAI

```
# training
flow = Flow()
optimizer = torch.optim.Adam(flow.parameters(), 1e-2)
loss_fn = nn.MSELoss()

for _ in range(10000):
  x_1  = Tensor(make_moons(256, noise=0.05)[0])
  x_0  = torch.randn_like(x_1)
  t    = torch.rand(len(x_1), 1)
  x_t  = (1 - t) * x_0 + t * x_1
  dx_t = x_1 - x_0
  optimizer.zero_grad()
  loss_fn(flow(x_t, t), dx_t).backward()
  optimizer.step()
```

No si
ODE

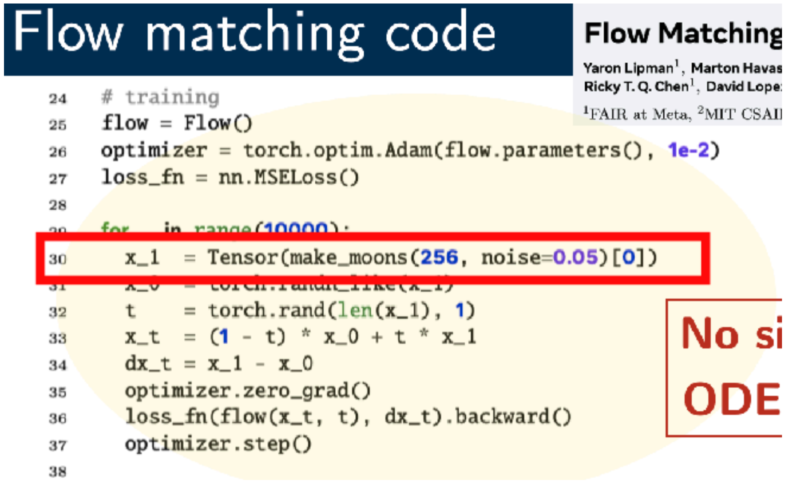
A small caveat in training



Flow matching training:

$$x_0 \sim p_0 = \mathcal{N}(0, \mathrm{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \mathrm{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

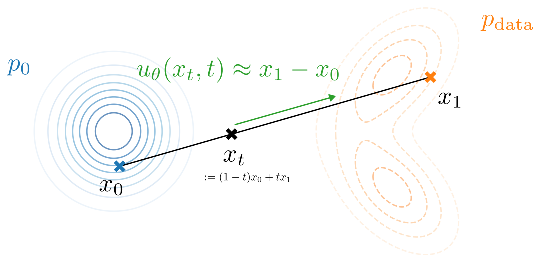
SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$




```
# training
flow = Flow()
optimizer = torch.optim.Adam(flow.parameters(), 1e-2)
loss_fn = nn.MSELoss()

for _ in range(10000):
    x_1  = Tensor(make_moons(256, noise=0.05)[0])
    x_0  = torch.randn_like(x_1)
    t    = torch.rand(len(x_1), 1)
    x_t  = (1 - t) * x_0 + t * x_1
    dx_t = x_1 - x_0
    optimizer.zero_grad()
    loss_fn(flow(x_t, t), dx_t).backward()
    optimizer.step()
```

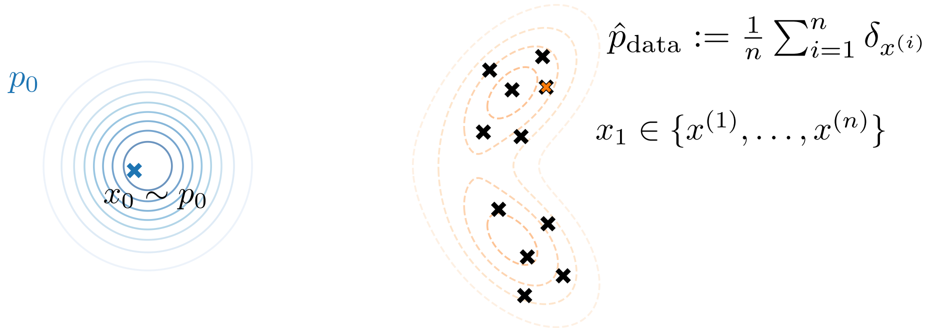
A small caveat in training



Flow matching training:

$$x_0 \sim p_0 = \mathcal{N}(0, \mathrm{Id}_d)$$
$$x_1 \sim p_{\text{data}}$$
$$t \sim \mathrm{Uniform}([0,1])$$
$$x_t = (1-t)x_0 + tx_1$$

SGD step on θ with loss $\|u_\theta(x_t, t) - (x_1 - x_0)\|^2$



$$\text{only usable loss:} \quad \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|^2$$

Expectation vs reality

only usable loss: $\mathbb{E}_{\substack{x_0\sim p_0 \\ x_1\sim \hat{p}_{\text{data}} \\ t\sim\text{unif}([0,1])}} \|u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|^2$

Proposition: when p_{data} is replaced by $\hat{p}_{\text{data}} := \frac{1}{n}\sum_{i=1}^n \delta_{x^{(i)}}$, the FM loss minimizer has a closed-form:

$$\hat{u}^\star(x,t) := \sum_{i=1}^n \lambda_i(x,t)\frac{x^{(i)} - x}{1-t}$$

with $\lambda_i(x,t) = \frac{\exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i)}\|^2)}{\sum_{i'=1}^n \exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2)})$

$\lambda(x,t) = \text{softmax}((-\frac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2)_{i'=1,\dots,n}) \in \mathbb{R}^n$

Expectation vs reality

$$\text{only usable loss:} \quad \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|^2$$

Proposition: when p_{data} is replaced by $\hat{p}_{\text{data}} := \frac{1}{n}\sum_{i=1}^{n} \delta_{x^{(i)}}$, the FM loss minimizer has a closed-form:

$$\hat{u}^\star(x,t) := \sum_{i=1}^{n} \lambda_i(x,t) \frac{x^{(i)} - x}{1-t}$$

$$\text{with} \quad \lambda_i(x,t) = \frac{\exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i)}\|^2)}{\sum_{i'=1}^{n} \exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2}\Big)$$

$$\lambda(x,t) = \text{softmax}((-\tfrac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2)_{i'=1,\ldots,n}) \in \mathbb{R}^n$$

$\hat{u}^\star$ blows up as $t \to 1$ unless $x \approx x^{(i)}$

Expectation vs reality

only usable loss: $\mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \text{unif}([0,1])}} \|u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)\|^2$

Proposition: when p_{data} is replaced by $\hat{p}_{\text{data}} := \frac{1}{n}\sum_{i=1}^n \delta_{x^{(i)}}$, the FM loss minimizer has a closed-form:

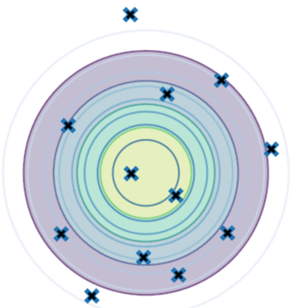
$$\hat{u}^\star(x,t) := \sum_{i=1}^n \lambda_i(x,t) \frac{x^{(i)} - x}{1-t}$$

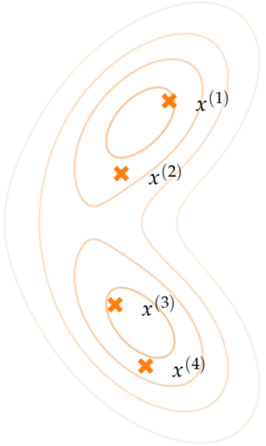
with $\lambda_i(x,t) = \frac{\exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i)}\|^2)}{\sum_{i'=1}^n \exp(-\frac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2)})$

$\lambda(x,t) = \text{softmax}((-\frac{1}{2(1-t)^2}\|x - tx^{(i')}\|^2)_{i'=1,\dots,n}) \in \mathbb{R}^n$

$\hat{u}^\star$ blows up as $t \to 1$ unless $x \approx x^{(i)}$

it only generates training points!





What about diffusion?

- Training diffusion is a particular instance of Flow matching parametrization
- The ideal diffusion score $s^\star(x,t) = \nabla \log p_t(x)$ is related to the ideal velocity by $s(x,t) = \frac{tu^\star(x,t)-x}{1-t}$
- Closed-form minimizer for velocity $\implies$ closed-form minimizer for score! (which also blows up)

Diffusion suffers from the same curse!

Wrap-up of Part I

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models
- Simple training algorithms

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models
- Simple training algorithms
- Closed-form for the ideal score/velocity

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models
- Simple training algorithms
- Closed-form for the ideal score/velocity
- Closed-form can only generate training samples

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models
- Simple training algorithms
- Closed-form for the ideal score/velocity
- Closed-form can only generate training samples
- Properly trained Flow matching and diffusion should only generate training data (like all unregularized generative models)

Wrap-up of Part I

- Flow matching & Diffusion: powerful generative models
- Simple training algorithms
- Closed-form for the ideal score/velocity
- Closed-form can only generate training samples
- Properly trained Flow matching and diffusion should only generate training data (like all unregularized generative models)
- So why do they generate new samples?

ICML 2026 Tutorial

Diffusion and Flow Matching

Part II: From Memorization to Generalization

Quentin Bertrand & Mathurin Massias

https://memorization-generalization.github.io

Inria | ENS Lyon | Laboratoire Hubert Curien | Mila Affiliated Member | CIFAR Global Scholar

Generative models: high dimensional density estimation

DALL·E, Ramesh et al.

Generative models: high dimensional density estimation

DALL·E, Ramesh et al.

Diffusion & Flow matching models sample from **high dimensional** densities

An unavoidable curse?

- Data in dimension d
  - e.g. moderate size images $d = 256 \times 256 \times 3$
- Estimation of the joint density $p(x_1, \ldots, x_d)$
- Number of samples needed to estimate a density for data in dimension d ?
  - Standard machine learning theory: exponential in d

An unavoidable curse?

- Data in dimension d
  - e.g. moderate size images $d = 256 \times 256 \times 3$
- Estimation of the joint density $p(x_1, \ldots, x_d)$
- Number of samples needed to estimate a density for data in dimension d ?
  - Standard machine learning theory: exponential in d

In theory: density estimation in such high dimension should fail!

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
    - 2 **disjoints** sets S_n^1 and S_n^2
    - of size n

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n

---

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2

---

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0

---

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0

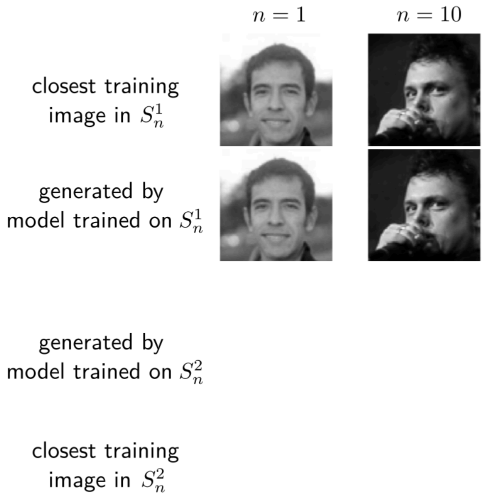
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



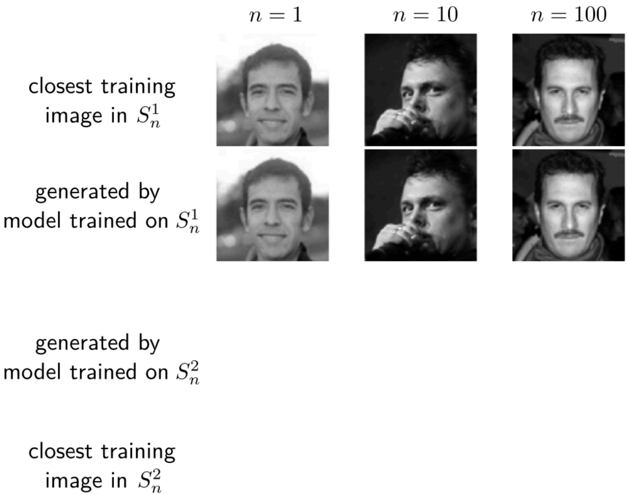
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



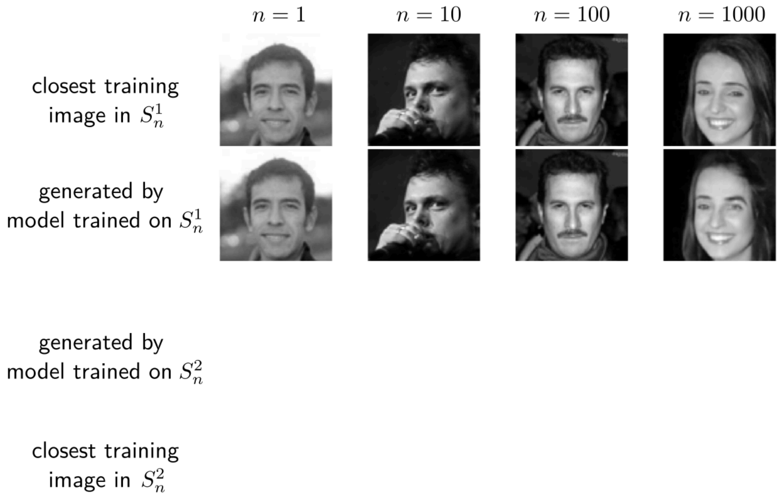
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



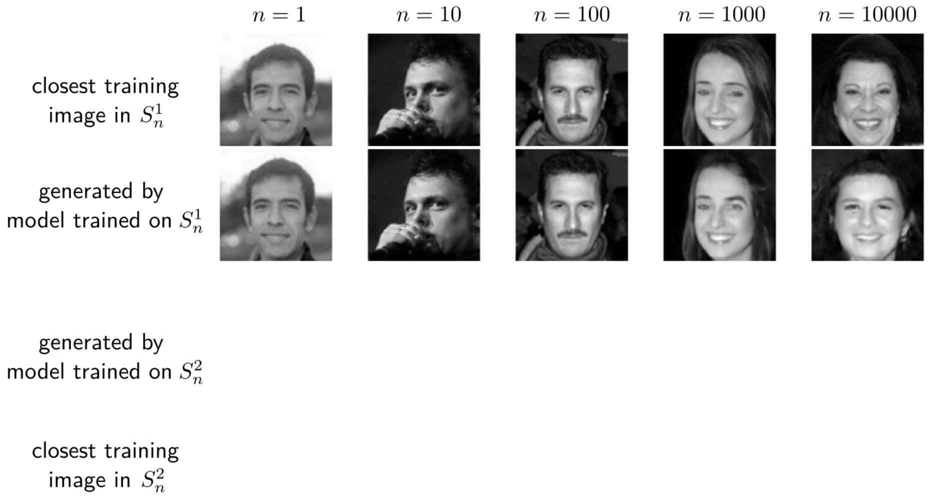
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



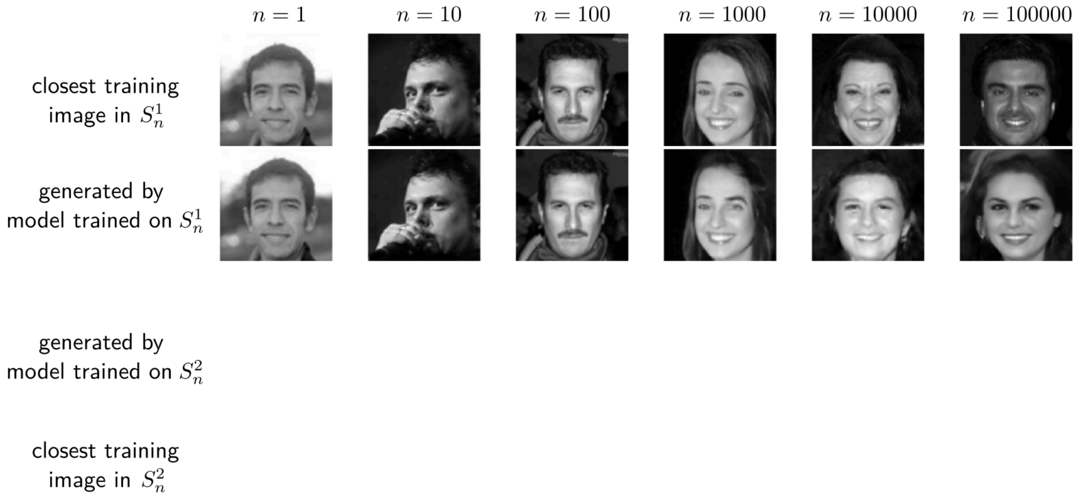
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



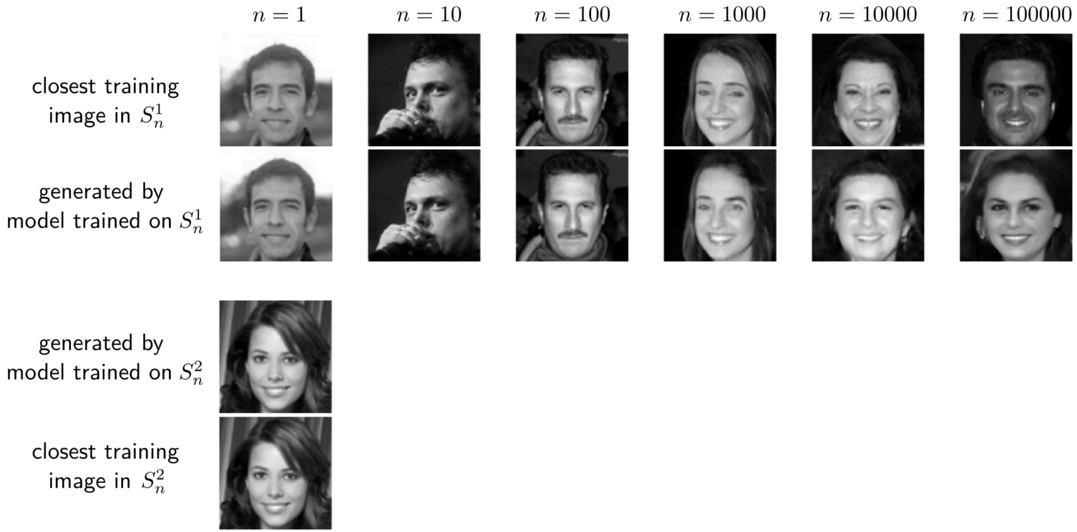
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



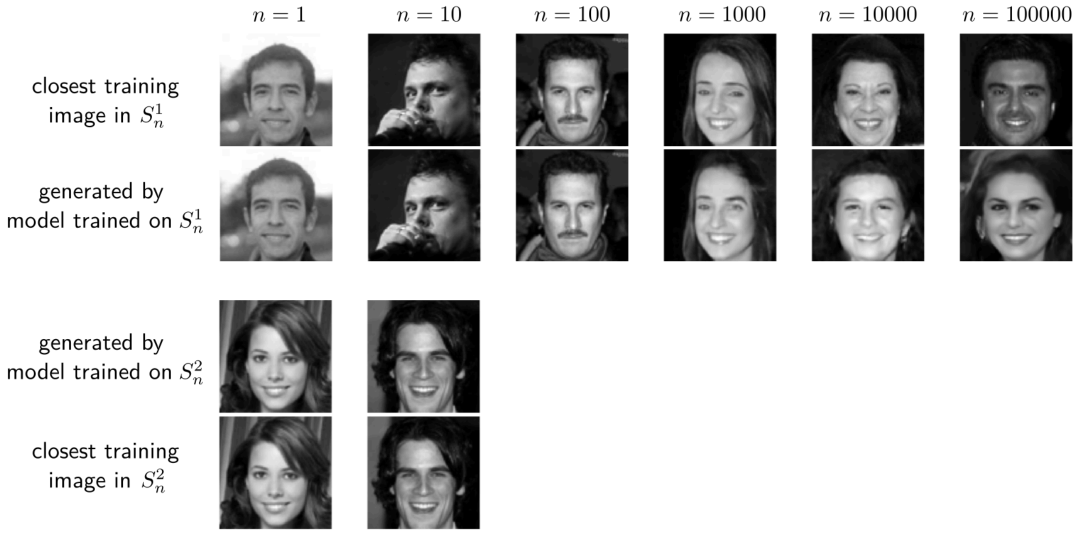
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



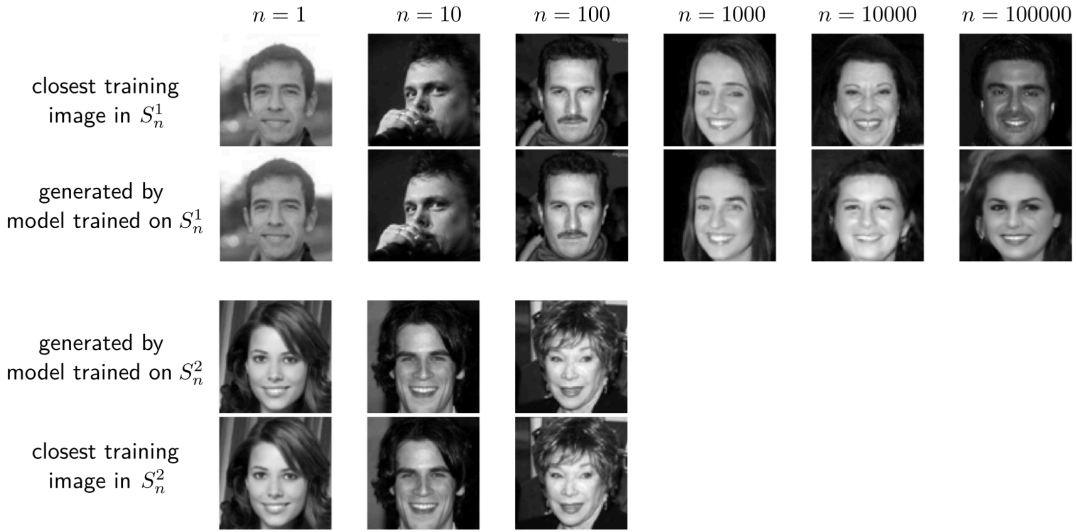
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



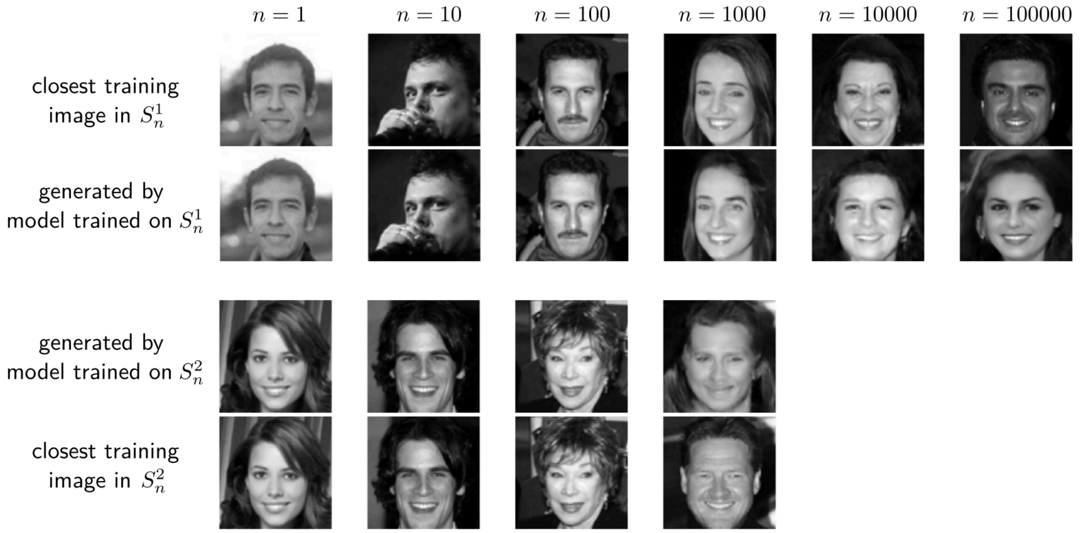
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



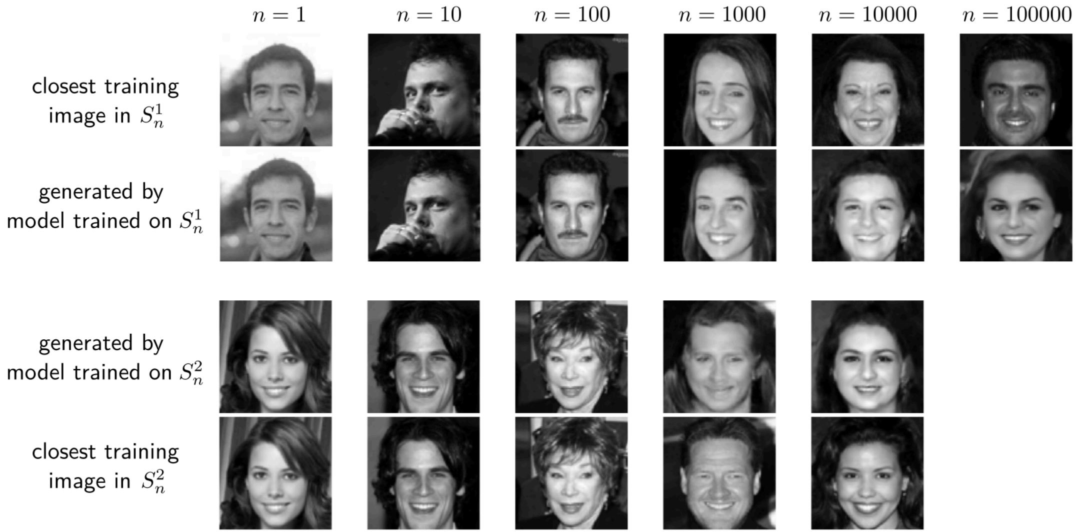
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



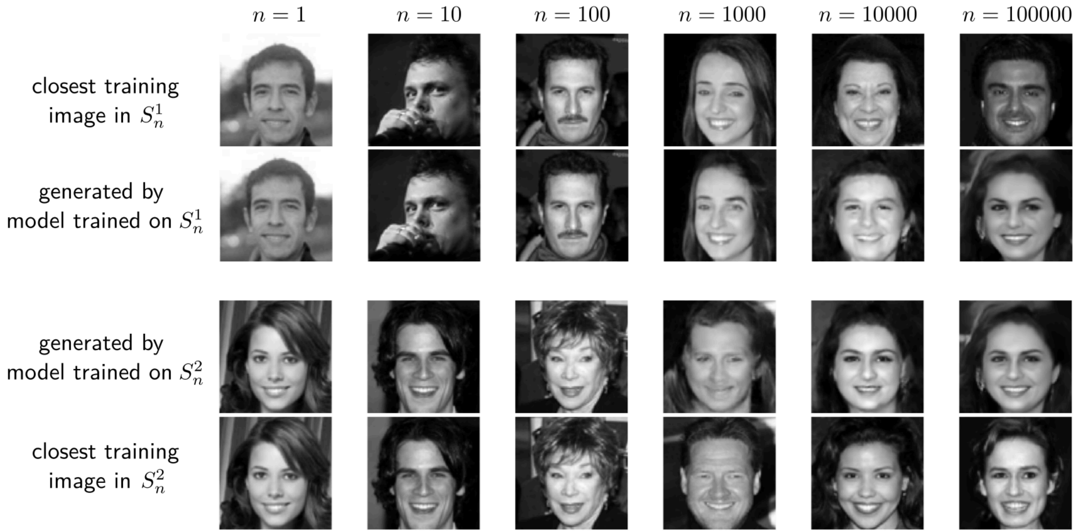
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
    - 2 **disjoints** sets S_n^1 and S_n^2
    - of size n
- For training set sample size n
    - Train a model on set S_n^1
    - Train a model on set S_n^2
    - Generate an image with each model using a common initialization x_0



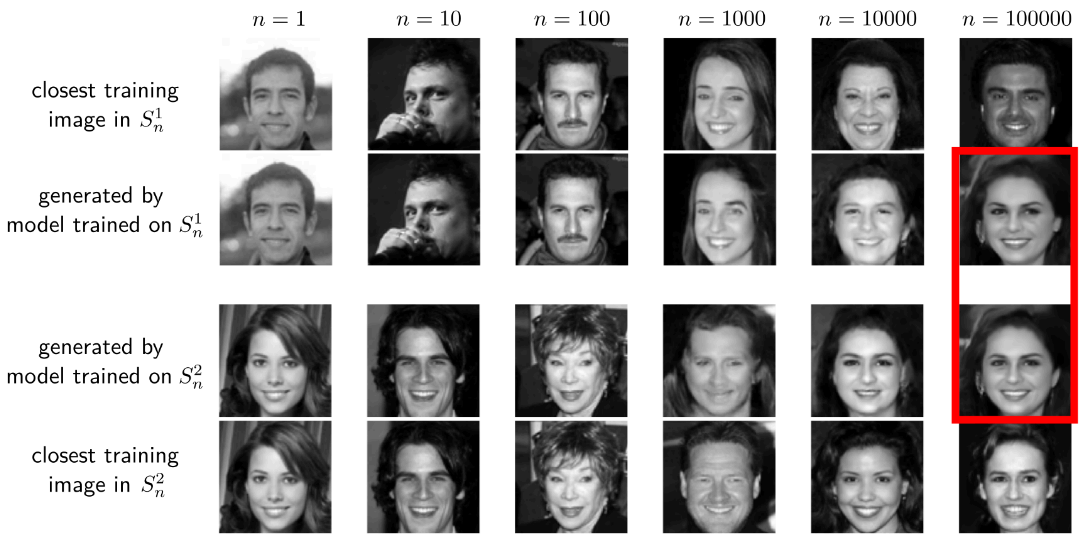
[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

From memorization to generalization

Memorization and closed-form minimizers observed since 2023[2,3]

Seminal experiment in 2024[4]:

- Split the training set
  - 2 **disjoints** sets S_n^1 and S_n^2
  - of size n
- For training set sample size n
  - Train a model on set S_n^1
  - Train a model on set S_n^2
  - Generate an image with each model using a common initialization x_0



---

[2] N. Carlini et al., **Extracting Training Data from Diffusion Models,** In: USENIX Security 23, 2023.
[3] X. Gu et al., **On memorization in diffusion models,** In: TMLR, 2023.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.

Empirical results: take away

Empirical results: take away

- In theory:

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality
- In practice:

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality
- In practice:
  - Models create good new samples (generalization)

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality
- In practice:
  - Models create good new samples (generalization)
  - 10k training samples are enough

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality
- In practice:
  - Models create good new samples (generalization)
  - 10k training samples are enough

How to resolve this incompatibility?

Empirical results: take away

- In theory:
  - Loss minimizer only generates training data (memorization)
  - Curse of dimensionality
- In practice:
  - Models create good new samples (generalization)
  - 10k training samples are enough

How to resolve this incompatibility?

Additional mystery: models generate the same data!

Memorization and generalization in Flow matching and Diffusion

What is memorization?

What is memorization?

- Commonly accepted definition: generating samples too similar to training data

What is memorization?

- Commonly accepted definition: generating samples too similar to training data

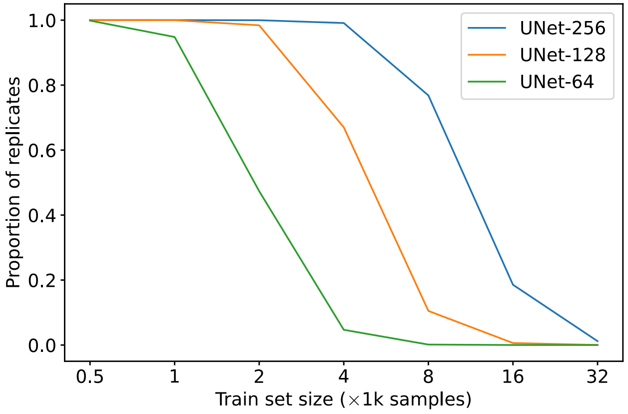
- Relatively easy to quantify: a generated sample is *memorized* if "its distance to the nearest training sample is smaller than 1/3 of that to the second nearest training sample"[5]

  the distance used and factor 1/3 can vary

[5] T. Yoon et al., **Diffusion probabilistic models generalize when they fail to memorize,** In: ICML 2023 workshop on structured probabilistic inference and generative modeling, 2023.

What is memorization?

- Commonly accepted definition: generating samples too similar to training data
- Relatively easy to quantify: a generated sample is *memorized* if "its distance to the nearest training sample is smaller than 1/3 of that to the second nearest training sample"[5]
  the distance used and factor 1/3 can vary
- Memorization ratio: generate many samples and measure proportions of memorized generated samples



Yoon et al. (2023)

[5] T. Yoon et al., **Diffusion probabilistic models generalize when they fail to memorize,** In: ICML 2023 workshop on structured probabilistic inference and generative modeling, 2023.

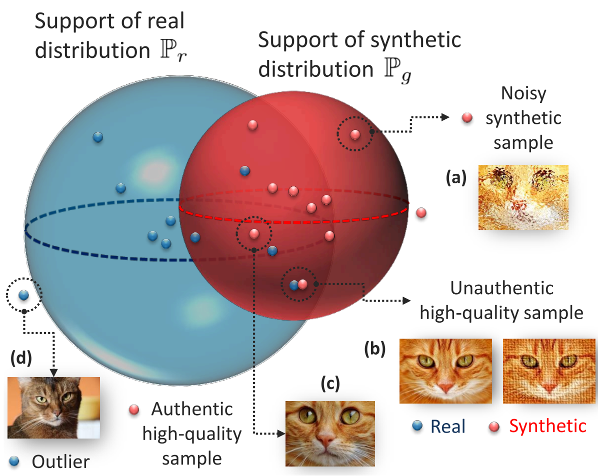
And what is generalization?

And what is generalization?

- Generating new data that *looks like* training data… but is not too close to it

And what is generalization?

- Generating new data that *looks like* training data… but is not too close to it
- Much harder to quantify, especially with a single metric. Want:
  - Fidelity
  - Diversity
  - Novelty

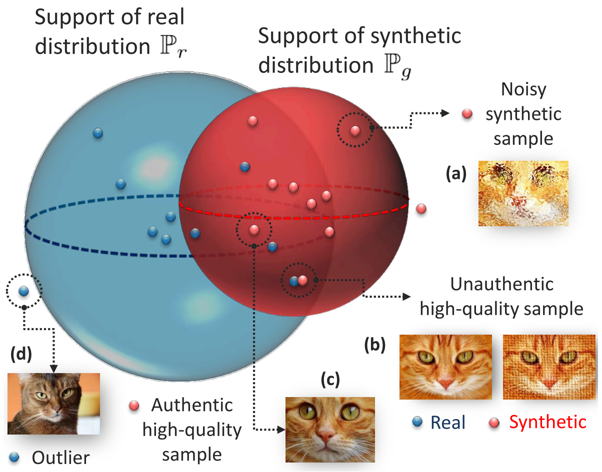


Alaa et al. (2022)

And what is generalization?

- Generating new data that *looks like* training data… but is not too close to it
- Much harder to quantify, especially with a single metric. Want:
  - Fidelity
  - Diversity
  - Novelty
- For images, commonly measured with Fréchet Inception Distance[6] (FID) despite several known limitations[7]

many complements/alternatives[9,10,11] to FID that we cannot cover…



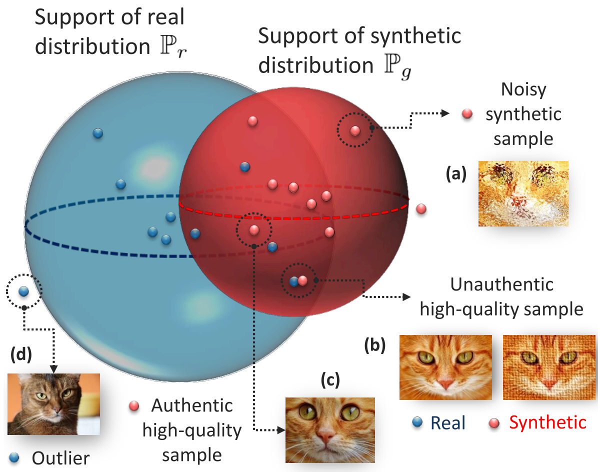
Alaa et al. (2022)

[6] M. Heusel et al., **Gans trained by a two time-scale update rule converge to a local nash equilibrium,** In: NeurIPS, 2017.
[7] G. Stein et al., **Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models,** In: NeurIPS, 2023.
[5] T. Yoon et al., **Diffusion probabilistic models generalize when they fail to memorize,** In: ICML 2023 workshop on structured probabilistic inference and generative modeling, 2023.
[8] S. Buchanan et al., **On the edge of memorization in diffusion models,** In: NeurIPS, 2025.
[9] M. Sajjadi et al., **Assessing generative models via precision and recall,** In: NeurIPS, 2018.
[10] T. Kynkaanniemi et al., **Improved precision and recall metric for assessing generative models,** In: NeurIPS, 2019.
[11] M. Jiralerspong et al., **Feature likelihood divergence: evaluating the generalization of generative models using samples,** In: NeurIPS, 2023.

And what is generalization?

- Generating new data that *looks like* training data… but is not too close to it
- Much harder to quantify, especially with a single metric. Want:
  - Fidelity
  - Diversity
  - Novelty
- For images, commonly measured with Fréchet Inception Distance[6] (FID) despite several known limitations[7]
- Memorization and generalization are often treated as a dichotomy[5,8]

many complements/alternatives[9,10,11] to FID that we cannot cover…



Alaa et al. (2022)

Diffusion Probabilistic Models Generalize when They Fail to Memorize

TaeHo Yoon[1] Joo Young Choi[1] Sehyun Kwon[2] Ernest K. Ryu[1,2]

Abstract

In this work, we study the training of diffusion probabilistic models through a series of hypotheses and carefully designed experiments. We call our key finding the memorization-generalization dichotomy, and it asserts that generalization and memorization are mutually exclusive phenomena.

[6] M. Heusel et al., **Gans trained by a two time-scale update rule converge to a local nash equilibrium,** In: NeurIPS, 2017.
[7] G. Stein et al., **Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models,** In: NeurIPS, 2023.
[5] T. Yoon et al., **Diffusion probabilistic models generalize when they fail to memorize,** In: ICML 2023 workshop on structured probabilistic inference and generative modeling, 2023.
[8] S. Buchanan et al., **On the edge of memorization in diffusion models,** In: NeurIPS, 2025.
[9] M. Sajjadi et al., **Assessing generative models via precision and recall,** In: NeurIPS, 2018.
[10] T. Kynkaanniemi et al., **Improved precision and recall metric for assessing generative models,** In: NeurIPS, 2019.
[11] M. Jiralerspong et al., **Feature likelihood divergence: evaluating the generalization of generative models using samples,** In: NeurIPS, 2023.

Where does generalization come from?

Hypotheses for main driver of generalization:

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
    - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

Conflicting points of view:

- Inductive bias:

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

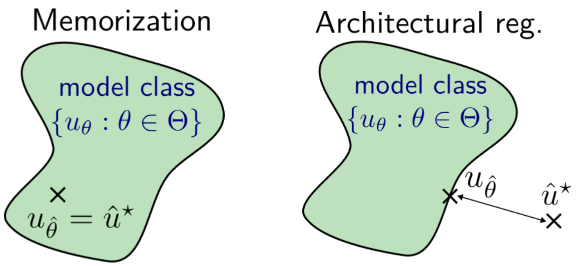
Conflicting points of view:

- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture

---

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?



Hypotheses for main driver of generalization:

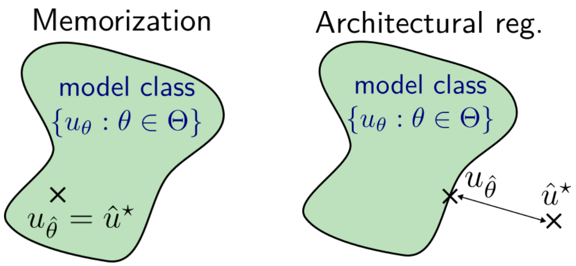
- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

Conflicting points of view:

- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture
  - Limits the capacity of the network to learn the ideal score/velocity

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?



Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

Conflicting points of view:

- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture
  - Limits the capacity of the network to learn the ideal score/velocity
- Optimization dynamics:

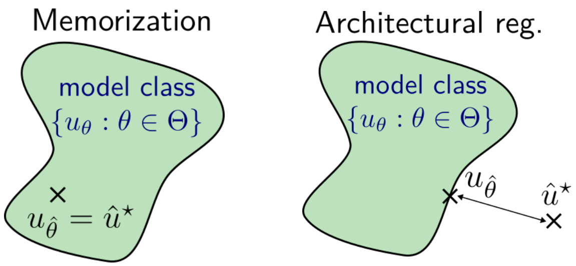
---

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties



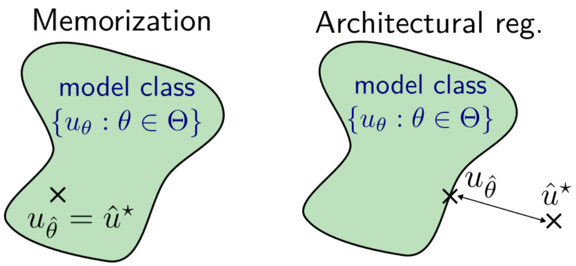
Conflicting points of view:

- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture
  - Limits the capacity of the network to learn the ideal score/velocity
- Optimization dynamics:
  - The network has enough capacity to learn the ideal velocity

---

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?



Hypotheses for main driver of generalization:

- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

Conflicting points of view:

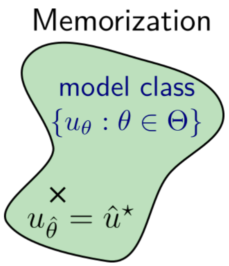
- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture
  - Limits the capacity of the network to learn the ideal score/velocity
- Optimization dynamics:
  - The network has enough capacity to learn the ideal velocity
  - Would memorize the training samples

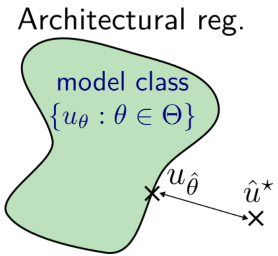
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Where does generalization come from?

Hypotheses for main driver of generalization:

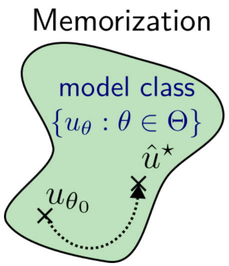
- **Inductive bias** of the architecture[4,12,13]
  - Idea: regularization through the architecture (CNN)
- Regularization from the **optimization dynamics**[14,15]
  - Idea: gradient descent yields solutions with specific properties

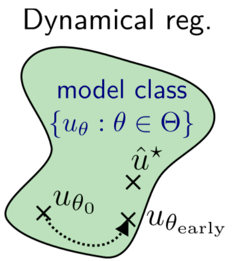




Conflicting points of view:

- Inductive bias:
  - The network explicitly regularizes the velocity with the architecture
  - Limits the capacity of the network to learn the ideal score/velocity
- Optimization dynamics:
  - The network has enough capacity to learn the ideal velocity
  - Would memorize the training samples
  - Works because of gradient descent + early stopping

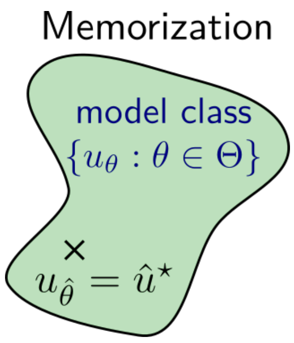




[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

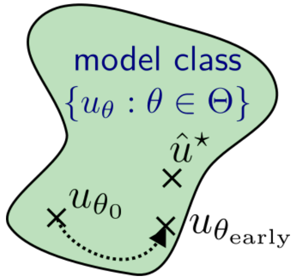
Dynamical and architectural regularization

Inductive bias and optimization

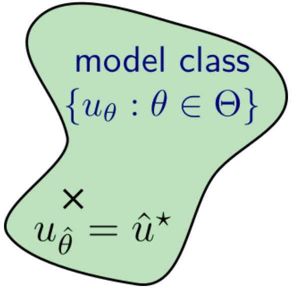


Inductive bias and optimization

Dynamical reg.

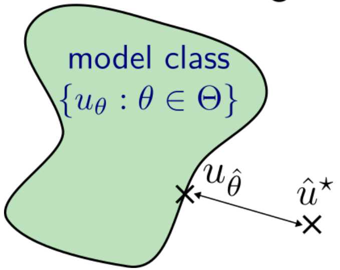


Memorization

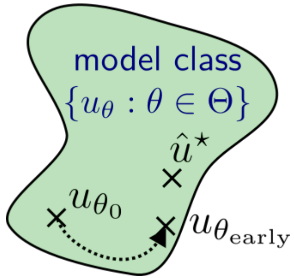


Inductive bias and optimization

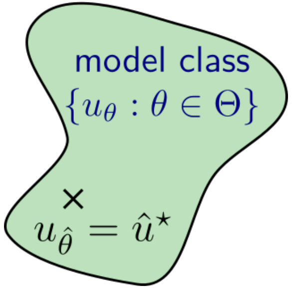
Architectural reg.



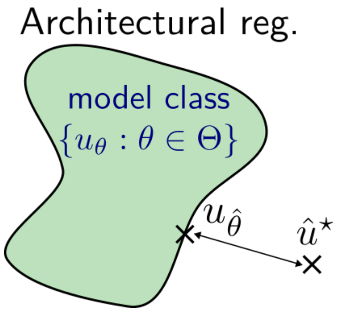
Dynamical reg.

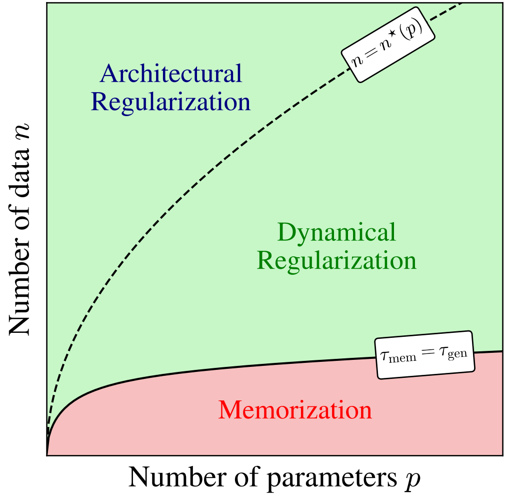


Memorization

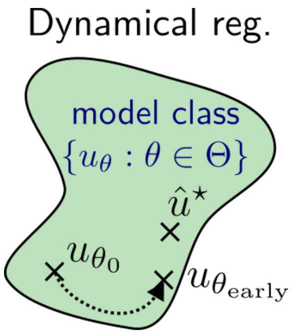


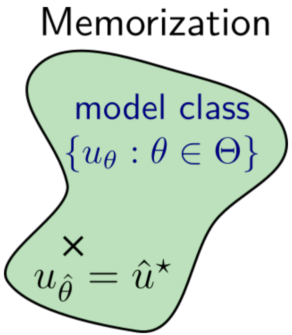
Inductive bias and optimization





Bonnaire et al. (2026)



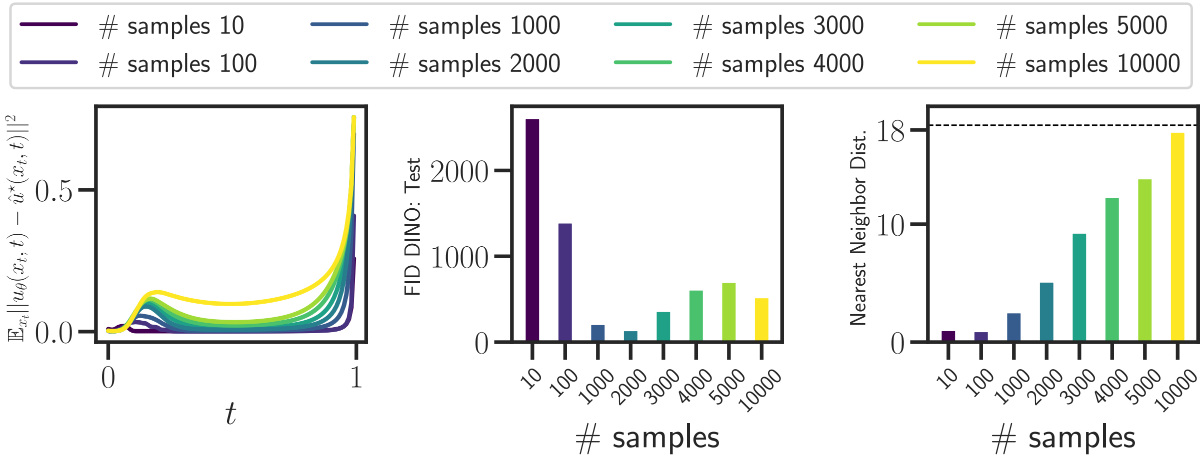


Dynamical[14,15] then Architectural[4,12,13]?

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[4] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[12] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[13] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Architectural regularization[16]

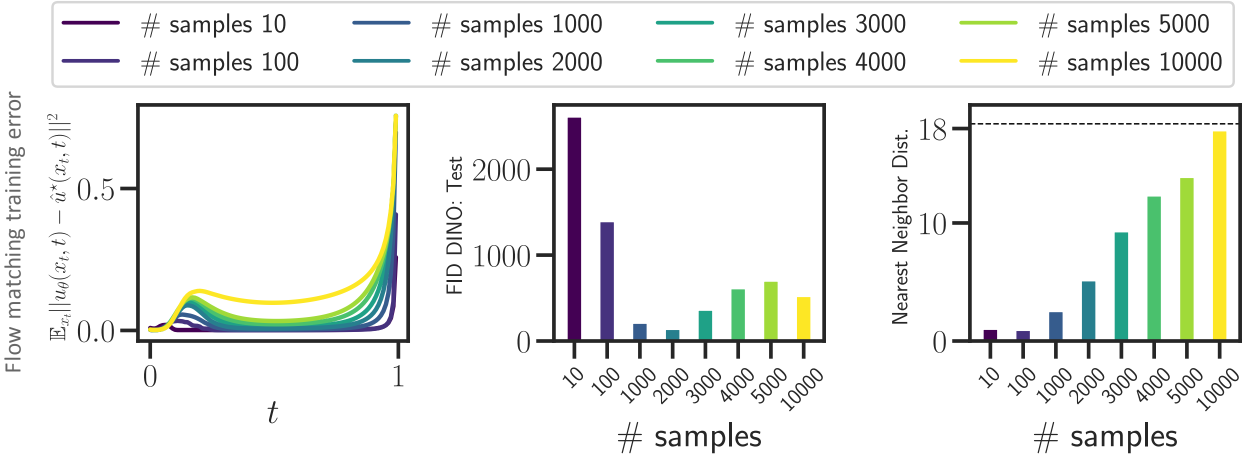
- Fixed architecture size, increase number of training samples n (CIFAR-10)



[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

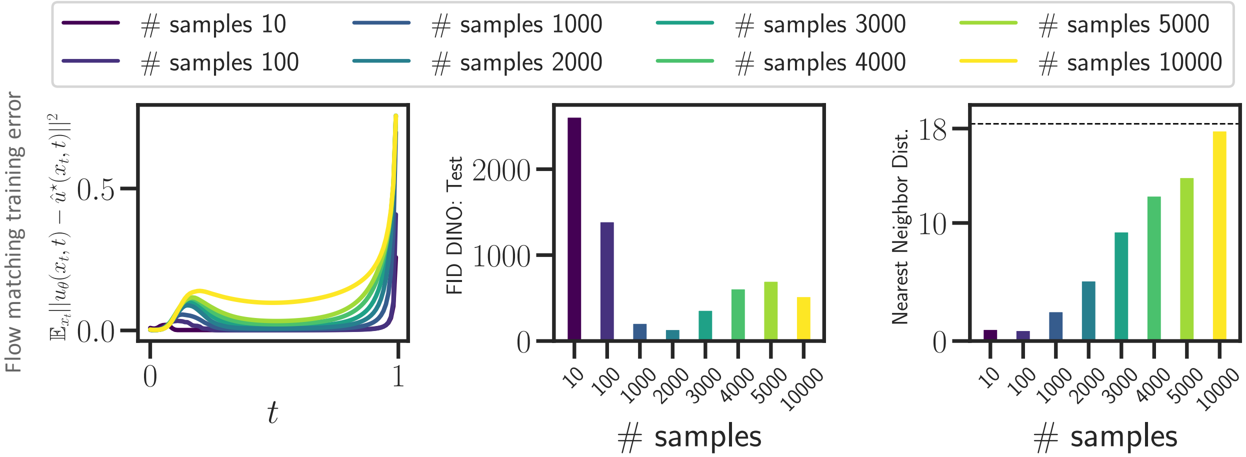
- Fixed architecture size, increase number of training samples n (CIFAR-10)



[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

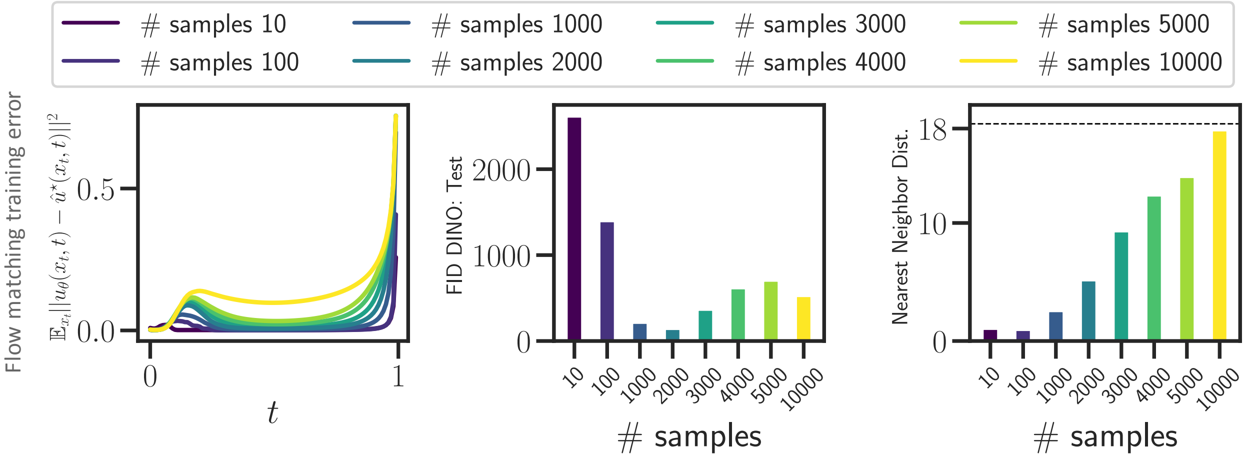
- Fixed architecture size, increase number of training samples n (CIFAR-10)



[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

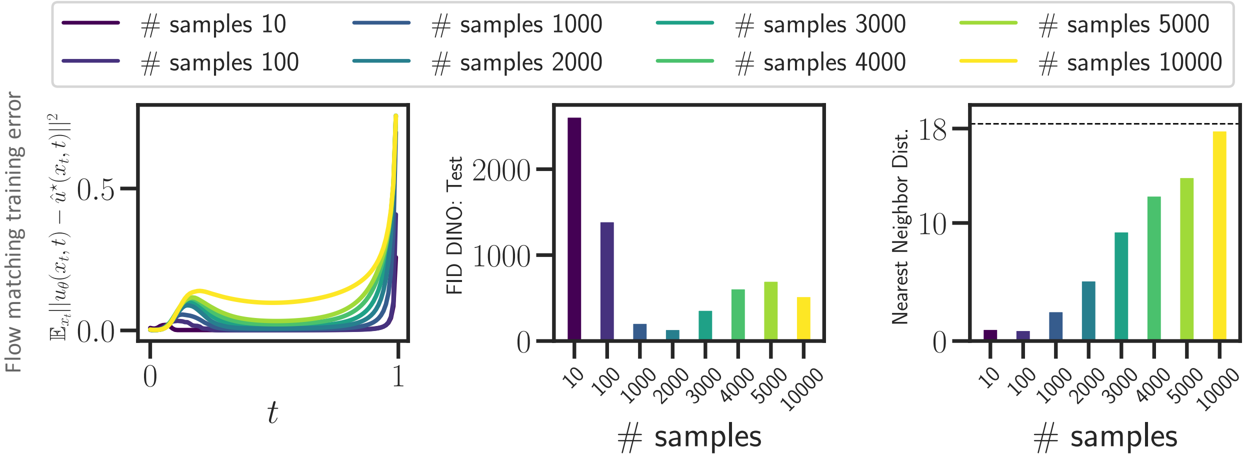
- Fixed architecture size, increase number of training samples n (CIFAR-10)



[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

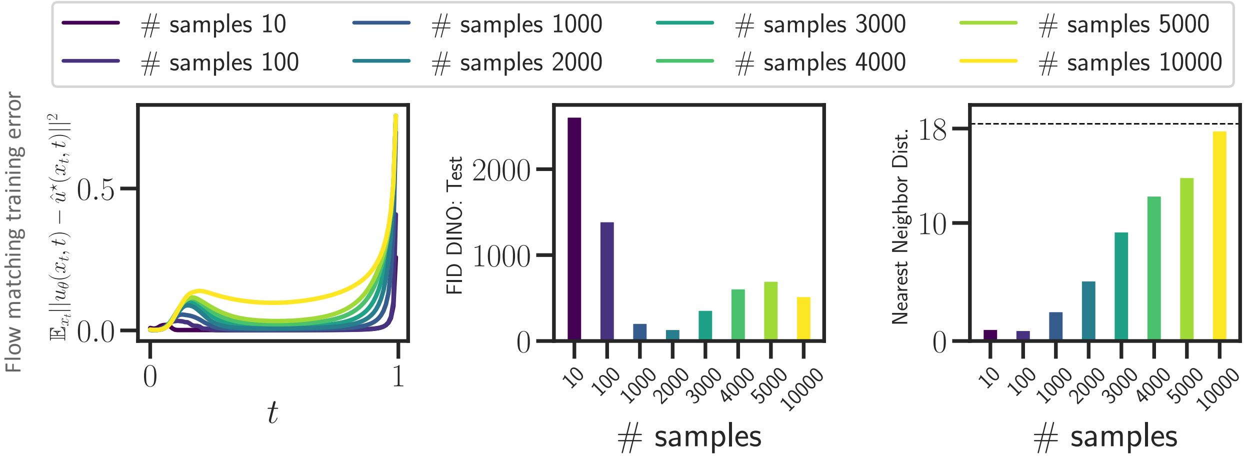
- Fixed architecture size, increase number of training samples n (CIFAR-10)



[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

- Fixed architecture size, increase number of training samples n (CIFAR-10)

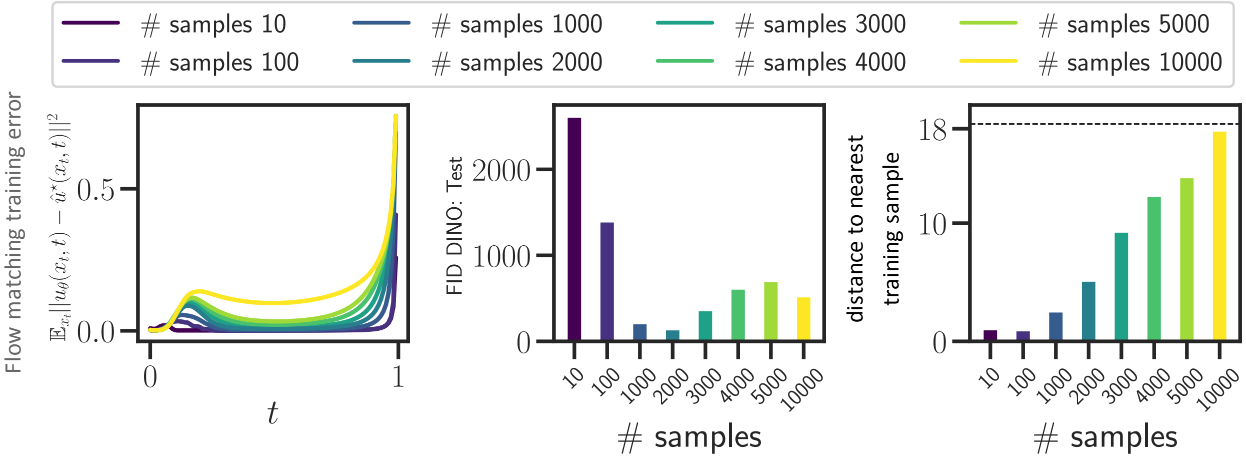


(FID going down at $n = 1000$ is a problem of FID)

[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

- Fixed architecture size, increase number of training samples n (CIFAR-10)

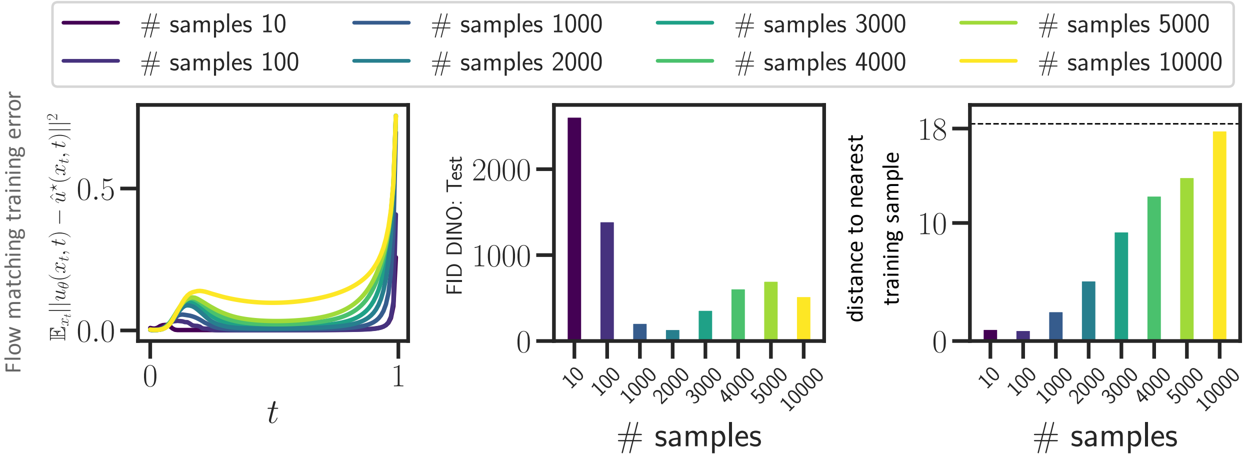


(FID going down at $n = 1000$ is a problem of FID)

[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Architectural regularization[16]

- Fixed architecture size, increase number of training samples n (CIFAR-10)



(FID going down at $n = 1000$ is a problem of FID)

Models stop memorizing when they can no longer fit $\hat{u}^\star$

[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Importance of small times[16]

From a good trained u_θ , we build a **hybrid model** (fixed $\tau \in [0,1]$):

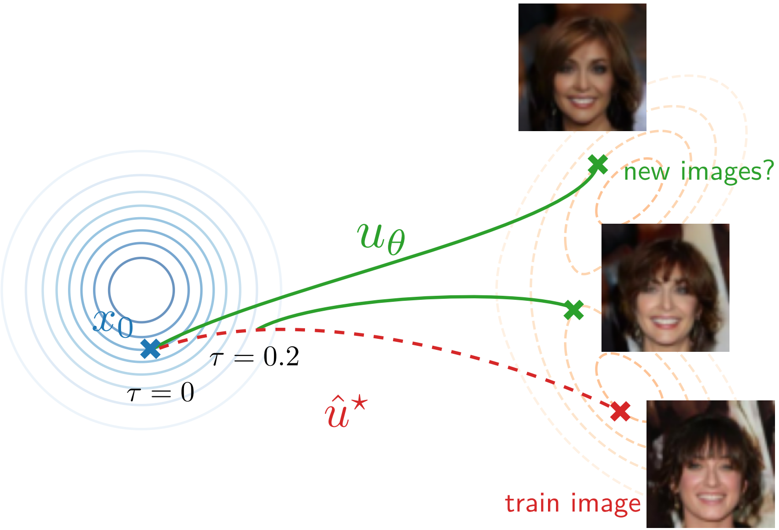
- on $[0,\tau]$: follow $\hat{u}^\star$
- on $[\tau,1]$: follow u_θ
- $\tau = 1$ means completely following $\hat{u}^\star$ (full memorization)
- $\tau = 0$ means completely following u_θ (good generalization)

[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Importance of small times[16]

From a good trained u_θ , we build a **hybrid model** (fixed $\tau \in [0,1]$):

- on $[0,\tau]$: follow $\hat{u}^\star$
- on $[\tau,1]$: follow u_θ
- $\tau = 1$ means completely following $\hat{u}^\star$ (full memorization)
- $\tau = 0$ means completely following u_θ (good generalization)

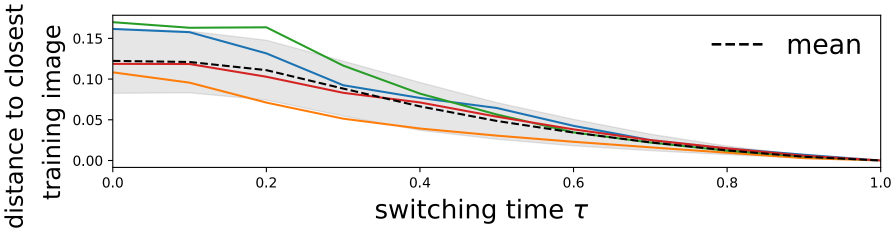


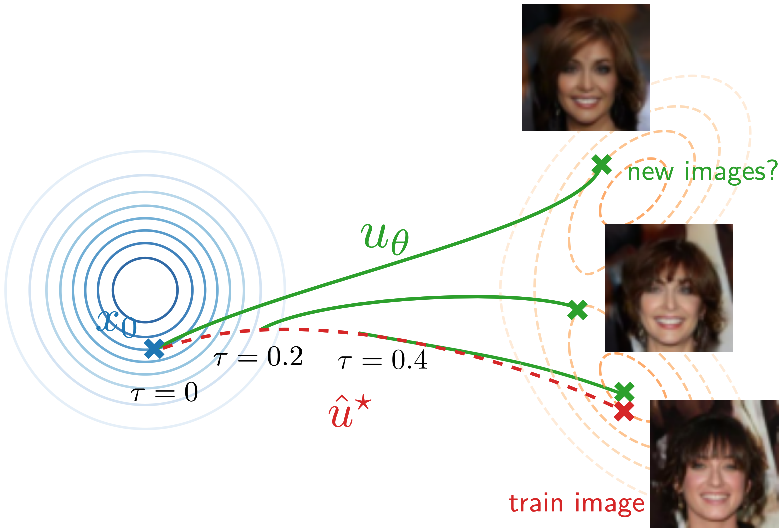
[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Importance of small times[16]

From a good trained u_θ , we build a **hybrid model** (fixed $\tau \in [0,1]$):

- on $[0,\tau]$: follow $\hat{u}^\star$
- on $[\tau,1]$: follow u_θ
- $\tau = 1$ means completely following $\hat{u}^\star$ (full memorization)
- $\tau = 0$ means completely following u_θ (good generalization)



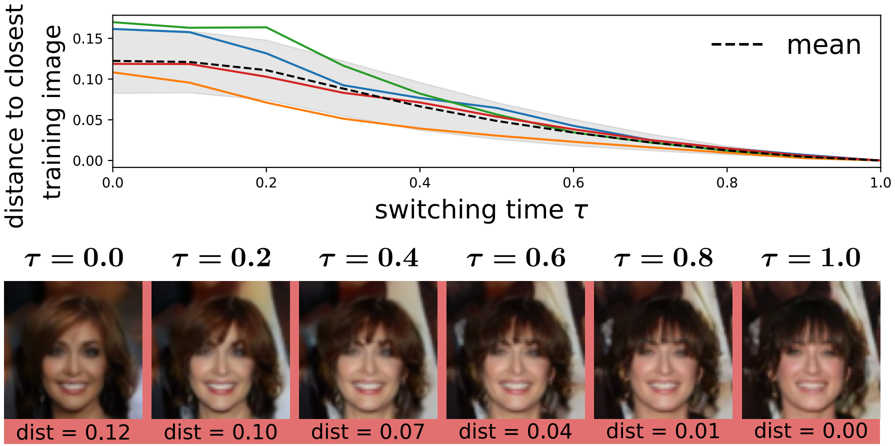


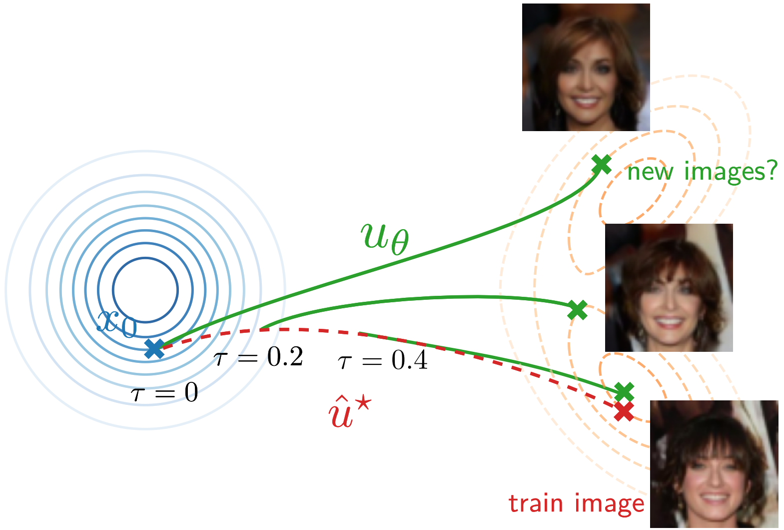
[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Importance of small times[16]

From a good trained u_θ, we build a **hybrid model** (fixed $\tau \in [0,1]$):

- on $[0,\tau]$: follow $\hat{u}^\star$
- on $[\tau,1]$: follow u_θ
- $\tau = 1$ means completely following $\hat{u}^\star$ (full memorization)
- $\tau = 0$ means completely following u_θ (good generalization)





Not fitting small times is critical

[16] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

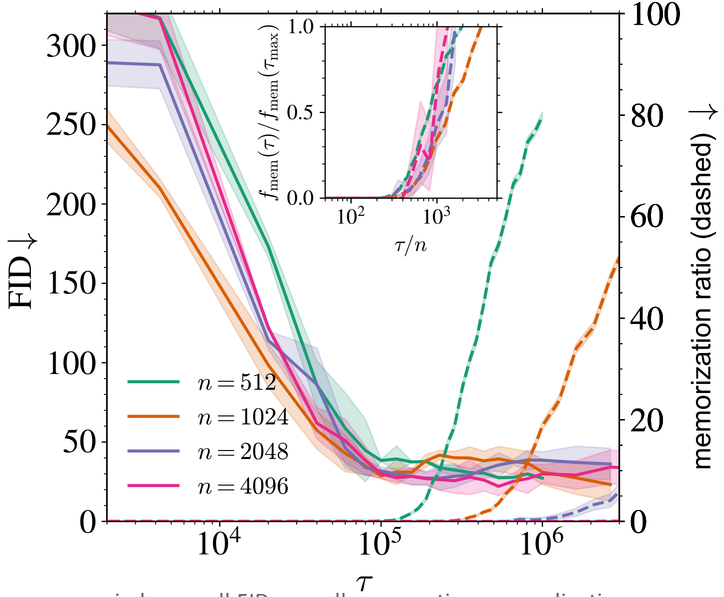
Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)



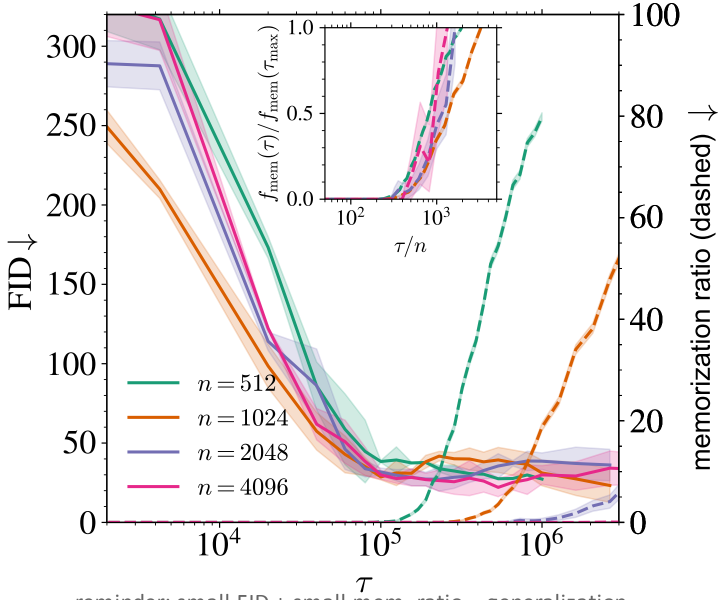
reminder: small FID + small mem. ratio = generalization

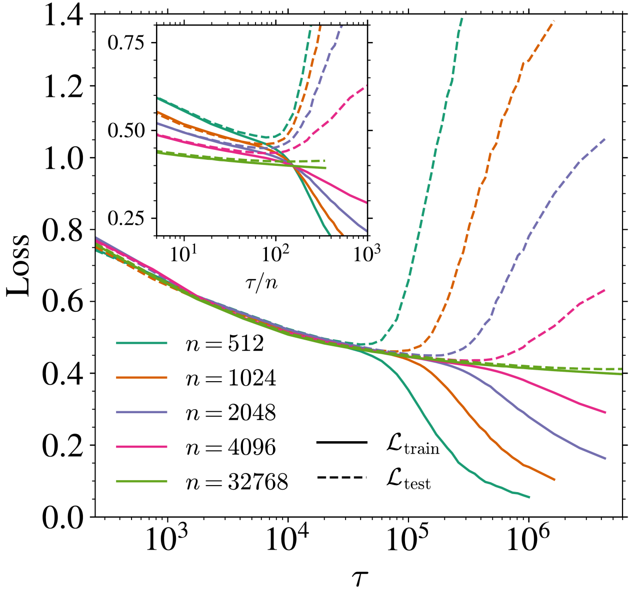
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.

[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)



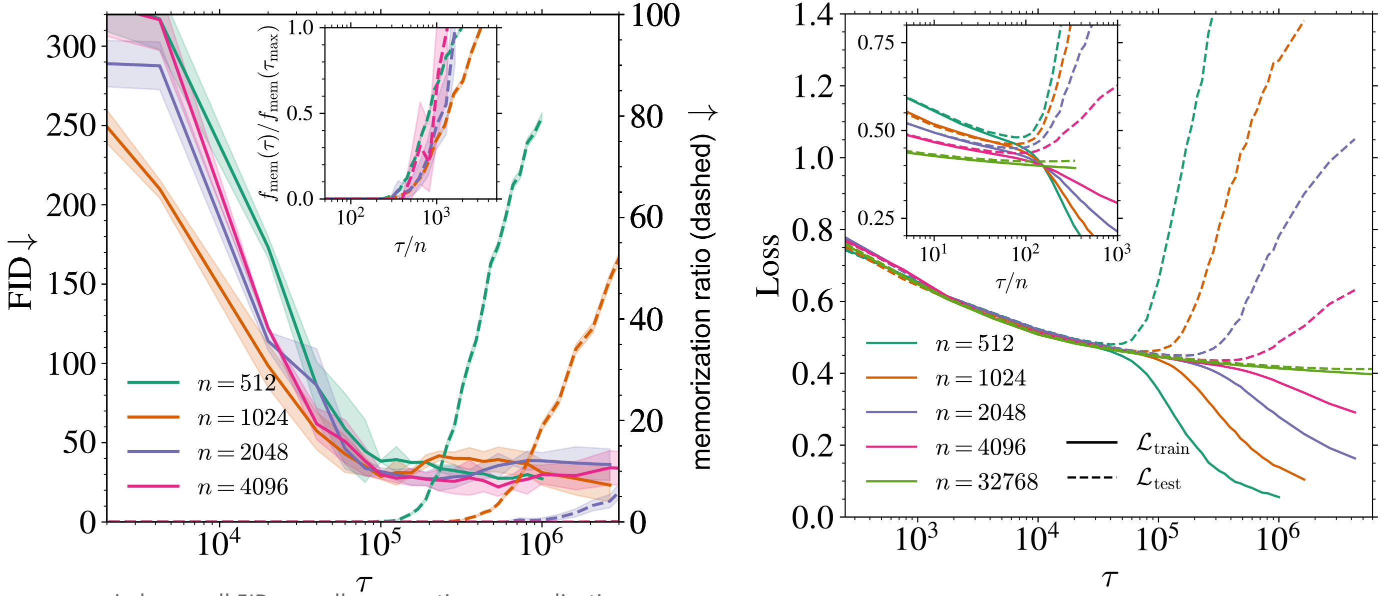


reminder: small FID + small mem. ratio = generalization

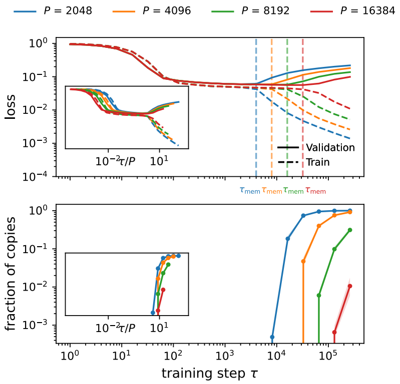
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)



reminder: small FID + small mem. ratio = generalization

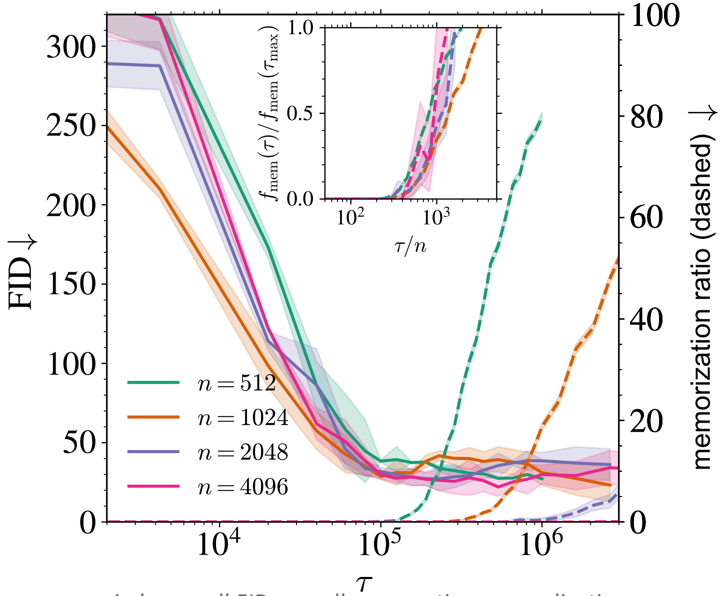


same result by Favero et al (2026)

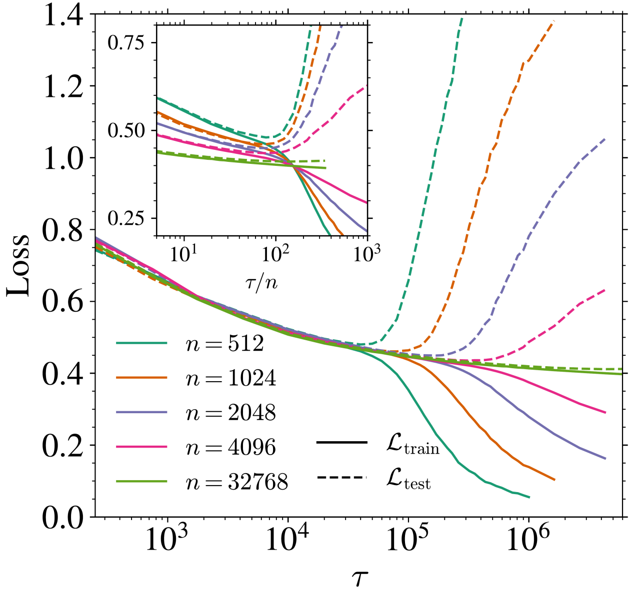
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

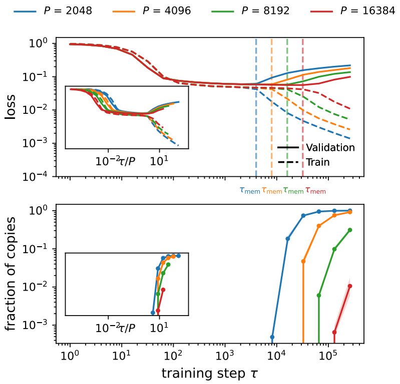
Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)



reminder: small FID + small mem. ratio = generalization





same result by Favero et al (2026)

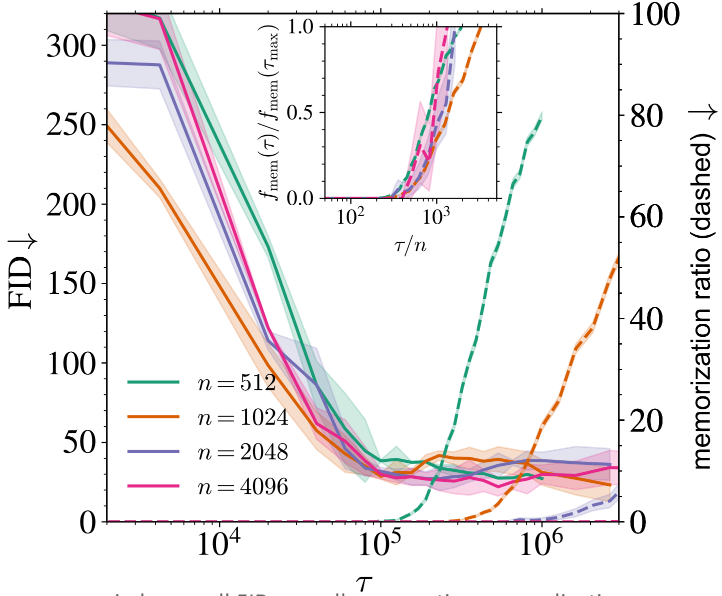
Overparametrized models can still generalize if early stopped

---

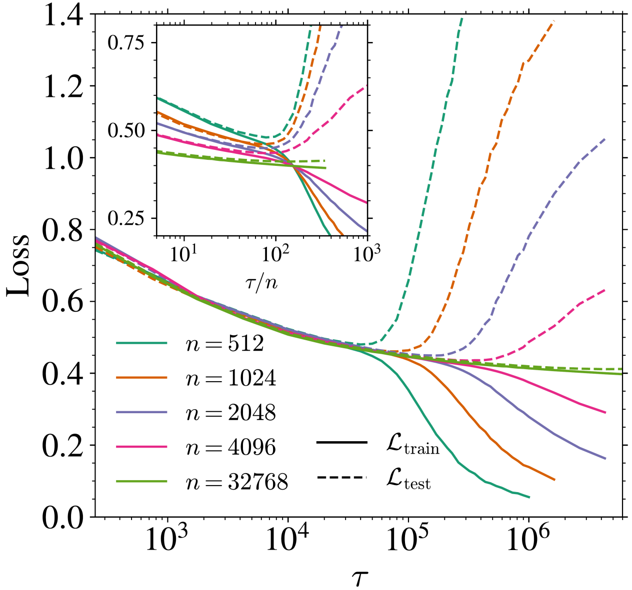
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

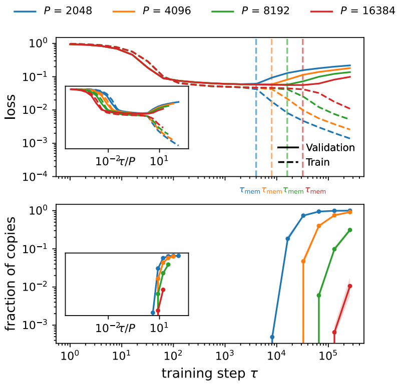
Dynamical regularization[14,15]

- Fixed architecture size, increase number of training samples n (CelebA 32×32)



reminder: small FID + small mem. ratio = generalization





same result by Favero et al (2026)

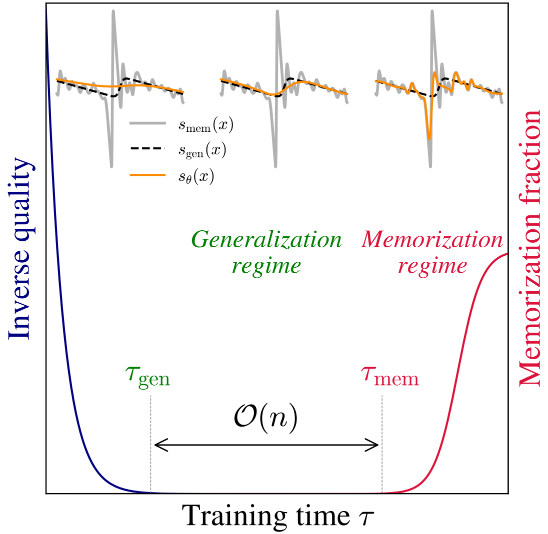
Memorization starts at $\tau_{\text{mem}} = \mathcal{O}(\#\text{samples})$

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

Generalization and memorization times[14,15]

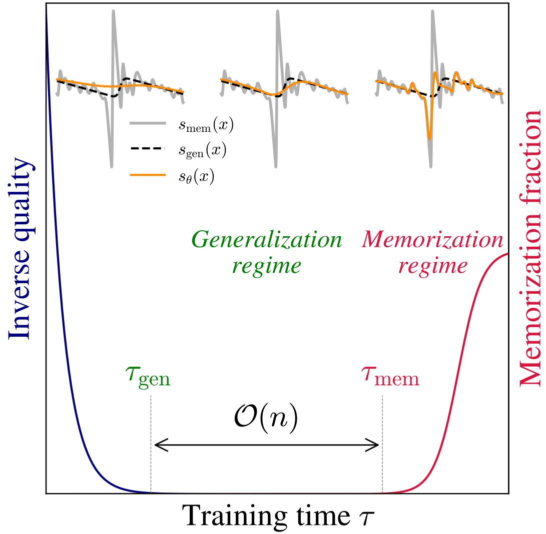
[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

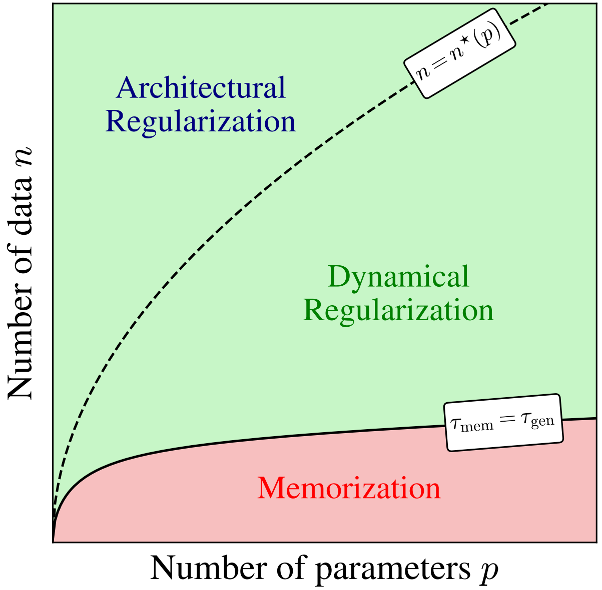
Generalization and memorization times[14,15]



[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

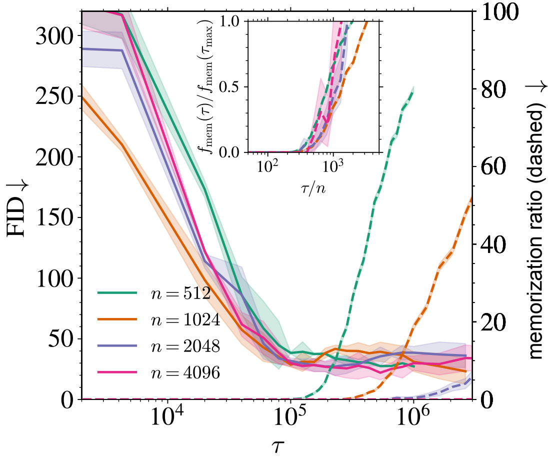
Generalization and memorization times[14,15]





[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[15] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.

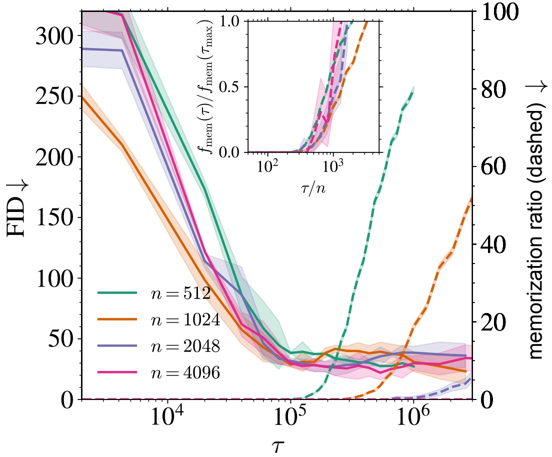
Caveats in overparametrization and early stopping[14]



Data & architecture. We conduct our experiments on the CelebA face dataset [26], which we convert to grayscale downsampled images of size $d = 32 \times 32$, and vary the training set size n from 128 up to 32768. Our score model has a U-Net architecture [35] with three resolution levels, and a base channel

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.

Caveats in overparametrization and early stopping[14]

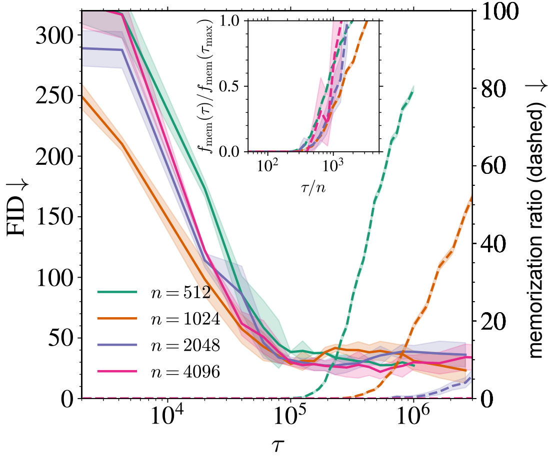


Data & architecture. We conduct our experiments on the CelebA face dataset [26], which we convert to grayscale downsampled images of size $d = 32 \times 32$, and vary the training set size n from 128 up to 32768. Our score model has a U-Net architecture [35] with three resolution levels, and a base channel

- In this experiment the dataset size is very reduced (both through n and d)

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.

Caveats in overparametrization and early stopping[14]

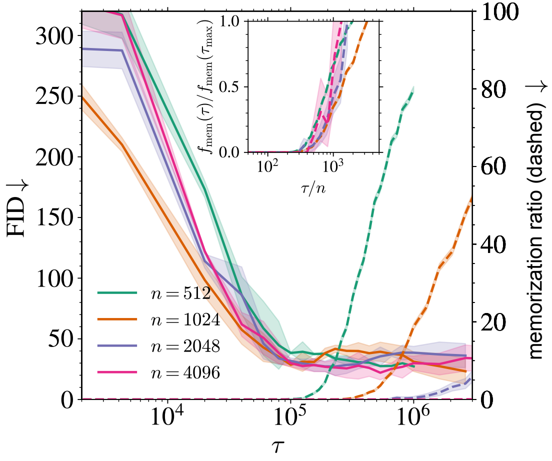


Data & architecture. We conduct our experiments on the CelebA face dataset [26], which we convert to grayscale downsampled images of size $d = 32 \times 32$, and vary the training set size n from 128 up to 32768. Our score model has a U-Net architecture [35] with three resolution levels, and a base channel

- In this experiment the dataset size is very reduced (both through n and d)
- Real CelebA has $178 \times 218 \times 3 \times 200000 \approx 2.3 \times 10^{10}$ pixels

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.

Caveats in overparametrization and early stopping[14]



Data & architecture. We conduct our experiments on the CelebA face dataset [26], which we convert to grayscale downsampled images of size $d = 32 \times 32$, and vary the training set size n from 128 up to 32768. Our score model has a U-Net architecture [35] with three resolution levels, and a base channel

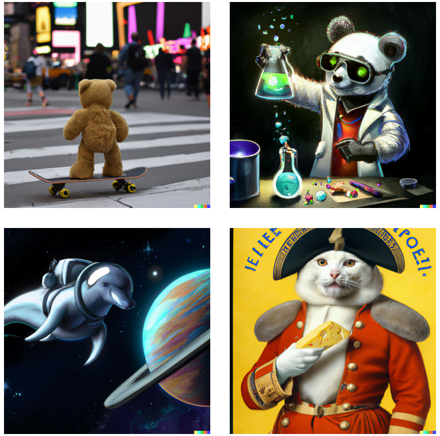
- In this experiment the dataset size is very reduced (both through n and d)
- Real CelebA has $178 \times 218 \times 3 \times 200000 \approx 2.3 \times 10^{10}$ pixels
- Open question: do models have enough capacity to memorize their datasets?

[14] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.

Conclusion

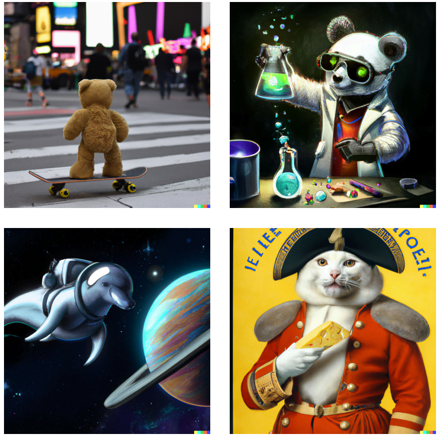



Conclusion

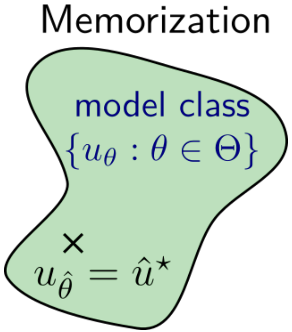


The generalization properties of flow matching/diffusion are poorly understood

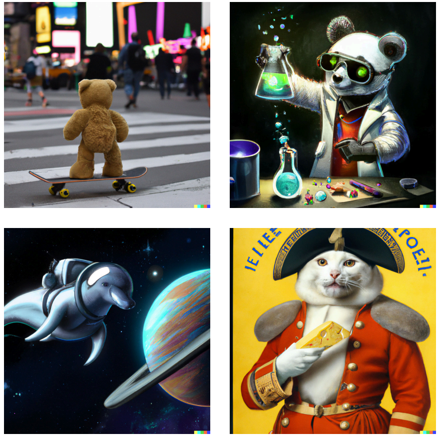
Conclusion

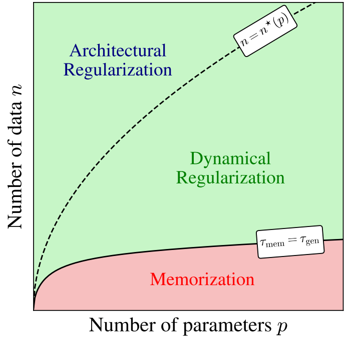


The generalization properties of flow matching/diffusion are poorly understood



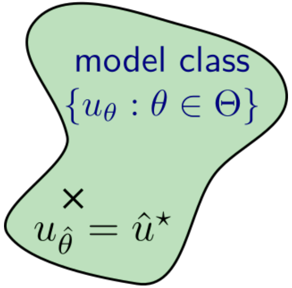
Conclusion





Bonnaire et al. (2026)

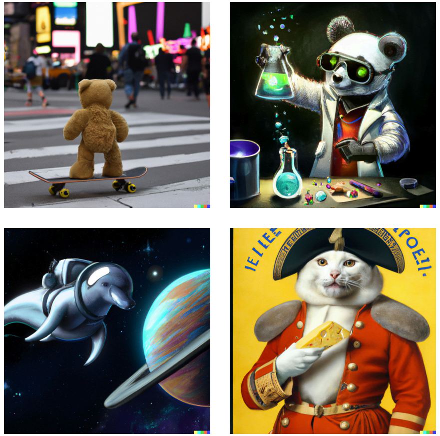
Memorization

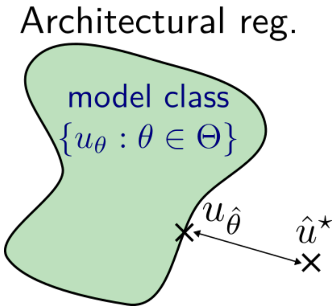


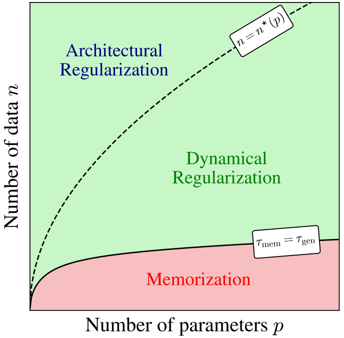
The generalization properties of flow matching/diffusion are poorly understood

- Mainly two hypotheses:

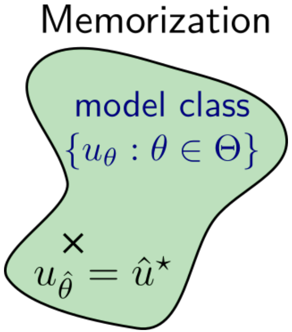
Conclusion







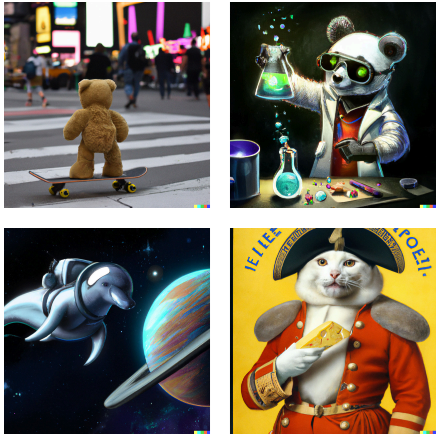
Bonnaire et al. (2026)



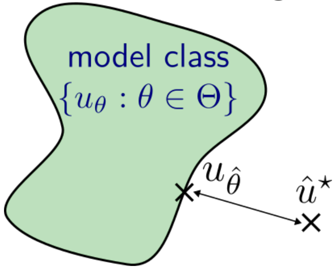
The generalization properties of flow matching/diffusion are poorly understood

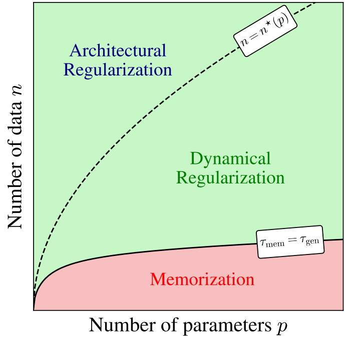
- Mainly two hypotheses:
  - Architecture regularization

Conclusion



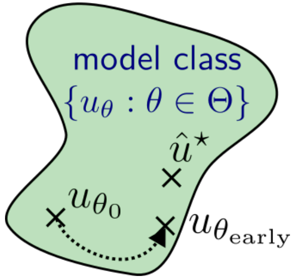
Architectural reg.



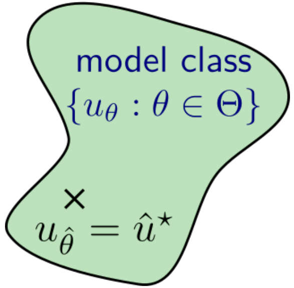


Bonnaire et al. (2026)

Dynamical reg.



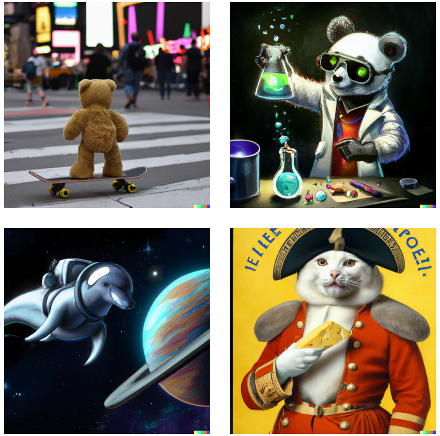
Memorization



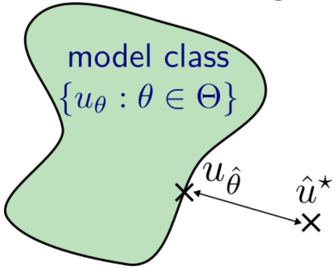
The generalization properties of flow matching/diffusion are poorly understood

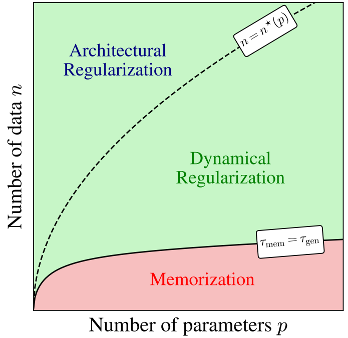
- Mainly two hypotheses:
  - Architecture regularization
  - Dynamical regularization through optimization

Conclusion



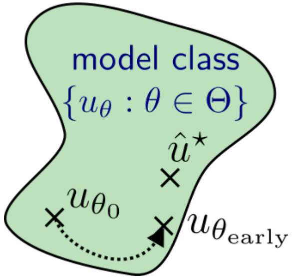
Architectural reg.



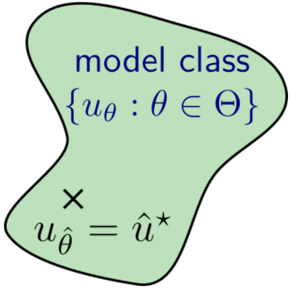


Bonnaire et al. (2026)

Dynamical reg.



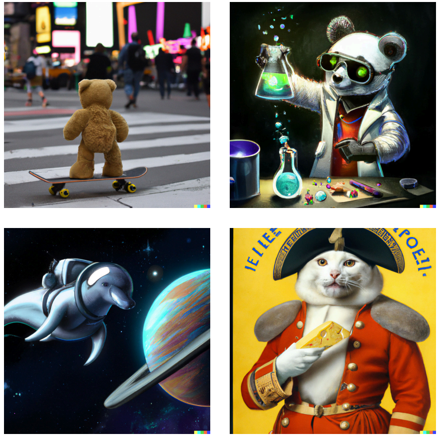
Memorization



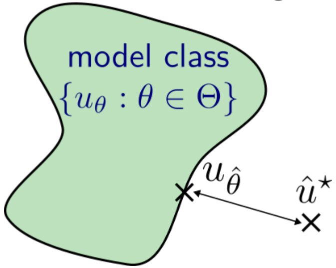
The generalization properties of flow matching/diffusion are poorly understood

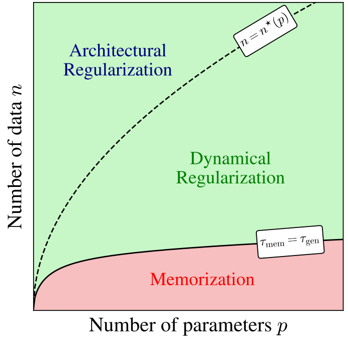
- Mainly two hypotheses:
  - Architecture regularization
  - Dynamical regularization through optimization
- Other source for the generalization

Conclusion



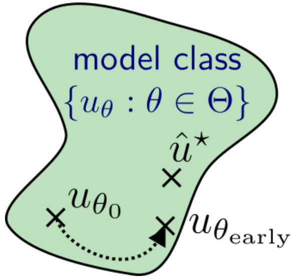
Architectural reg.



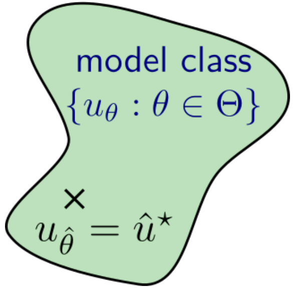


Bonnaire et al. (2026)

Dynamical reg.



Memorization

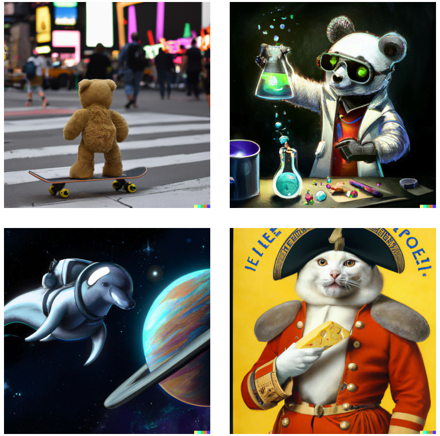


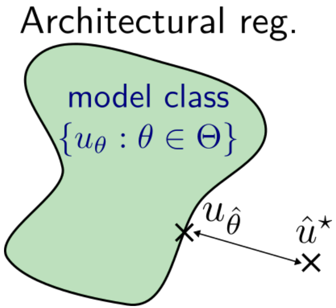
The generalization properties of flow matching/diffusion are poorly understood

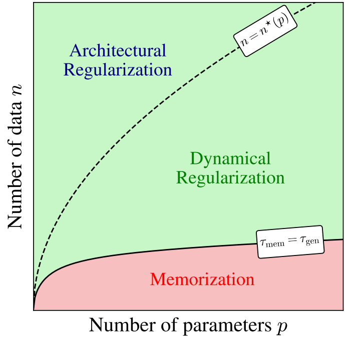
- Mainly two hypotheses:
  - Architecture regularization
  - Dynamical regularization through optimization
- Other source for the generalization
  - Stochasticity at inference/Errors in numerical ODE/SDE resolution[17,18]

[17] É. Pierret et al., **Diffusion Models for Gaussian Distributions: Exact Solutions and Wasserstein Errors,** In: ICML, 2025.
[18] S. Hurault et al., **From Score Matching to Diffusion: A Fine-Grained Error Analysis in the Gaussian Setting,** In: arXiv preprint arXiv:2503.11615, 2025.

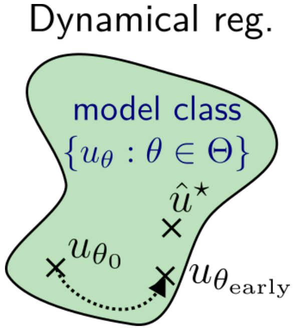
Conclusion

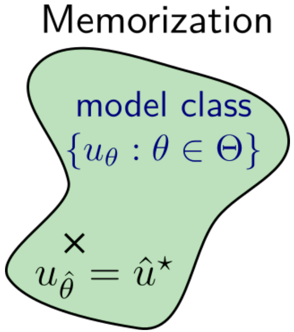






Bonnaire et al. (2026)





Thank you for your attention!

The generalization properties of flow matching/diffusion are poorly understood

- Mainly two hypotheses:
  - Architecture regularization
  - Dynamical regularization through optimization
- Other source for the generalization
  - Stochasticity at inference/Errors in numerical ODE/SDE resolution[17,18]

[17] É. Pierret et al., **Diffusion Models for Gaussian Distributions: Exact Solutions and Wasserstein Errors,** In: ICML, 2025.
[18] S. Hurault et al., **From Score Matching to Diffusion: A Fine-Grained Error Analysis in the Gaussian Setting,** In: arXiv preprint arXiv:2503.11615, 2025.

ICML 2026 tutorial

Diffusion and Flow Matching

Part III: Beyond Generalization: Creativity

Quentin Bertrand & Mathurin Massias

https://memorization-generalization.github.io

Inria | ENS Lyon | Laboratoire Hubert Curien | Mila Affiliated Member | CIFAR Global Scholar

Why do Diffusion & Flow matching work so well?

Reminders:

Why do Diffusion & Flow matching work so well?

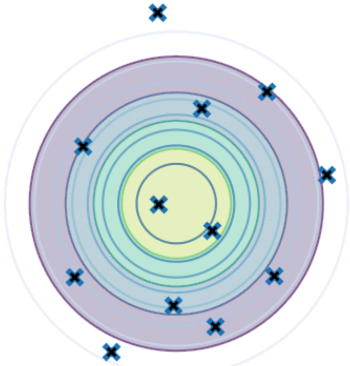
Reminders:

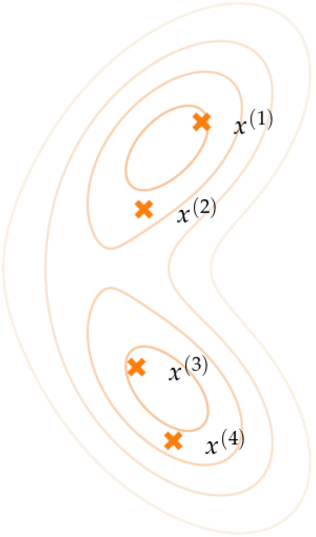
- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity

Why do Diffusion & Flow matching work so well?

Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$



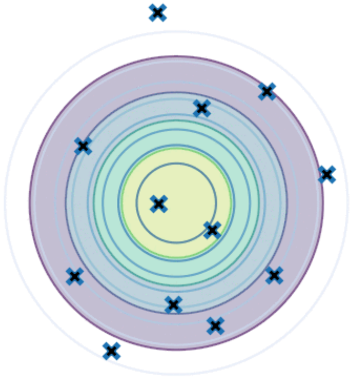


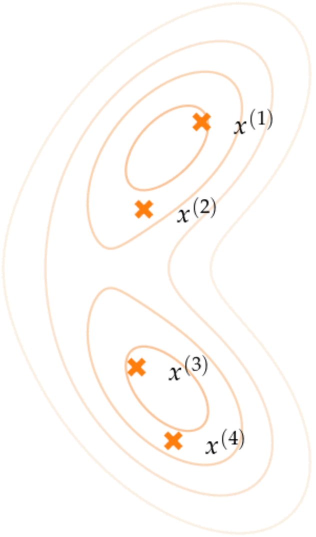
Why do Diffusion & Flow matching work so well?

Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models





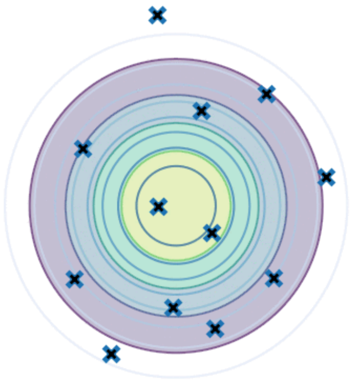
Why do Diffusion & Flow matching work so well?

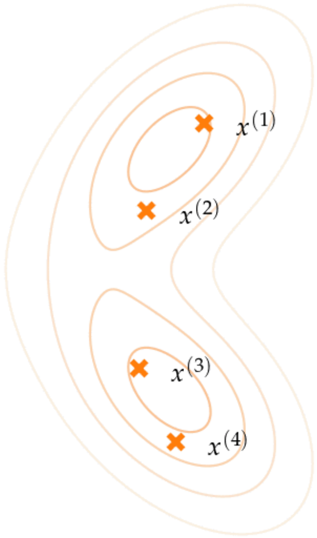
Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models

- "Should" only reproduce samples fom the training set





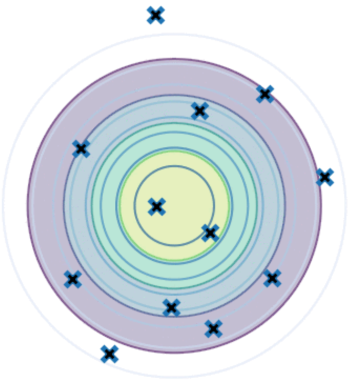
Why do Diffusion & Flow matching work so well?

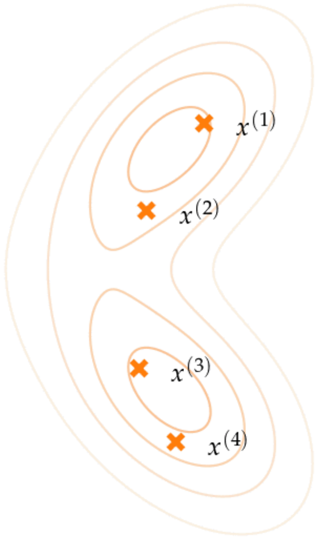
Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models

- "Should" only reproduce samples fom the training set
  - With the "optimal"/closed-form





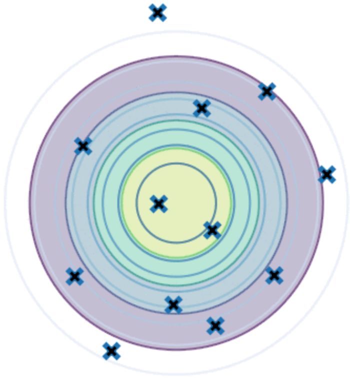
Why do Diffusion & Flow matching work so well?

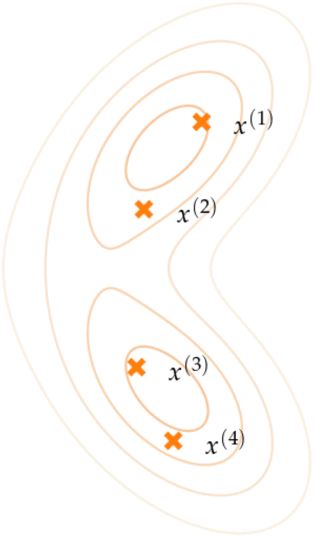
Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models

- "Should" only reproduce samples fom the training set
  - With the "optimal"/closed-form
- Yet they do generate new samples…





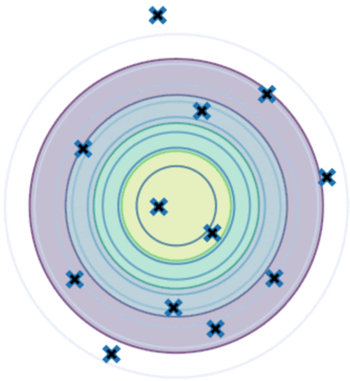
Why do Diffusion & Flow matching work so well?

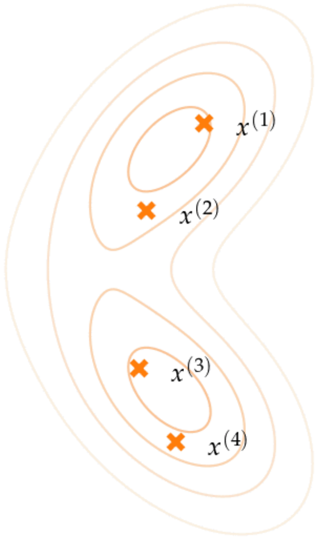
Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models

- "Should" only reproduce samples fom the training set
  - With the "optimal"/closed-form
- Yet they do generate new samples…
- And easily mix and match images/concepts





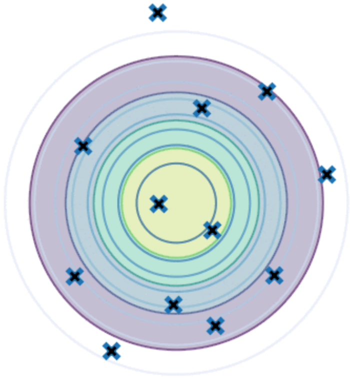
Why do Diffusion & Flow matching work so well?

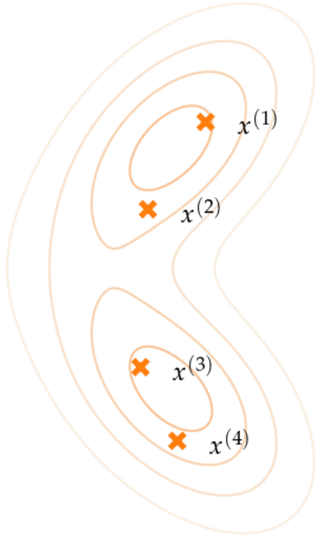
Reminders:

- There is a closed-form formula for the "ideal"/"perfect"/optimal score/velocity
  - That can only generate samples from the training set $\{x^{(1)}, \ldots, x^{(n)}\}$

Diffusion/flow matching models

- "Should" only reproduce samples fom the training set
  - With the "optimal"/closed-form
- Yet they do generate new samples…
- And easily mix and match images/concepts
- How can diffusion/flow matching combine concepts?





How to prevent memorization?

Proposed explanations:

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$
  - But optimization prevents memorization

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$
  - But optimization prevents memorization
- Architectural regularization[4,5,6,7,8]

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

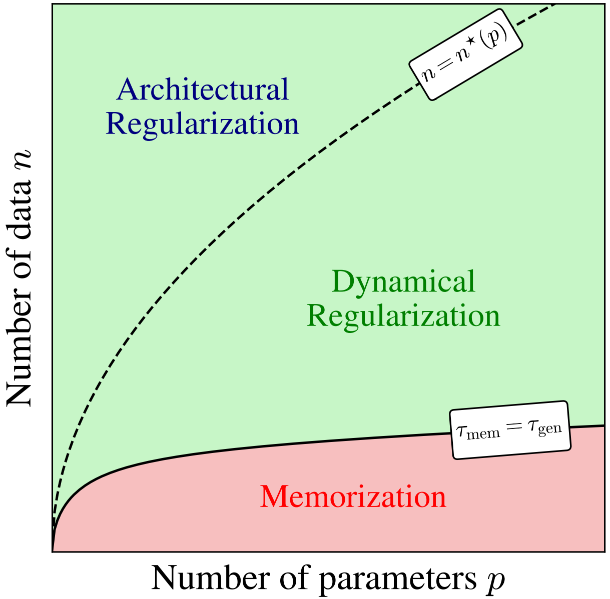
- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$
  - But optimization prevents memorization
- Architectural regularization[4,5,6,7,8]
  - Network capacity prevents memorization

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$
  - But optimization prevents memorization
- Architectural regularization[4,5,6,7,8]
  - Network capacity prevents memorization



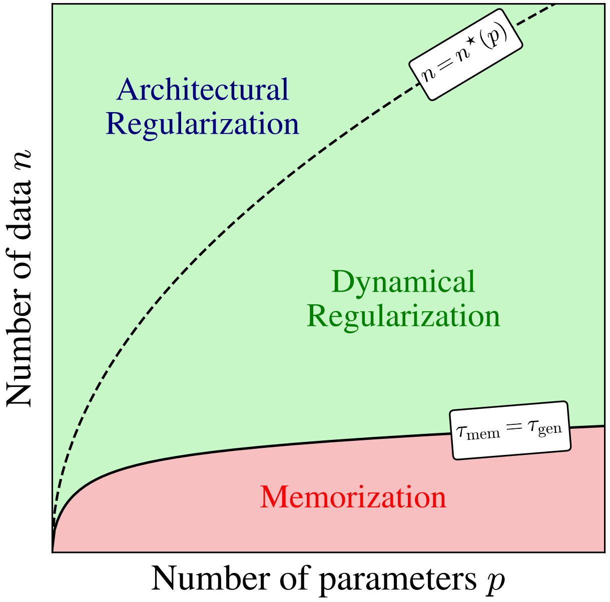
Bonnaire et al. (2025)

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

How to prevent memorization?

Proposed explanations:

- Dynamical regularization/Early stopping[1,2,3]
  - i.e. large enough network capacity
  - Able to memorize all the samples
  - $n_{\text{params}} \gg n_{\text{samples}}$
  - But optimization prevents memorization
- Architectural regularization[4,5,6,7,8]
  - Network capacity prevents memorization



Bonnaire et al. (2025)

Rest of this part: architectural regularization

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[1] T. Bonnaire et al., **Why Diffusion Models Don't Memorize: The Role of Implicit Dynamical Regularization in Training,** In: NeurIPS, 2025.
[2] T. Farghly et al., **Diffusion models and the manifold hypothesis: Log-domain smoothing is geometry adaptive,** In: NeurIPS, 2025.
[3] A. Favero et al., **Bigger Isn't Always Memorizing: Early Stopping Overparameterized Diffusion Models,** In: TMLR, 2026.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Beyond memorization: Mix and Match capabilities[4]

DALL·E, Ramesh et al.

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Beyond memorization: Mix and Match capabilities[4]

DALL·E, Ramesh et al.

How to understand diffusion mix and match capabilities in a minimalistic framework?

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Roadmap

Roadmap

- Goal: Understand diffusion/flow matching "generalization" capabilities

Roadmap

- Goal: Understand diffusion/flow matching "generalization" capabilities

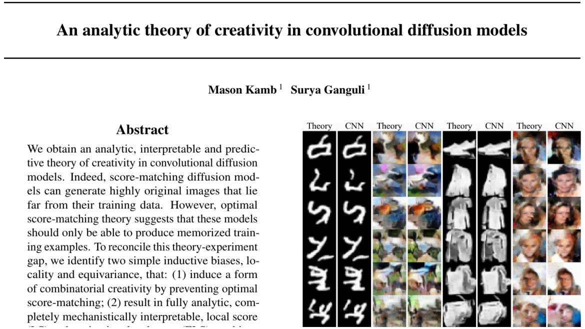
Today’s main references:

Roadmap

- Goal: Understand diffusion/flow matching "generalization" capabilities

Today's main references:

- An analytic theory of creativity in convolutional diffusion models[4]



An analytic theory of creativity in convolutional diffusion models

Mason Kamb[1] Surya Ganguli[1]

Abstract

We obtain an analytic, interpretable and predictive theory of creativity in convolutional diffusion models. Indeed, score-matching diffusion models can generate highly original images that lie far from their training data. However, optimal score-matching theory suggests that these models should only be able to produce memorized training examples. To reconcile this theory-experiment gap, we identify two simple inductive biases, locality and equivariance, that: (1) induce a form of combinatorial creativity by preventing optimal score-matching; (2) result in fully analytic, completely mechanistically interpretable, local score

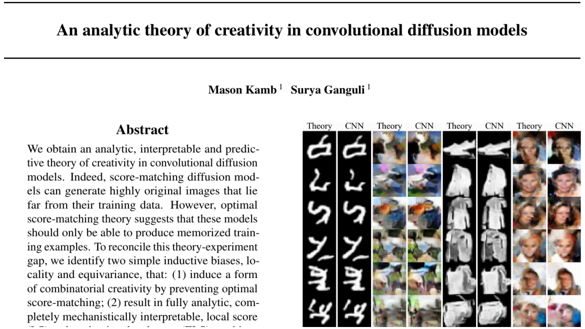
[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Roadmap

- Goal: Understand diffusion/flow matching "generalization" capabilities

Today's main references:

- An analytic theory of creativity in convolutional diffusion models[4]
- Towards a mechanistic explanation of diffusion model generalization[5]

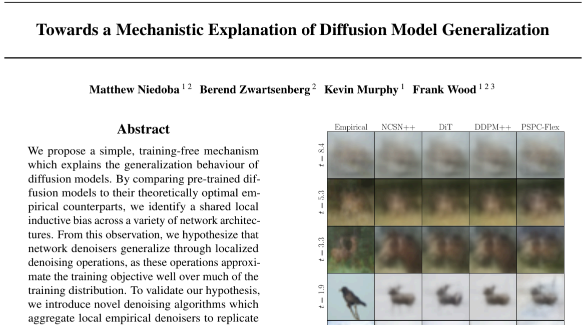


An analytic theory of creativity in convolutional diffusion models

Mason Kamb[1] Surya Ganguli[1]

Abstract

We obtain an analytic, interpretable and predictive theory of creativity in convolutional diffusion models. Indeed, score-matching diffusion models can generate highly original images that lie far from their training data. However, optimal score-matching theory suggests that these models should only be able to produce memorized training examples. To reconcile this theory-experiment gap, we identify two simple inductive biases, locality and equivariance, that: (1) induce a form of combinatorial creativity by preventing optimal score-matching; (2) result in fully analytic, completely mechanistically interpretable, local score



Towards a Mechanistic Explanation of Diffusion Model Generalization

Matthew Niedoba[1 2] Berend Zwartsenberg[2] Kevin Murphy[1] Frank Wood[1 2 3]

Abstract

We propose a simple, training-free mechanism which explains the generalization behaviour of diffusion models. By comparing pre-trained diffusion models to their theoretically optimal empirical counterparts, we identify a shared local inductive bias across a variety of network architectures. From this observation, we hypothesize that network denoisers generalize through localized denoising operations, as these operations approximate the training objective well over much of the training distribution. To validate our hypothesis, we introduce novel denoising algorithms which aggregate local empirical denoisers to replicate

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Roadmap

- Goal: Understand diffusion/flow matching "generalization" capabilities

Today’s main references:

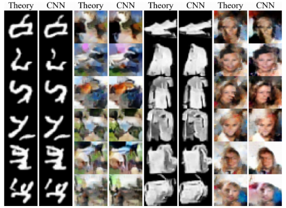
- An analytic theory of creativity in convolutional diffusion models[4]
- Towards a mechanistic explanation of diffusion model generalization[5]
- Locality in image diffusion models emerges from data statistics[6]

An analytic theory of creativity in convolutional diffusion models

Mason Kamb [1] Surya Ganguli [1]

Abstract

We obtain an analytic, interpretable and predictive theory of creativity in convolutional diffusion models. Indeed, score-matching diffusion models can generate highly original images that lie far from their training data. However, optimal score-matching theory suggests that these models should only be able to produce memorized training examples. To reconcile this theory-experiment gap, we identify two simple inductive biases, locality and equivariance, that: (1) induce a form of combinatorial creativity by preventing optimal score-matching; (2) result in fully analytic, completely mechanistically interpretable, local score

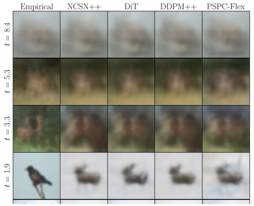


Towards a Mechanistic Explanation of Diffusion Model Generalization

Matthew Niedoba [1 2] Berend Zwartsenberg [2] Kevin Murphy [1] Frank Wood [1 2 3]

Abstract

We propose a simple, training-free mechanism which explains the generalization behaviour of diffusion models. By comparing pre-trained diffusion models to their theoretically optimal empirical counterparts, we identify a shared local inductive bias across a variety of network architectures. From this observation, we hypothesize that network denoisers generalize through localized denoising operations, as these operations approximate the training objective well over much of the training distribution. To validate our hypothesis, we introduce novel denoising algorithms which aggregate local empirical denoisers to replicate



Locality in Image Diffusion Models Emerges from Data Statistics

Artem Lukoianov
Massachusetts Institute of Technology
arteml@mit.edu

Chenyang Yuan
Toyota Research Institute
chenyang.yuan@tri.global

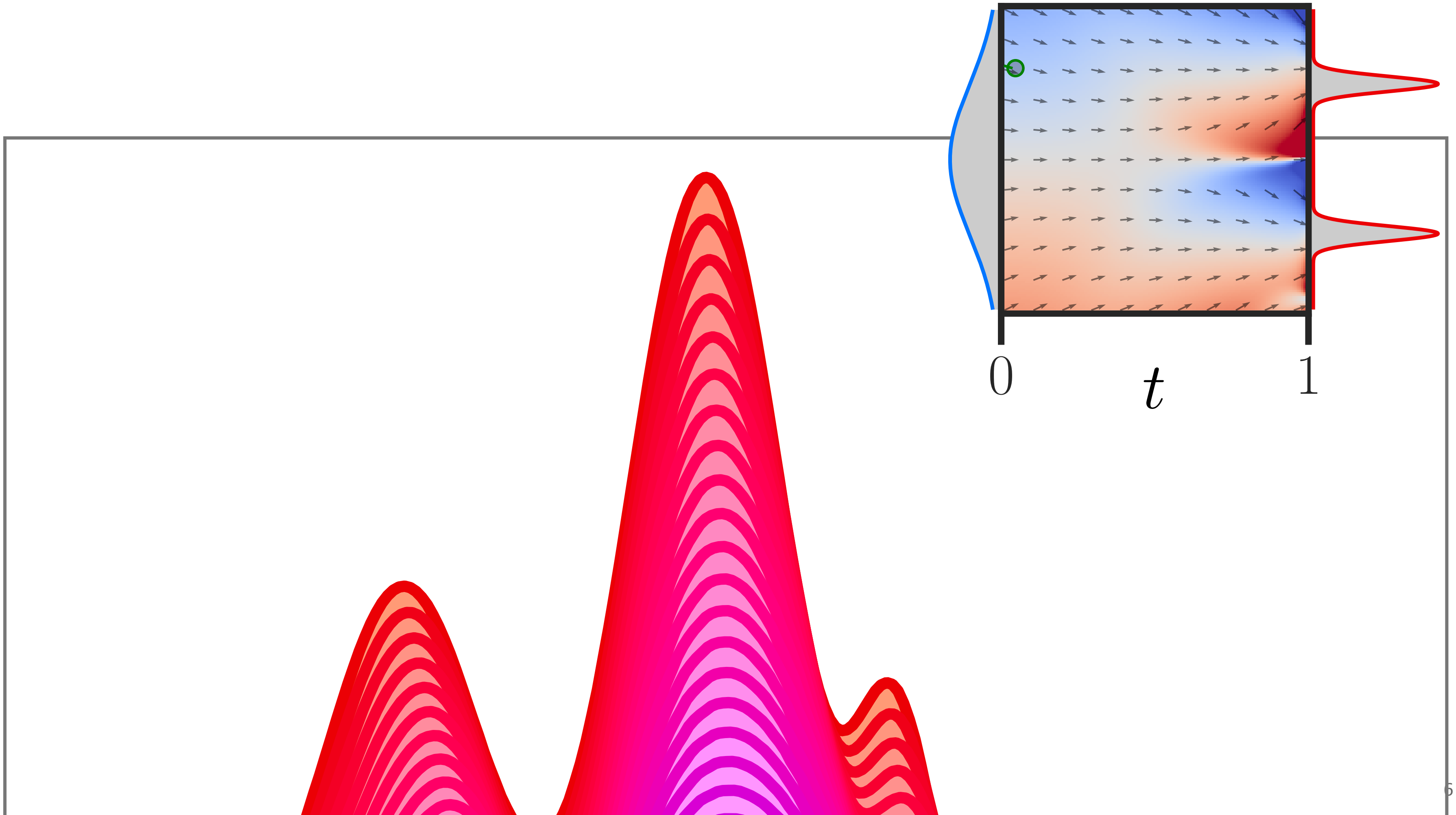
Justin Solomon
Massachusetts Institute of Technology
jsolomon@mit.edu

Vincent Sitzmann
Massachusetts Institute of Technology
sitzmann@mit.edu

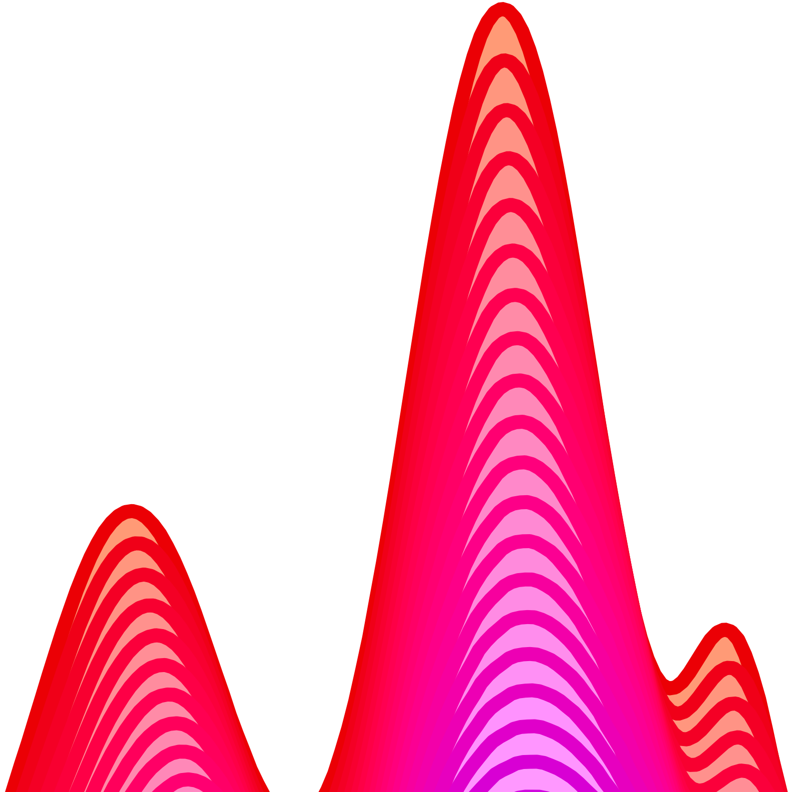
https://locality.lukoianov.com

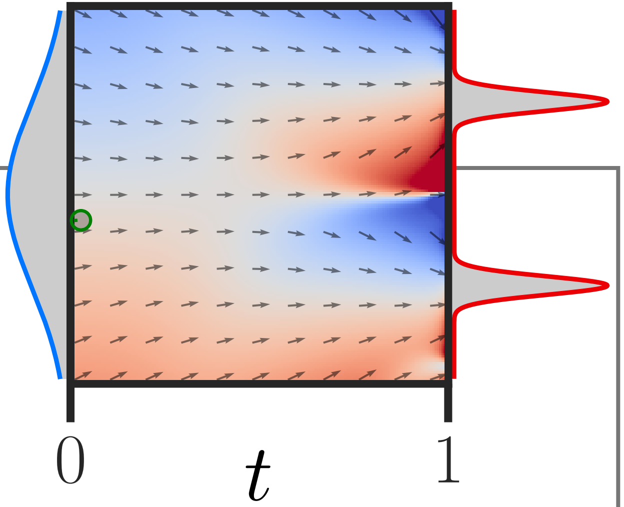
[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Reminder

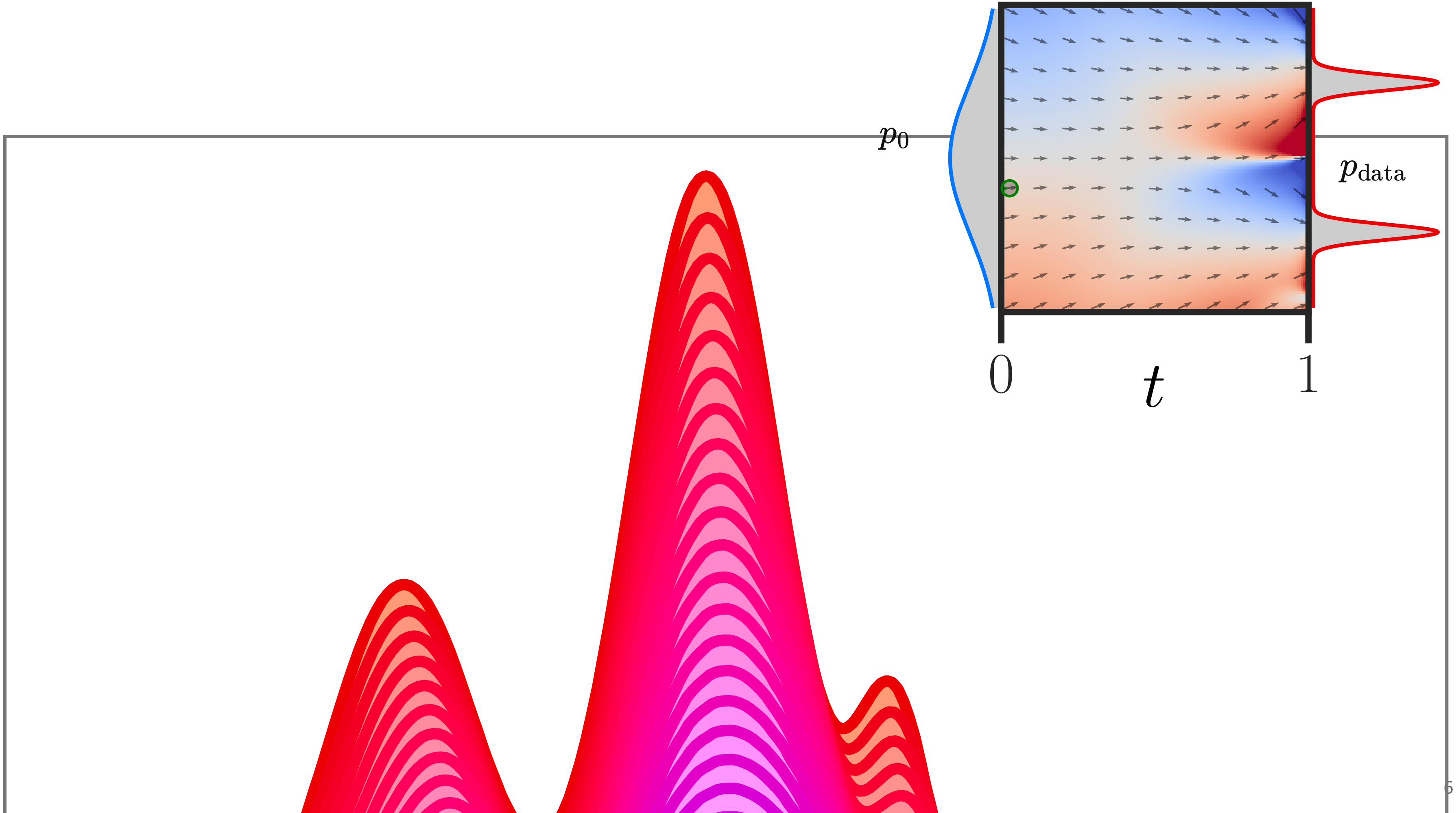


Reminder





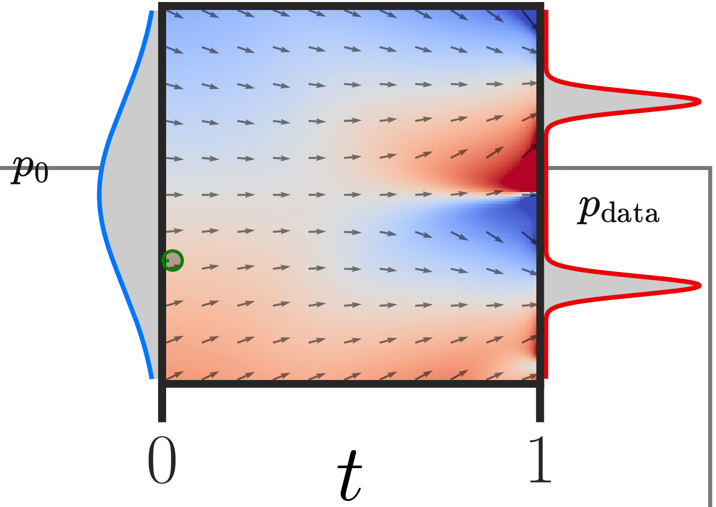
Reminder

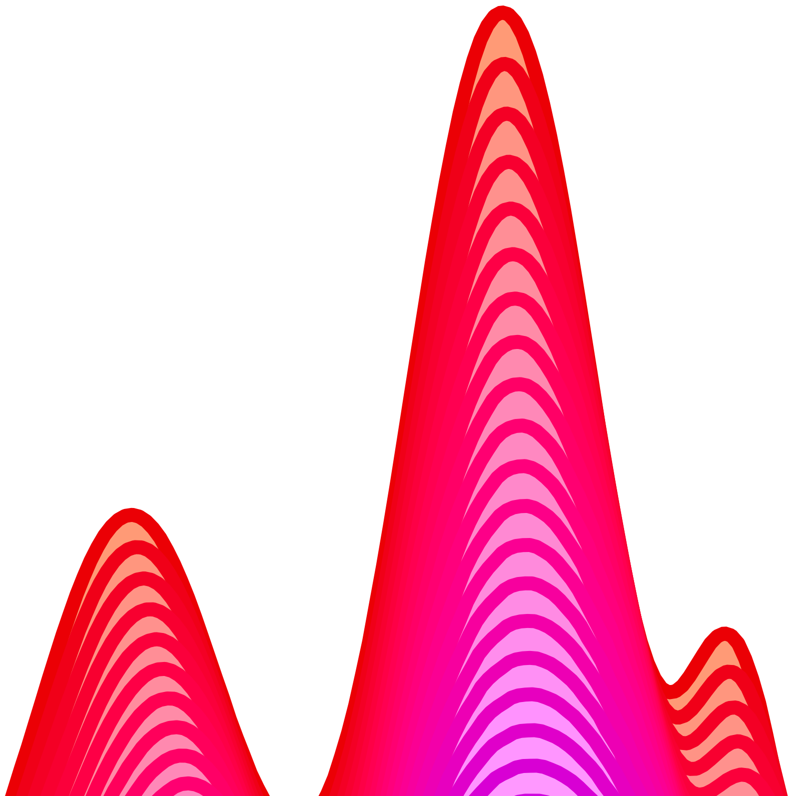


Problem Setting:

Reminder

- Access to n samples $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

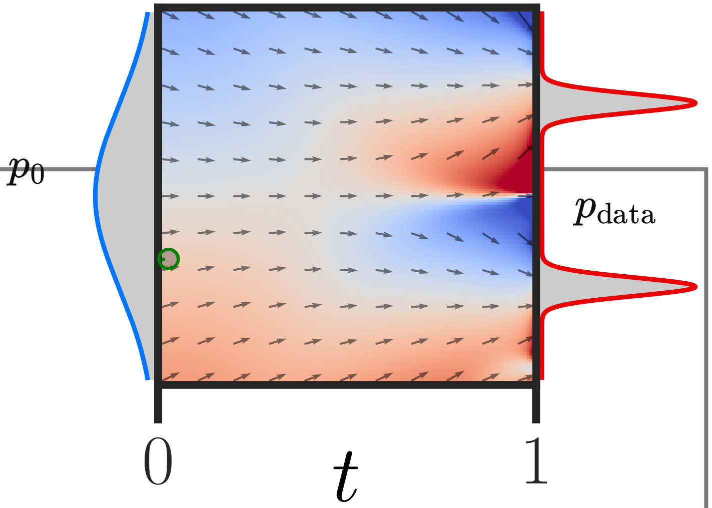


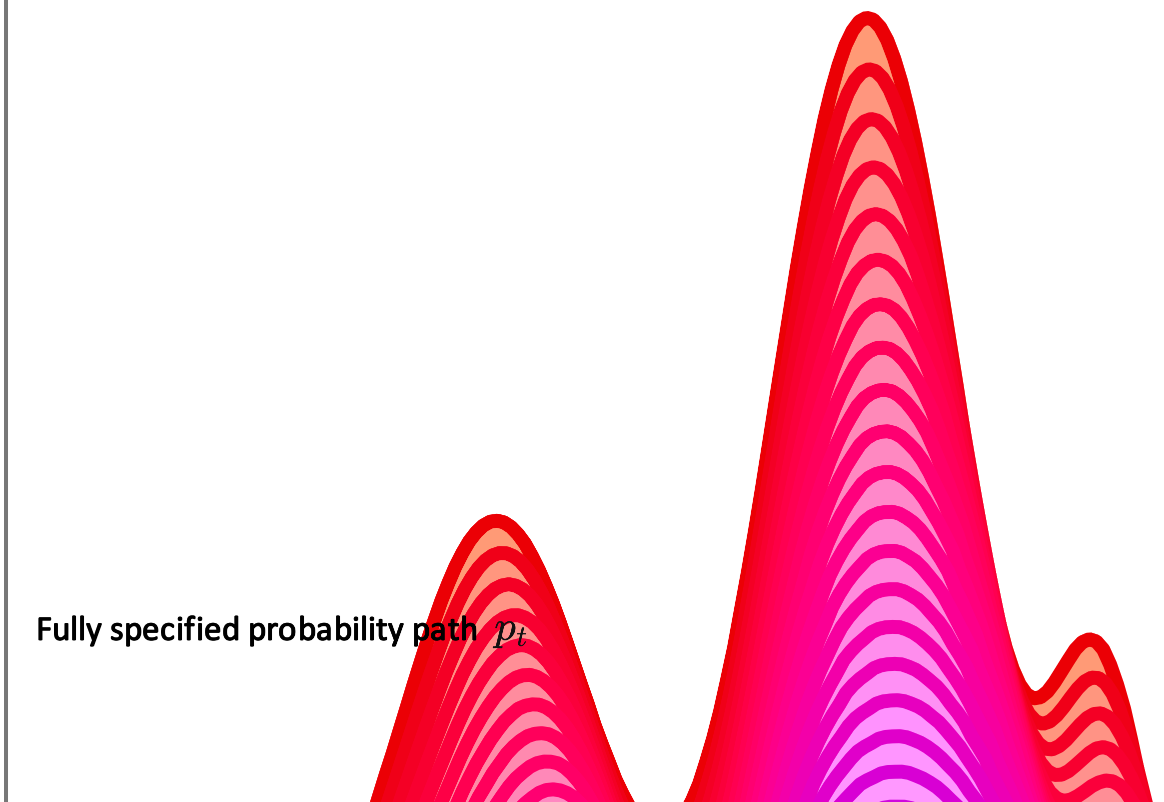


Problem Setting:

Reminder

- Access to n samples $x^{(1)}, \dots, x^{(n)} \in \mathbb{R}^d$
- $\underbrace{x^{(1)}, \dots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$

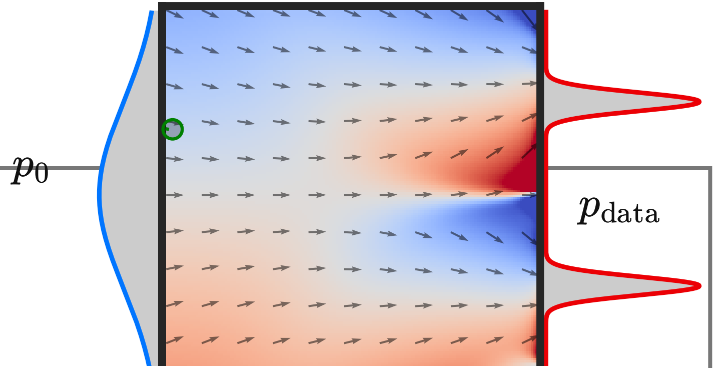


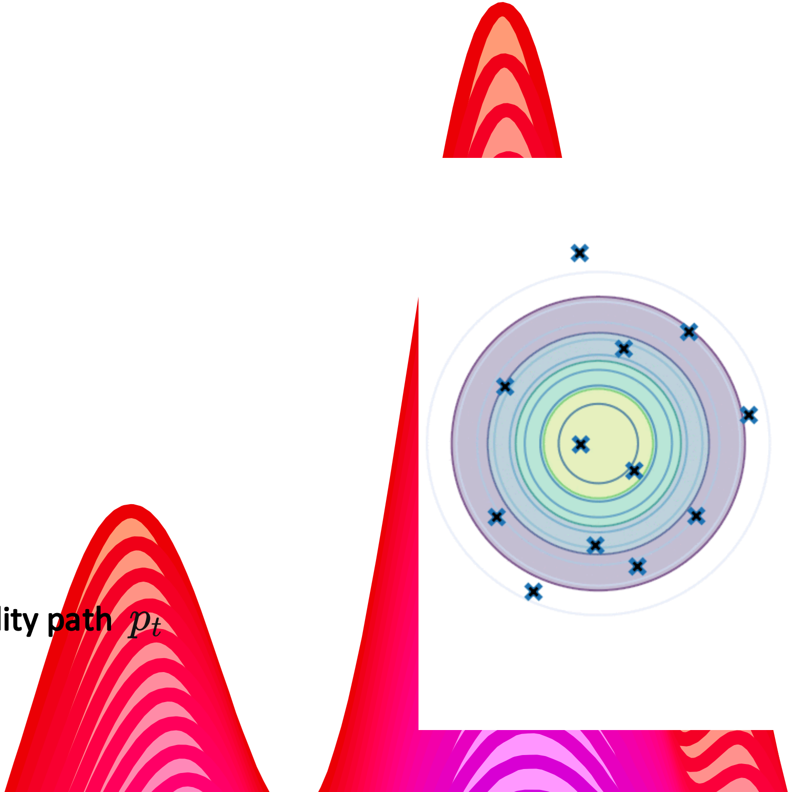


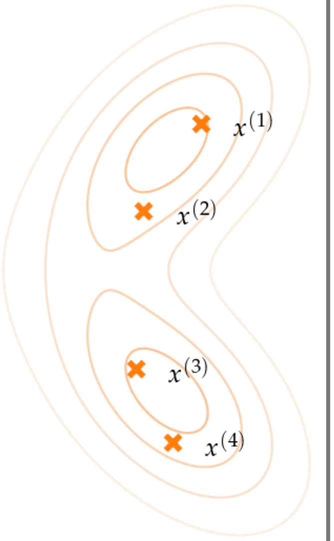
Problem Setting:

Reminder

- Access to n samples $x^{(1)}, \ldots, x^{(n)} \in \mathbb{R}^d$
- $\underbrace{x^{(1)}, \ldots, x^{(n)}}_{\text{known}} \sim \underbrace{p_{\text{data}}}_{\text{unknown}}$



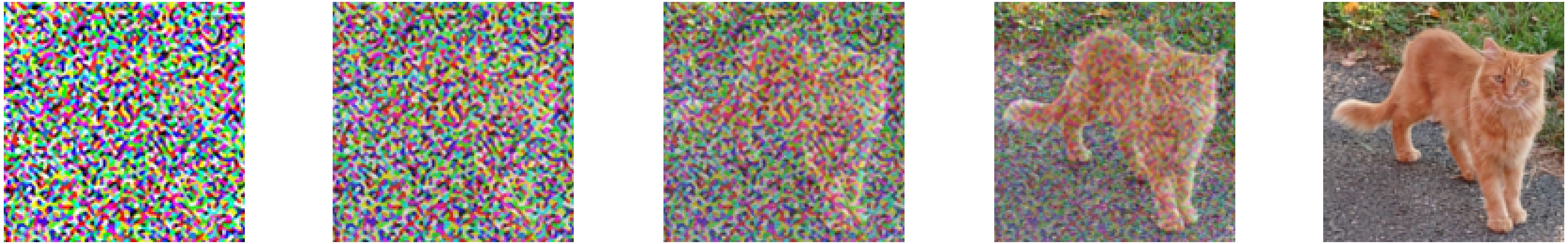




Fully specified probability path p_t

Flow matching reminder

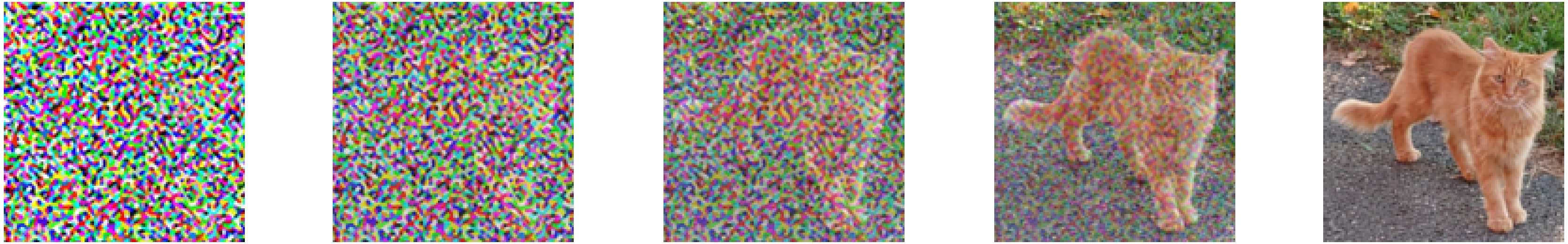
Goal: learn a velocity field $u : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$



$X_{t=0}$ $X_{t=0.25}$ $X_{t=0.5}$ $X_{t=0.75}$ $X_{t=1}$

Flow matching reminder

Goal: learn a velocity field $u : \mathbb{R}^d \times [0,1] \rightarrow \mathbb{R}^d$



$X_{t=0}$ $X_{t=0.25}$ $X_{t=0.5}$ $X_{t=0.75}$ $X_{t=1}$

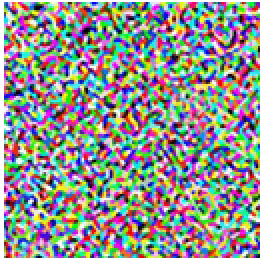
Practical loss[9,10,11]:

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

[9] Y. Lipman et al., **Flow Matching for Generative Modeling,** In: ICLR, 2023.
[10] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.
[11] X. Liu et al., **Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow,** In: ICLR, 2023.

Flow matching reminder

Goal: learn a velocity field $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

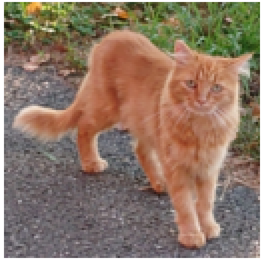


$X_{t=0}$

Closed-form solution for the minimizer $\hat{u}^\star$

$$\hat{u}^\star(x,t) = \sum_{i=1}^{n} \lambda_i(x,t) \frac{x^{(i)} - x}{1-t}$$

(if infinite capacity/no constraints on u_θ)



$X_{t=1}$

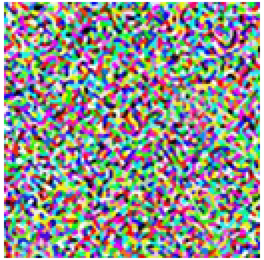
Practical loss[9,10,11]:

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

[9] Y. Lipman et al., **Flow Matching for Generative Modeling,** In: ICLR, 2023.
[10] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.
[11] X. Liu et al., **Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow,** In: ICLR, 2023.

Flow matching reminder

Goal: learn a velocity field $u : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$

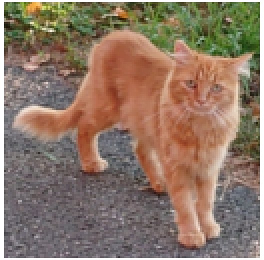


$X_{t=0}$

Closed-form solution for the minimizer $\hat{u}^\star$

$$\hat{u}^\star(x,t) = \sum_{i=1}^{n} \lambda_i(x,t) \frac{x^{(i)} - x}{1-t}$$

(if infinite capacity/no constraints on u_θ)



$X_{t=1}$

Can only generate samples $x^{(1)}, \ldots, x^{(n)}$ from the training set

[9] Y. Lipman et al., **Flow Matching for Generative Modeling,** In: ICLR, 2023.
[10] M. Albergo et al., **Building Normalizing Flows with Stochastic Interpolants,** In: ICLR, 2023.
[11] X. Liu et al., **Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow,** In: ICLR, 2023.

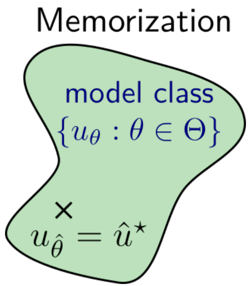
Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

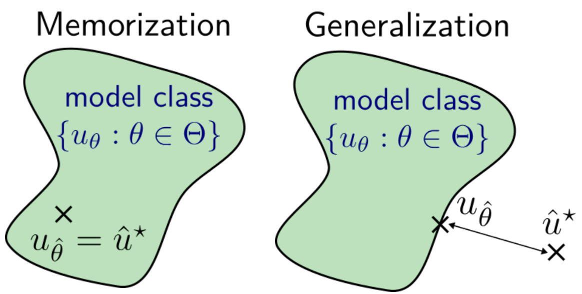
- If no constraints on u_θ /infinite capacity: **memorization**



Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

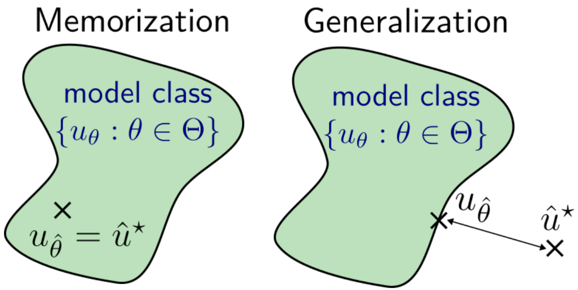
- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$



Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

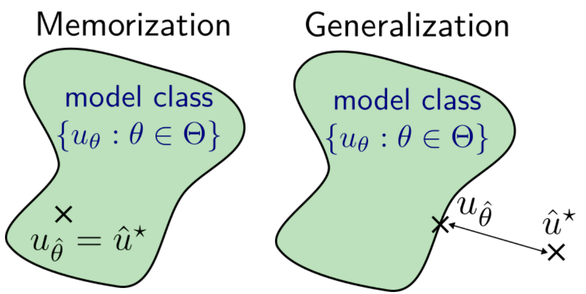
- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$
- **Minimalist model**: what about explicit constraints on the velocity field u ?[4,5]



[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$



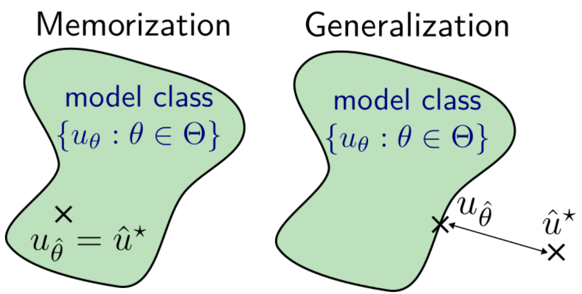
- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$
- **Minimalist model**: what about explicit constraints on the velocity field u ?[4,5]

Main idea: add explicit regularization on the score/velocity u

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim\hat{p}_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u_\theta((1-t)x_0+tx_1,t)-(x_1-x_0)||^2$$



- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$
- **Minimalist model**: what about explicit constraints on the velocity field u ?[4,5]

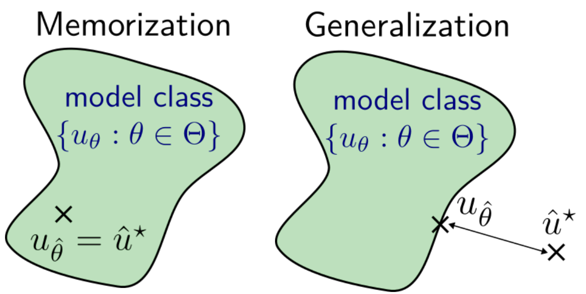
Main idea: add explicit regularization on the score/velocity u

$$\min_{u} \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim\hat{p}_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u((1-t)x_0+tx_1,t)-(x_1-x_0)||^2$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$



- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$
- **Minimalist model**: what about explicit constraints on the velocity field u ?[4,5]

Main idea: add explicit regularization on the score/velocity u

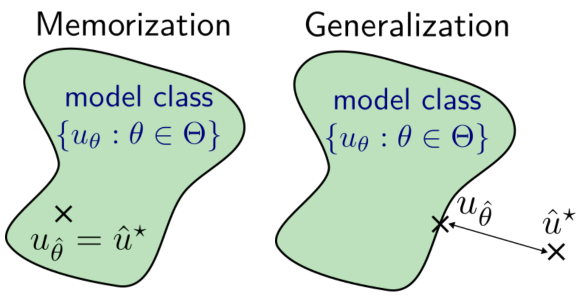
$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies equivariance constraints } (E)$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization: idea

$$\min_{u_\theta} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u_\theta((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$



- If no constraints on u_θ /infinite capacity: **memorization**
- **In practice**: architecture regularization $u_\theta, \theta \in \Theta$
- **Minimalist model**: what about explicit constraints on the velocity field u ?[4,5]

Main idea: add explicit regularization on the score/velocity u

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies equivariance constraints } (E)$$

$$u \text{ satisfies locality constraints } (L)$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization: idea

Main results:

- Derive a closed-form formula for this constrained optimization problem
- Show that it matches images generated by a U-Net in practice

Main idea: add explicit regularization on the score/velocity u

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies equivariance constraints } (E)$$

$$u \text{ satisfies locality constraints } (L)$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

[4] M. Kamb et al., An Analytic Theory of Creativity in Convolutional Diffusion Models, In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies equivariance constraints (Eq.)}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } T(u(x,t)) = u(T(x),t), \text{ for all 2D translations } T \text{ (Eq.)}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } T(u(x,t)) = u(T(x),t), \text{ for all 2D translations } T \text{ (Eq.)}$$

Closed-form solution for the minimizer $\hat{u}^\star_{\text{Eq.}}$ with equivariance constraints

$$\hat{u}^\star_{\text{Eq.}}(x,t) := \sum_{i=1}^{n} \sum_{T \in \mathcal{T}^{2D}} \lambda_{i,T}(x,t) \left(\frac{T(x^{(i)}) - x}{1-t} \right)$$

$$\lambda_{i,T}(x,t) := \text{softmax}\left(-\frac{||tT(x^{(j)}) - x||^2}{2(1-t)^2} \right)_{i,T}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

Reminder: closed-form formula without equivariance

$$\hat{u}^\star(x,t) = \sum_{i=1}^{n} \lambda_i(x,t)\left(\frac{x^{(i)} - x}{1-t}\right)$$

Closed-form solution for the minimizer $\hat{u}^\star_{\text{Eq.}}$ with equivariance constraints

$$\hat{u}^\star_{\text{Eq.}}(x,t) := \sum_{i=1}^{n} \sum_{T \in \mathcal{T}^{2D}} \lambda_{i,T}(x,t)\left(\frac{T(x^{(i)}) - x}{1-t}\right)$$

$$\lambda_{i,T}(x,t) := \text{softmax}\left(-\frac{\|tT(x^{(j)}) - x\|^2}{2(1-t)^2}\right)_{i,T}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #1: Equivariance[4]

Reminder: closed-form formula without equivariance

$$\hat{u}^\star(x,t) = \sum_{i=1}^{n} \lambda_i(x,t) \left(\frac{x^{(i)} - x}{1-t} \right)$$

Closed-form solution for the minimizer $\hat{u}^\star_{\mathrm{Eq.}}$ with equivariance constraints

$$\hat{u}^\star_{\mathrm{Eq.}}(x,t) := \sum_{i=1}^{n} \sum_{T \in \mathcal{T}^{2D}} \lambda_{i,T}(x,t) \left(\frac{T(x^{(i)}) - x}{1-t} \right)$$

Equivalent to data augmentation

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

[4] M. Kamb et al., An Analytic Theory of Creativity in Convolutional Diffusion Models, In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies locality constraints (Loc.)}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t) \;\; \text{(Loc.)}$$

$$M_t^q : \text{binary mask centered on pixel } q$$

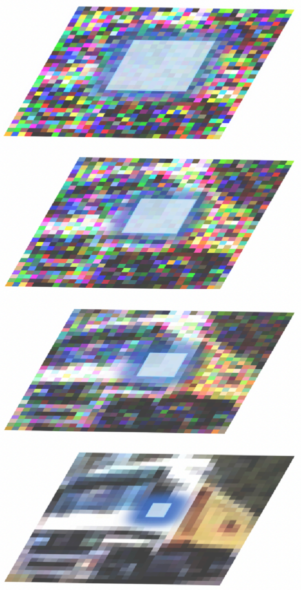
[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t) \;\; \text{(Loc.)}$$

$$M_t^q : \text{binary mask centered on pixel } q$$



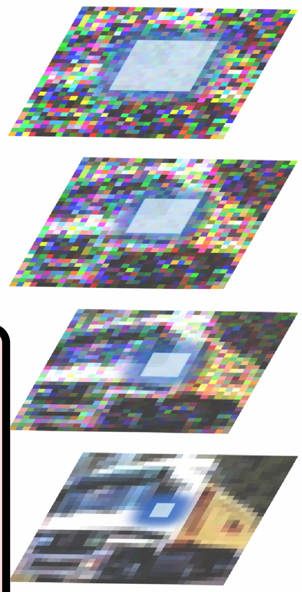
[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Explicit architectural regularization #2: Locality[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t) \quad \text{(Loc.)}$$

M_t^q : binary mask centered on pixel q



Closed-form solution for the minimizer $\hat{u}^\star_{\text{Loc.}}$ with locality constraints

$$\hat{u}^\star_{\text{Loc.}}(x,t) = \sum_{i=1}^{n} \lambda_i(x,t) \odot \left(\frac{x^{(i)} - x}{1-t} \right)$$

$$\lambda_i(x,t)^q = \text{softmax}\left(-\frac{\|M_t^q \odot (tx^{(j)} - x)\|^2}{2(1-t)^2} \right)_i$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

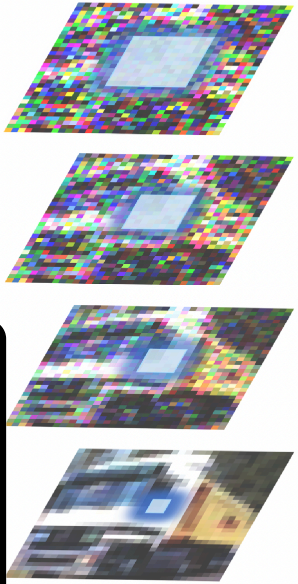
Explicit architectural regularization #2: Locality[4]

Constraint is on the denoiser. Not on the velocity field.

$$\min_u \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim\hat{p}_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u((1-t)x_0+tx_1,t)-(x_1-x_0)||^2$$

$$\text{s.t. } u^q(x,t)=u^q(M_t^q\odot x,t)\ \ \text{(Loc.)}$$

$$M_t^q : \text{binary mask centered on pixel } q$$



Closed-form solution for the minimizer $\hat{u}^\star_{\text{Loc.}}$ with locality constraints

$$\hat{u}^\star_{\text{Loc.}}(x,t)=\sum_{i=1}^{n}\lambda_i(x,t)\odot\left(\frac{x^{(i)}-x}{1-t}\right)$$

$$\lambda_i(x,t)^q=\text{softmax}\left(-\frac{\|M_t^q\odot(tx^{(j)}-x)\|^2}{2(1-t)^2}\right)_i$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Equivalence AND Locality[4]

[4] M. Kamb et al., An Analytic Theory of Creativity in Convolutional Diffusion Models, In: ICML, 2025.

Equivalance AND Locality[4]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

[4] M. Kamb et al., An Analytic Theory of Creativity in Convolutional Diffusion Models, In: ICML, 2025.

Equivariance AND Locality[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u \text{ satisfies equivariance constraints (Eq.)}$$

$$\text{s.t. } u \text{ satisfies locality constraints (Loc.)}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

Equivariance AND Locality[4]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } T(u(x,t)) = u(T(x),t), \text{ for all 2D translations } T \text{ (Eq.)}$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t) \text{ (Loc.)}$$

M_t^q : Binary mask centered on pixel q

[4] M. Kamb et al., An Analytic Theory of Creativity in Convolutional Diffusion Models, In: ICML, 2025.

Equivariance AND Locality[4]

$$\min_u \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim\hat{p}_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u((1-t)x_0+tx_1,t)-(x_1-x_0)||^2$$

$$\text{s.t. } T(u(x,t)) = u(T(x),t),\ \text{for all 2D translations } T \text{ (Eq.)}$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t)\ \ \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

Closed-form solution for the minimizer $\hat{u}^\star_{\text{E+L}}$ with locality constraints

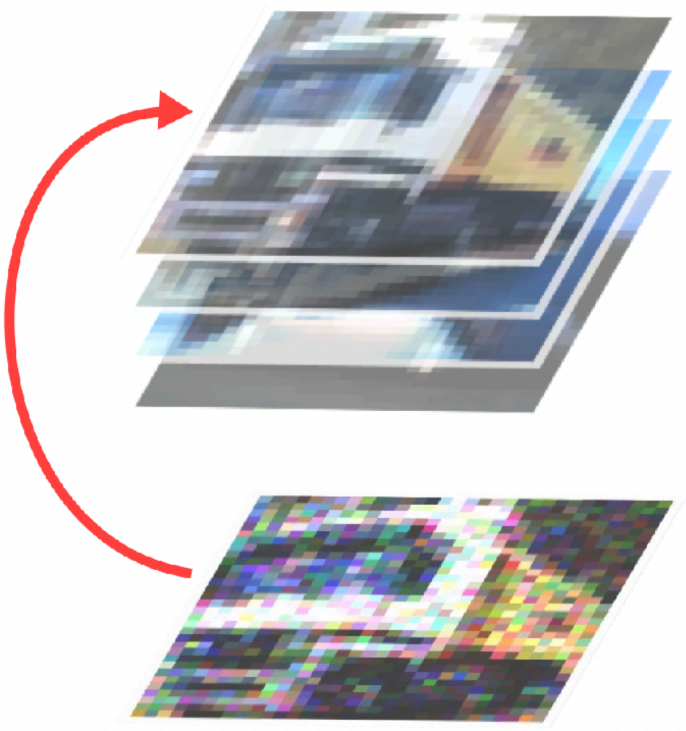
$$\hat{u}^\star_{\text{Eq.+Loc.}}(x,t) = \sum_{i=1}^{n}\sum_{T\in\mathcal{T}^{2D}} \lambda_{i,T}(x,t)\odot\left(\frac{T(x^{(i)})-x}{1-t}\right)$$

$$\lambda_{i,T}(x,t)^q = \text{softmax}\left(-\frac{\|M_t^q\odot(tT(x^{(j)})-x)\|^2}{2(1-t)^2}\right)_{i,T}$$

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.

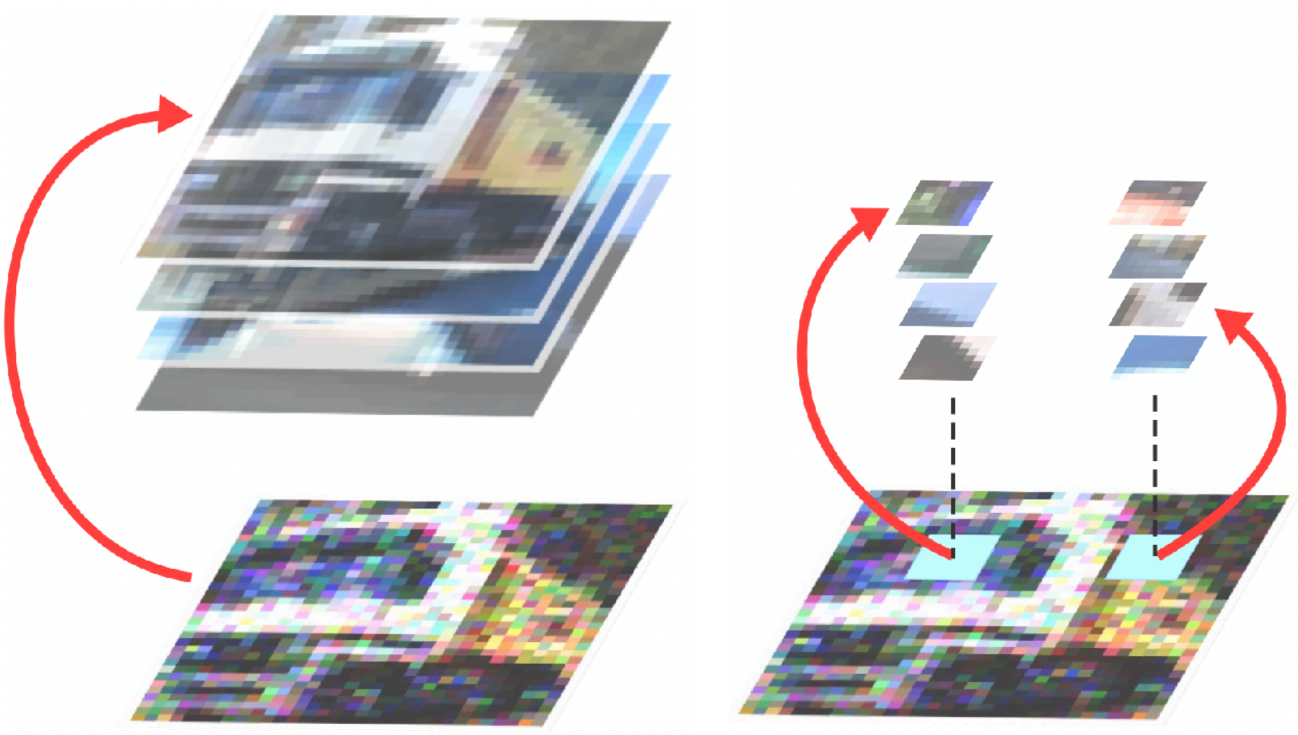
Illustration of the impact of locality & equiv. constraints

Illustration of the impact of locality & equiv. constraints



Vanilla closed-form

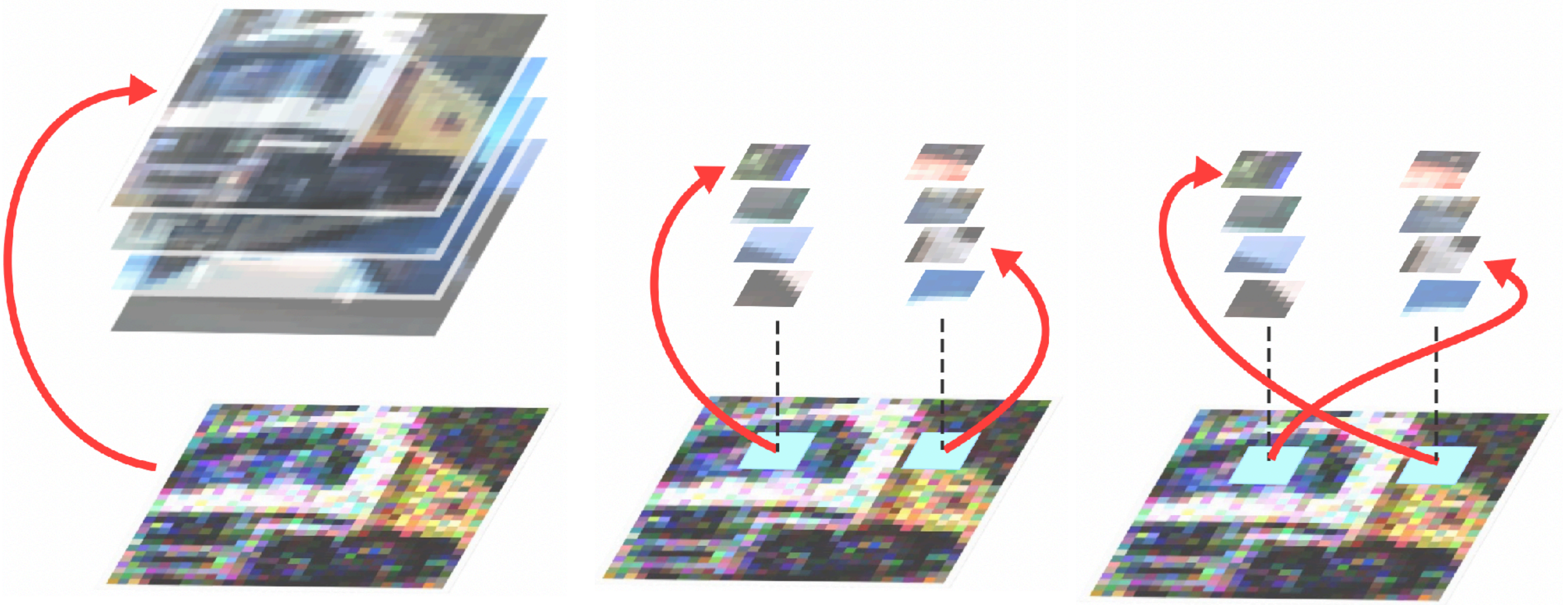
Illustration of the impact of locality & equiv. constraints



Vanilla closed-form

Closed-form for locality constraints

Illustration of the impact of locality & equiv. constraints



Vanilla closed-form

Closed-form for locality constraints

Closed-form for locality + equivariance constraints

Empirical validation: closed-form vs U-Net

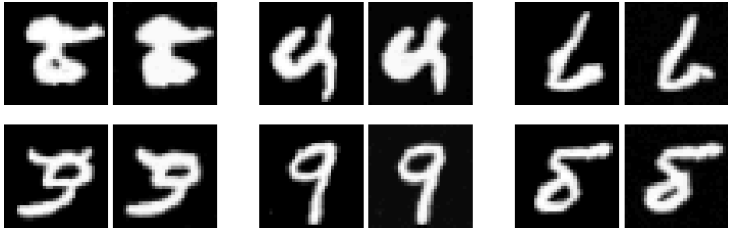
Empirical validation: closed-form vs U-Net

Left: Closed-Form with Locality and Equivariance Constraints
Right: Learned (small) U-Net

Empirical validation: closed-form vs U-Net

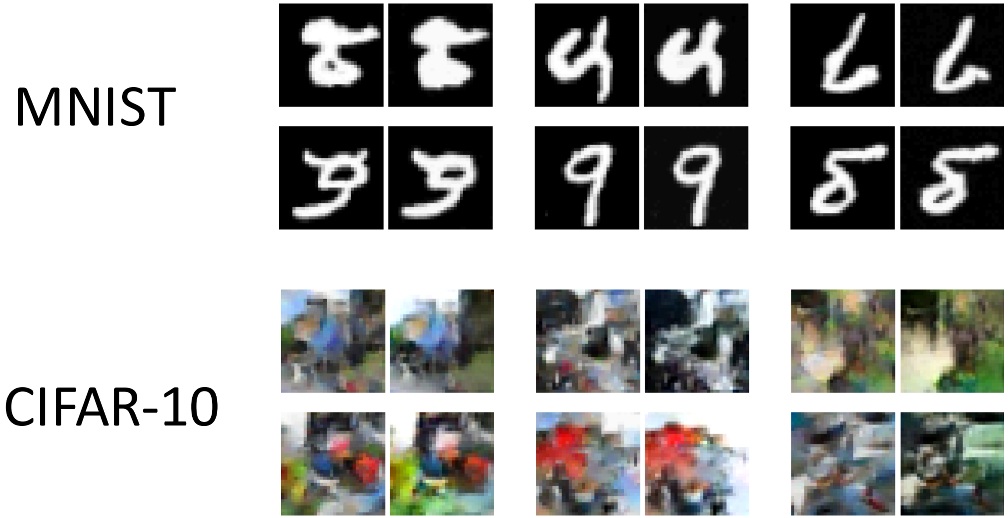
Left: Closed-Form with Locality and Equivariance Constraints
Right: Learned (small) U-Net

MNIST



Empirical validation: closed-form vs U-Net

Left: Closed-Form with Locality and Equivariance Constraints
Right: Learned (small) U-Net



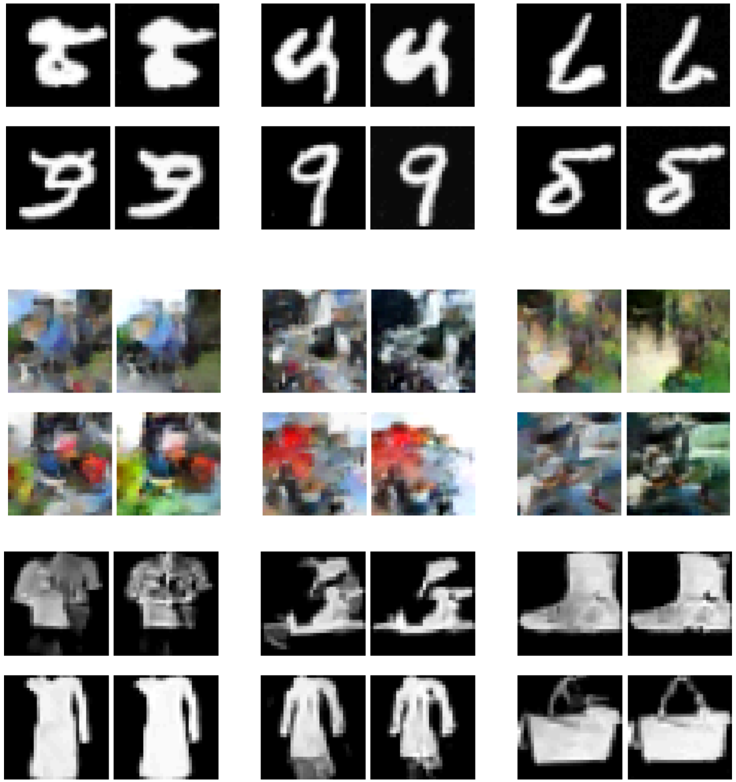
Empirical validation: closed-form vs U-Net

Left: Closed-Form with Locality and Equivariance Constraints
Right: Learned (small) U-Net

MNIST

CIFAR-10

Fashion MNIST



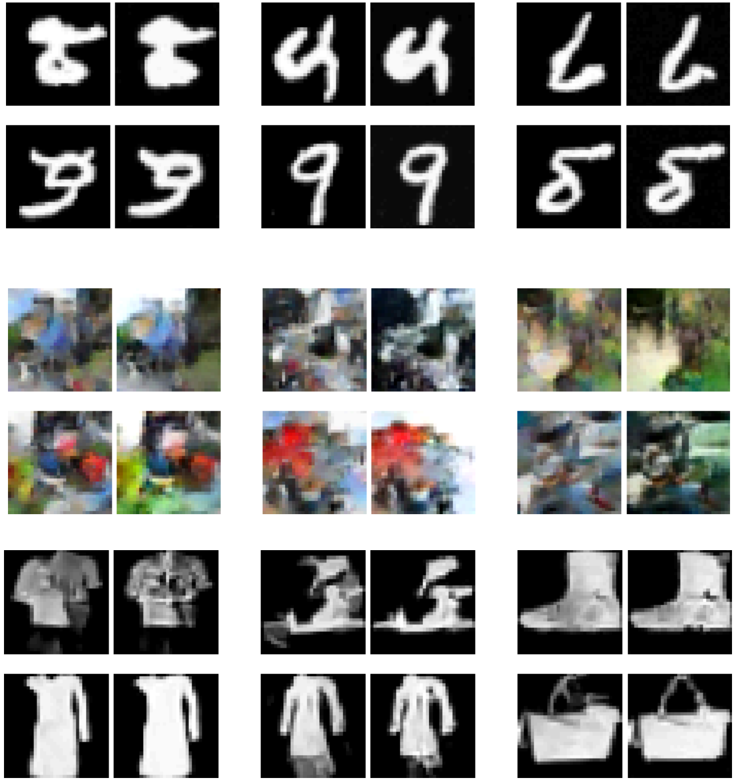
Empirical validation: closed-form vs U-Net

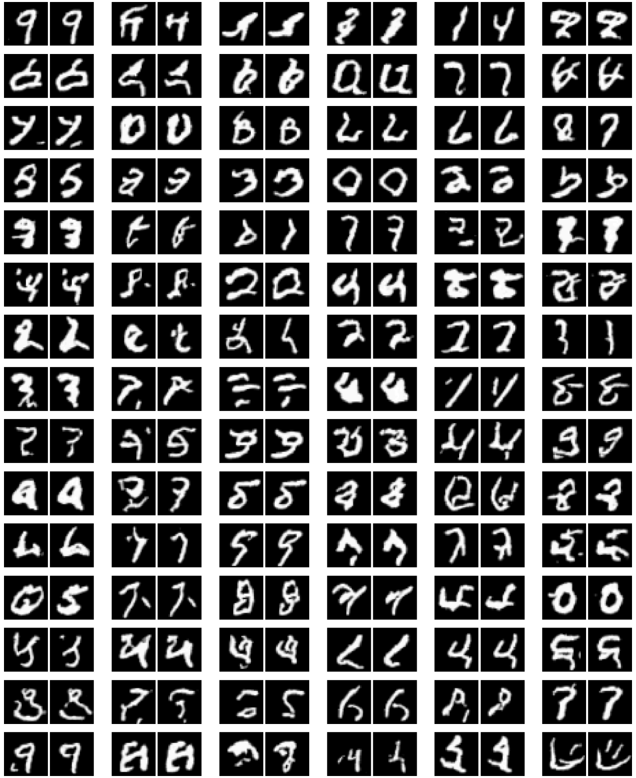
Left: Closed-Form with Locality and Equivariance Constraints
Right: Learned (small) U-Net

MNIST

CIFAR-10

Fashion MNIST





Empirical validation: receptive field comparison 1/2

Empirical validation: receptive field comparison 1/2

What is a receptive field?

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$

What is a receptive field?

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$

What is a receptive field?

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$

What is a receptive field?

- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\in\mathbb{R},\ \text{real valued}}$

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$

What is a receptive field?

- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\in\mathbb{R},\ \text{real valued}}$

Question

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$

What is a receptive field?

- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\in\mathbb{R}, \text{ real valued}}$

Question

What is the size of $\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)$?

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x,t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}},t)}_{\text{image}}$
- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}},t)}_{\in\mathbb{R},\text{ real valued}}$
- $\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}},t)}_{\text{image}}$?

What is a receptive field?

Question

What is the size of $\nabla_x u_\theta^q(\overbrace{x}^{\text{image}},t)$?

Empirical validation: receptive field comparison 1/2

- Learned velocity field $u_\theta(x, t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$
- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\in\mathbb{R},\ \text{real valued}}$
- $\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{image}}$?

What is a receptive field?

Question

What is the size of $\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)$?

$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{image; receptive field at pixel } q}$$

Empirical validation: receptive field comparison 1/2

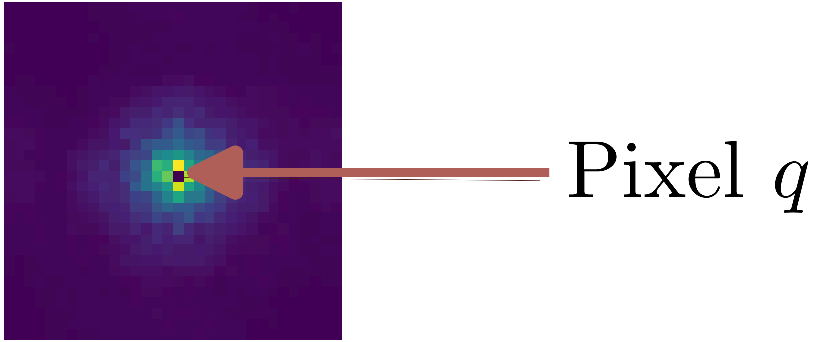
- Learned velocity field $u_\theta(x,t)$
- Learned velocity field $\underbrace{u_\theta(\overbrace{x}^{\text{image}},t)}_{\text{image}}$
- Value of the velocity field at pixel q : $\underbrace{u_\theta^q(\overbrace{x}^{\text{image}},t)}_{\in\mathbb{R},\text{ real valued}}$
- $\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}},t)}_{\text{image}}$?

What is a receptive field?

Question

What is the size of $\nabla_x u_\theta^q(\overbrace{x}^{\text{image}},t)$?

$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}},t)}_{\text{image; receptive field at pixel } q}$$

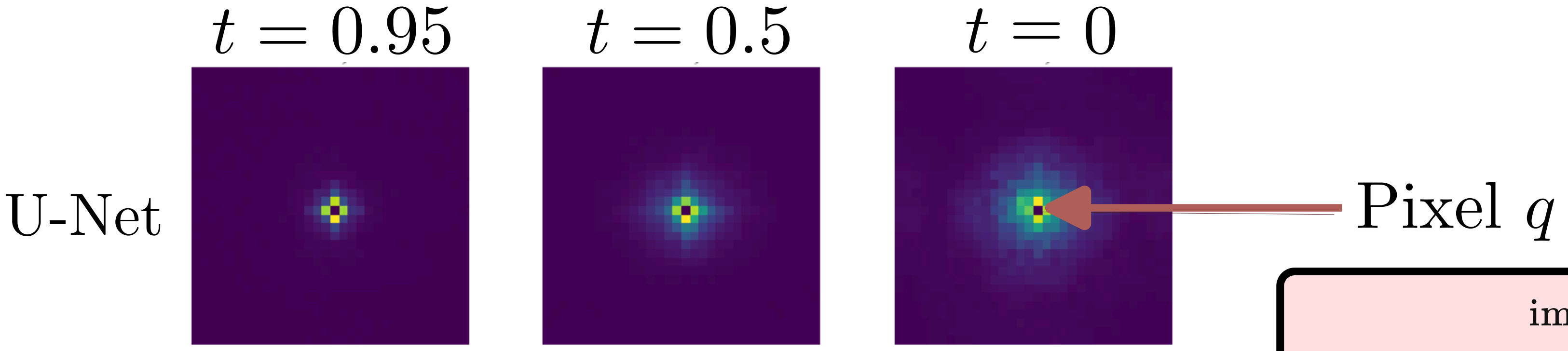


Empirical validation: receptive field comparison 2/2

Empirical validation: receptive field comparison 2/2

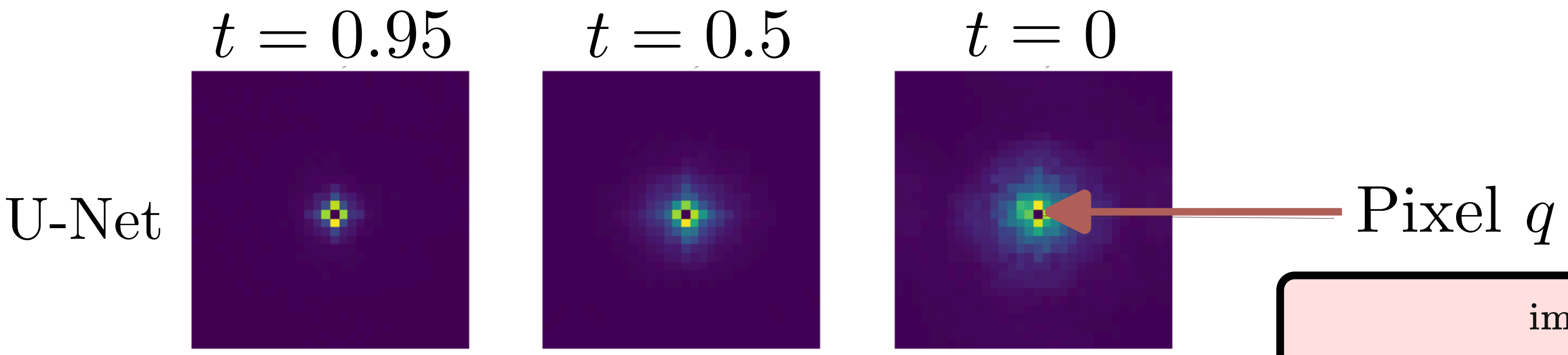
$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{receptive field at pixel } q}$$

Empirical validation: receptive field comparison 2/2



$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{receptive field at pixel } q}$$

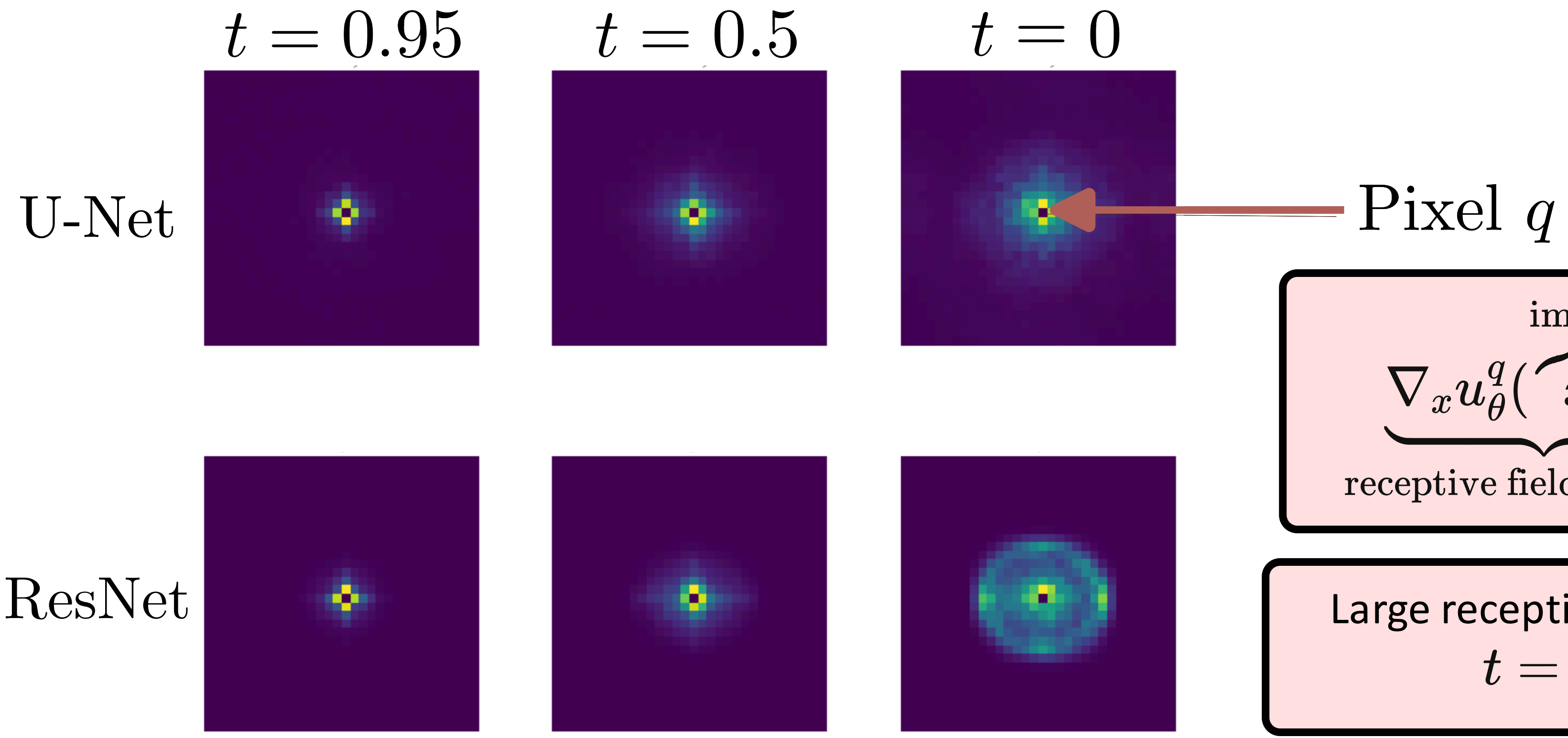
Empirical validation: receptive field comparison 2/2



$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{receptive field at pixel } q}$$

Large receptive field at $t = 0$

Empirical validation: receptive field comparison 2/2



$$\underbrace{\nabla_x u_\theta^q(\overbrace{x}^{\text{image}}, t)}_{\text{receptive field at pixel } q}$$

Large receptive field at $t = 0$

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_u \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim p_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u(tx_1+(1-t)x_0,t)-(x_1-x_0)||^2$$

$$\text{s.t. } u^q(x,t)=u^q(M_t^q\odot x,t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

Limitations:

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$
$$M_t^q : \text{Binary mask centered on pixel } q$$

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

Limitations:

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

How to choose the binary mask constraints M_t^q ?

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

Limitations:

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

Limitations:

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q
- And for each time t

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_u \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim p_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

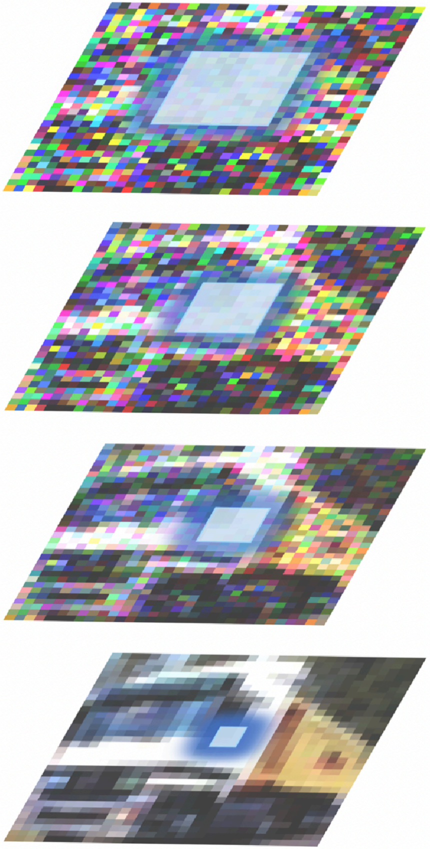
$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

Limitations:

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q
- And for each time t



[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

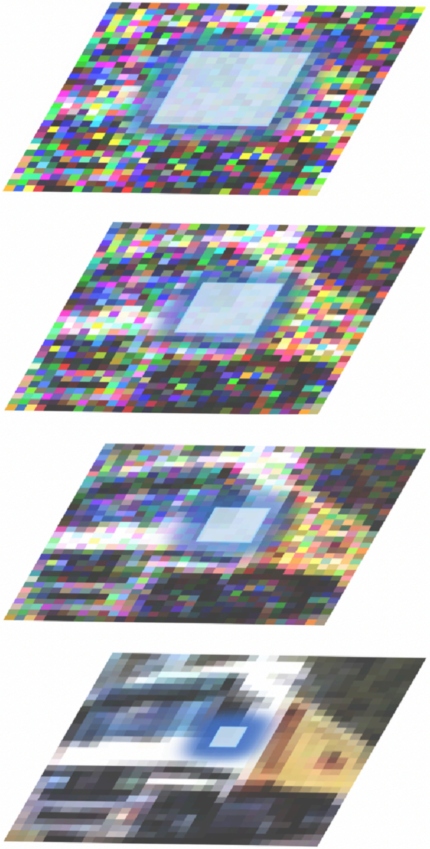
Local mask constraint limitations[6]

Locality constraint reminder

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

M_t^q : Binary mask centered on pixel q



Limitations:

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q
- And for each time t
- Not trivial

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim p_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u(tx_1 + (1-t)x_0, t) - (x_1 - x_0)||^2$$

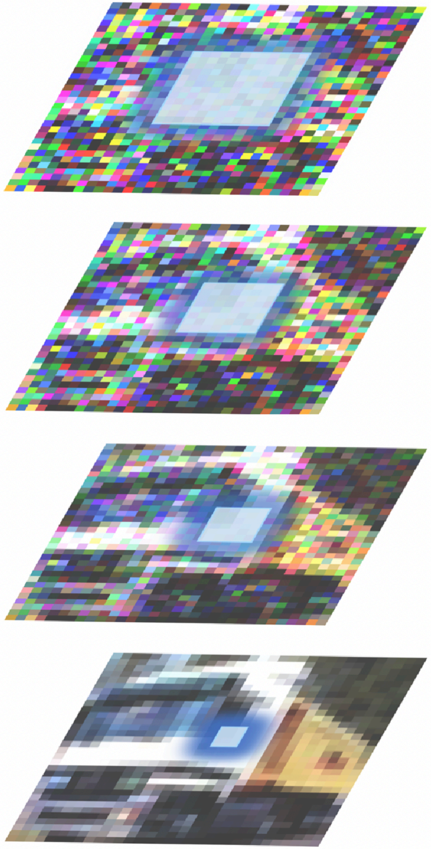
Limitations:

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q
- And for each time t
- Not trivial
- Cross-validated using a pre-trained U-Net u_θ



[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Local mask constraint limitations[6]

Locality constraint reminder

$$\min_u \mathbb{E}_{\substack{x_0\sim p_0\\ x_1\sim p_{\text{data}}\\ t\sim\mathcal{U}([0,1])}} ||u(tx_1+(1-t)x_0,t)-(x_1-x_0)||^2$$

Limitations:

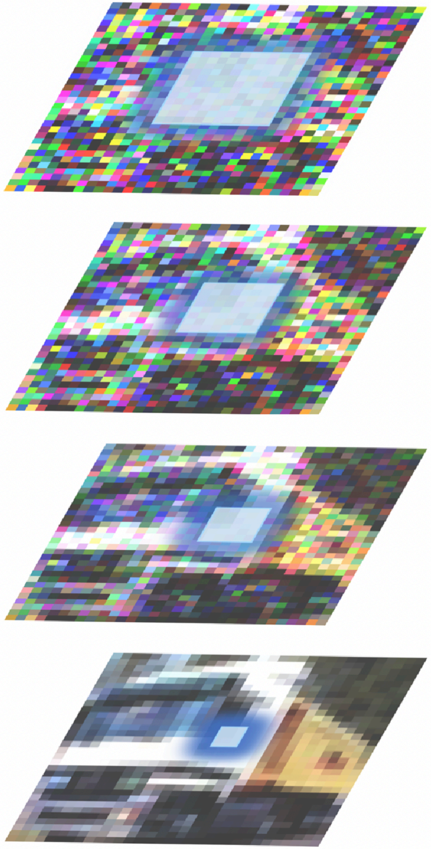
$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Loc.)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

How to choose the binary mask constraints M_t^q ?

- One mask for each pixel q
- And for each time t
- Not trivial
- Cross-validated using a pre-trained U-Net u_θ

Local receptive field: too restrictive

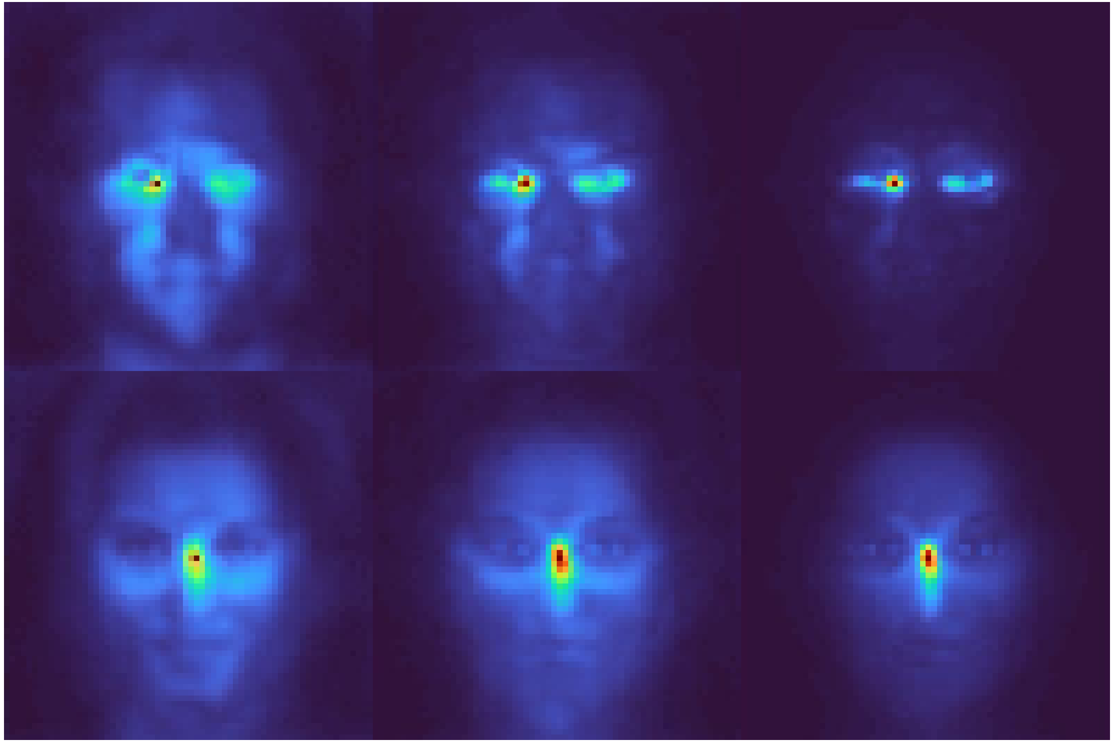


[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Example of non-local receptive field[6]

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

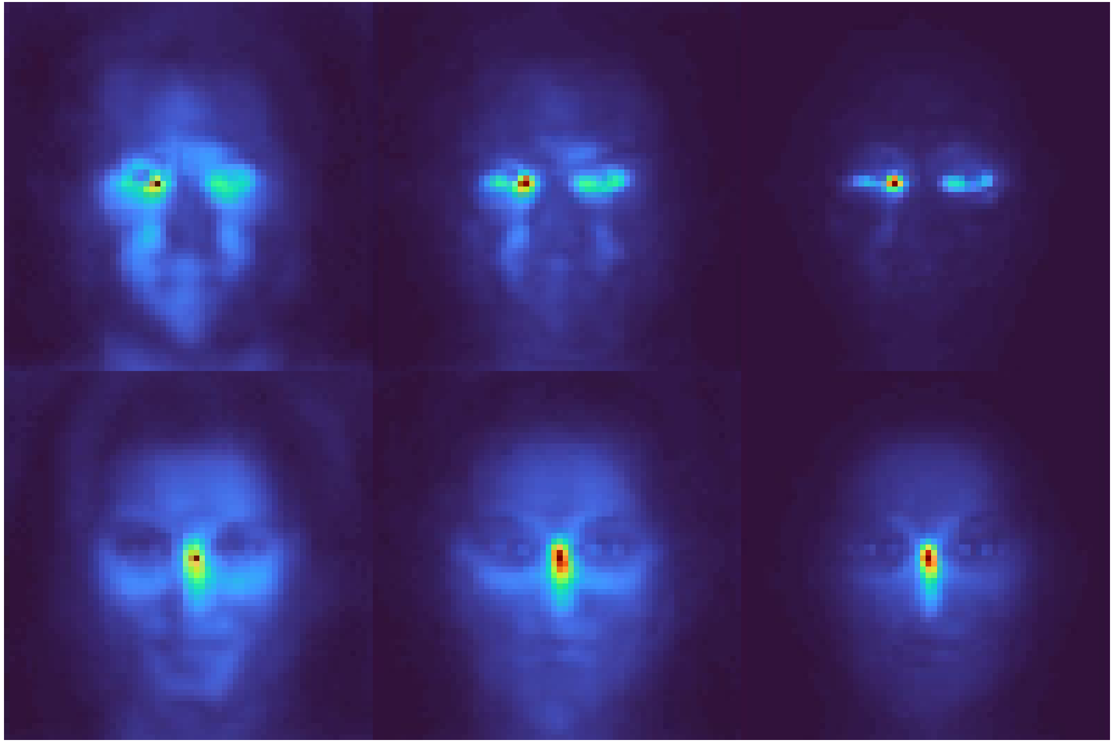
Example of non-local receptive field[6]



[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Example of non-local receptive field[6]

Receptive field for the pixel in the eye: non local



[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

- Generalize the locality constraints M_t^q , to "correlation" constraints

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

- Generalize the locality constraints M_t^q , to "correlation" constraints
- Learn these correlation constraints M_t^q using a simple linear denoiser/velocity

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

- Generalize the locality constraints M_t^q , to "correlation" constraints
- Learn these correlation constraints M_t^q using a simple linear denoiser/velocity

Main Idea 2: linear score/velocity

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

- Generalize the locality constraints M_t^q , to "correlation" constraints
- Learn these correlation constraints M_t^q using a simple linear denoiser/velocity

Main Idea 2: linear score/velocity

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \| u(tx_1 + (1-t)x_0, t) - (x_1 - x_0) \|^2$$

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Locality constraints: too restrictive?[6]

Main Idea 1: generalize the locality constraints

- Generalize the locality constraints M_t^q, to "correlation" constraints
- Learn these correlation constraints M_t^q using a simple linear denoiser/velocity

Main Idea 2: linear score/velocity

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \left\| u(tx_1 + (1-t)x_0, t) - (x_1 - x_0) \right\|^2$$

What about the simple linear velocity case $u(x,t) = W_t x$?

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Linear Score/Velocity[12]

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Linear Score/Velocity[12]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \| \overbrace{u((1-t)x_0 + t x_1, t)}^{\text{simple case } u(x,t) = W_t x} - (x_1 - x_0) \|^2$$

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Linear Score/Velocity[12]

$$\min_{u} \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \| \overbrace{u((1-t)x_0 + t x_1, t)}^{\text{simple case } u(x,t)=W_t x} - (x_1 - x_0) \|^2$$

Well-known result: Wiener filter

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Linear Score/Velocity[12]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \|\overbrace{u((1-t)x_0 + tx_1, t)}^{\text{simple case } u(x,t)=W_t x} - (x_1 - x_0)\|^2$$

Well-known result: Wiener filter

$$W_t = U \text{diag} \frac{(1-t) - t\lambda_i^2}{(1-t)^2 + t^2\lambda_i^2} U^\top \qquad \text{with } \text{Cov}(X) = U \text{diag}(\lambda_1^2, \ldots, \lambda_d^2) U^\top$$

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Linear Score/Velocity[12]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \| \overbrace{u((1-t)x_0 + tx_1, t)}^{\text{simple case } u(x,t) = W_t x} - (x_1 - x_0) \|^2$$

Well-known result: Wiener filter

$$W_t = U \text{diag} \frac{(1-t) - t\lambda_i^2}{(1-t)^2 + t^2 \lambda_i^2} U^\top \qquad \text{with } \text{Cov}(X) = U \text{diag}(\lambda_1^2, \ldots, \lambda_d^2) U^\top$$

For each time t and pixel q, the most predictive pixels are given by the q-th row of W_t

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Linear Score/Velocity[12]

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} \| \overbrace{u((1-t)x_0 + tx_1, t)}^{\text{simple case } u(x,t) = W_t x} - (x_1 - x_0) \|^2$$

Well-known result: Wiener filter

$$W_t = U \text{diag} \frac{(1-t) - t\lambda_i^2}{(1-t)^2 + t^2 \lambda_i^2} U^\top \qquad \text{with } \text{Cov}(X) = U \text{diag}(\lambda_1^2, \ldots, \lambda_d^2) U^\top$$

For each time t and pixel q, the most predictive pixels are given by the q-th row of W_t

The masks M_t^q are then chosen as a binarization of the q-th row of the Wiener filter W_t

[12] B. Wang et al., **The unreasonable effectiveness of Gaussian score approximation for diffusion models and its applications,** In: TMLR, 2024.

Combining all the building blocks[6]

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Combining all the building blocks[6]

W_t : Wiener filter

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Combining all the building blocks[6]

W_t : Wiener filter

$$W_t = U\mathrm{diag}\frac{(1-t) - t\lambda_i^2}{(1-t)^2 + t^2\lambda_i^2}U^\top; \mathrm{Cov}(X) = U\mathrm{diag}(\lambda_1^2, \ldots, \lambda_d^2)U^\top$$

$$\min_u \mathbb{E}_{\substack{x_0 \sim p_0 \\ x_1 \sim \hat{p}_{\text{data}} \\ t \sim \mathcal{U}([0,1])}} ||u((1-t)x_0 + tx_1, t) - (x_1 - x_0)||^2$$

$$\text{s.t. } u^q(x,t) = u^q(M_t^q \odot x, t), \text{(Correlation)}$$

$$M_t^q : \text{Binary mask centered on pixel } q$$

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Combining all the building blocks[6]

W_t : Wiener filter

$W_t = U\mathrm{diag}\frac{(1-t)-t\lambda_i^2}{\ldots}U^\top ; \mathrm{Cov}(X) = U\mathrm{diag}(\lambda^2 \ldots \lambda^2)U^\top$

Binary masks M_t^q : binarized q -th row of the Wiener filter W_t

Closed-form solution for the minimizer $u^\star_{\mathrm{Corr.}}$ with correlations

$$u^\star_{\mathrm{Corr.}}(x,t) = \sum_{i=1}^{n} \lambda_i(x,t)\left(\frac{x^{(i)} - x}{1-t}\right)$$

$$\lambda_i(x,t) = \mathrm{softmax}\left(-\frac{\|M_t^q \odot (tx^{(j)} - x)\|^2}{2(1-t)^2}\right)_i$$

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Empirical results[6]

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Empirical results[6]

[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

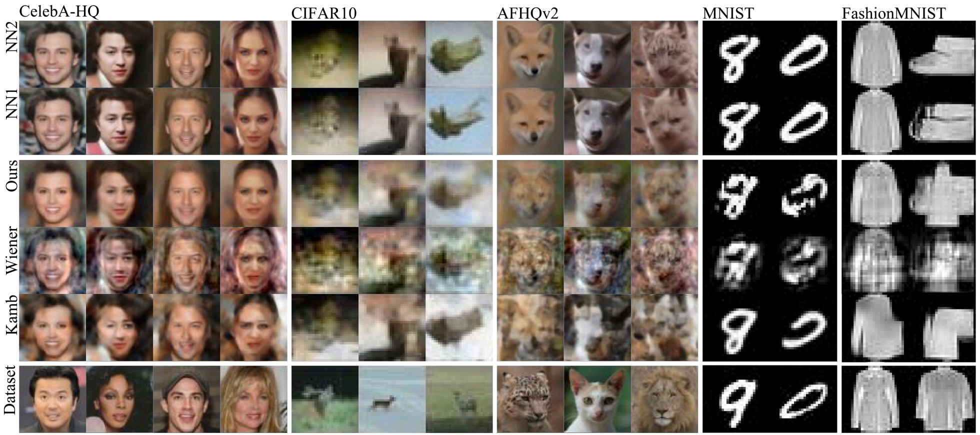
Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints
  - Generalization of locality to correlation constraints

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints
  - Generalization of locality to correlation constraints
- Closed-form formula for the constrained problem



[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints
  - Generalization of locality to correlation constraints
- Closed-form formula for the constrained problem
  - Empirically "match well" small U-Nets

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

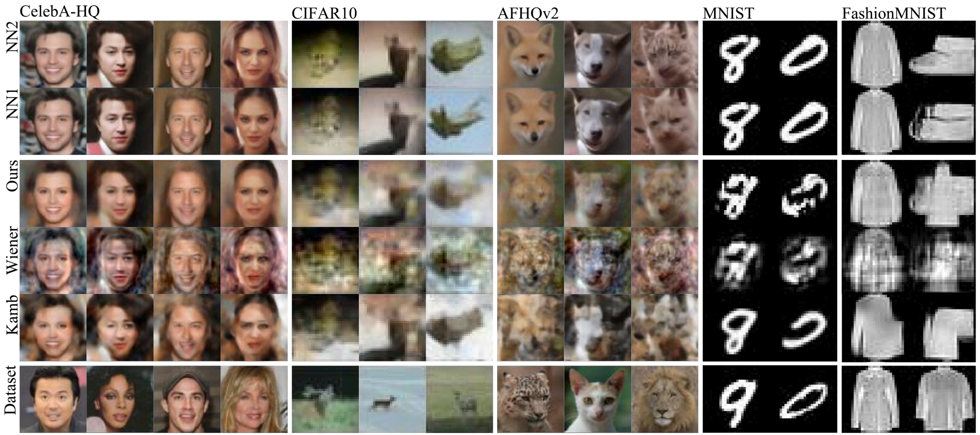
Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints
  - Generalization of locality to correlation constraints
- Closed-form formula for the constrained problem
  - Empirically "match well" small U-Nets
  - Gives intuition on how small diffusion models works

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

Summary and Conclusion[4,5,6,7,8]

- Explicit regularization of the score/velocity
  - Equivariance
  - Locality constraints
  - Generalization of locality to correlation constraints
- Closed-form formula for the constrained problem
  - Empirically "match well" small U-Nets
  - Gives intuition on how small diffusion models works



Thank you for your attention!

[4] M. Kamb et al., **An Analytic Theory of Creativity in Convolutional Diffusion Models,** In: ICML, 2025.
[5] M. Niedoba et al., **Towards a mechanistic explanation of diffusion model generalization,** In: ICML, 2025.
[6] A. Lukoianov et al., **Locality in Image Diffusion Models Emerges From Data Statistics,** In: NeurIPS, 2025.
[7] Z. Kadkhodaie et al., **Generalization in Diffusion Models Arises from Geometry-Adaptive Harmonic Representations,** In: ICLR, 2024.
[8] Q. Bertrand et al., **On the Closed-Form of Flow Matching: Generalization Does Not Arise from Target Stochasticity,** In: NeurIPS, 2025.

ICML 2026 tutorial

Diffusion and Flow Matching

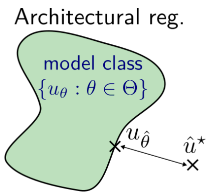
Part IV: Open Questions & Discussions

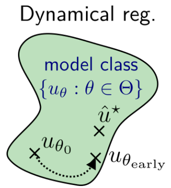
Quentin Bertrand & Mathurin Massias

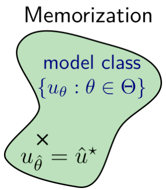
https://memorization-generalization.github.io

Inria | ENS Lyon | Laboratoire Hubert Curien | Mila Affiliated Member | CIFAR Global Scholar

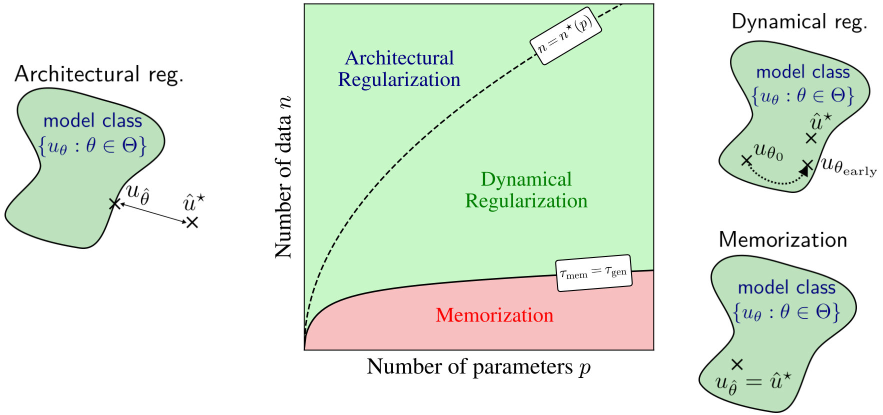
Over- vs under-parameterized models



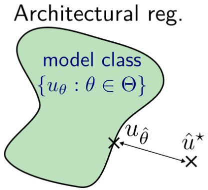


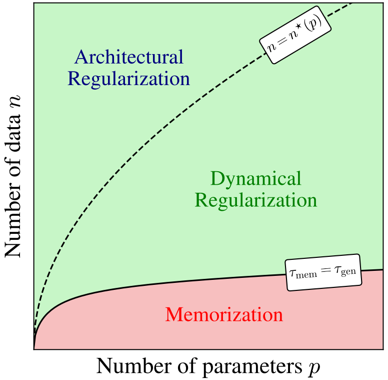


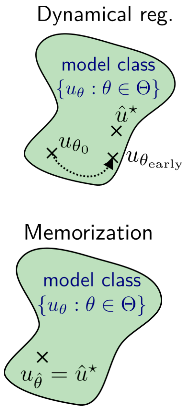
Over- vs under-parameterized models



Over- vs under-parameterized models

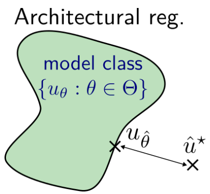


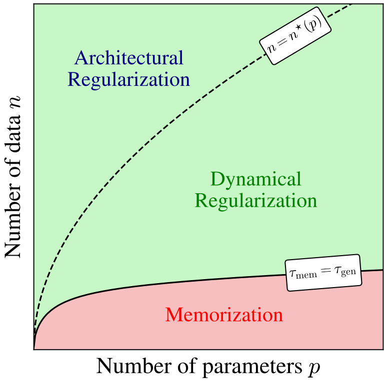


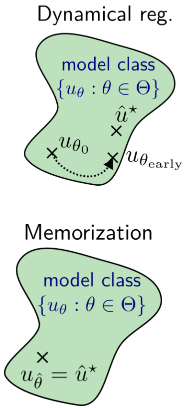


- Which regime are we in?

Over- vs under-parameterized models

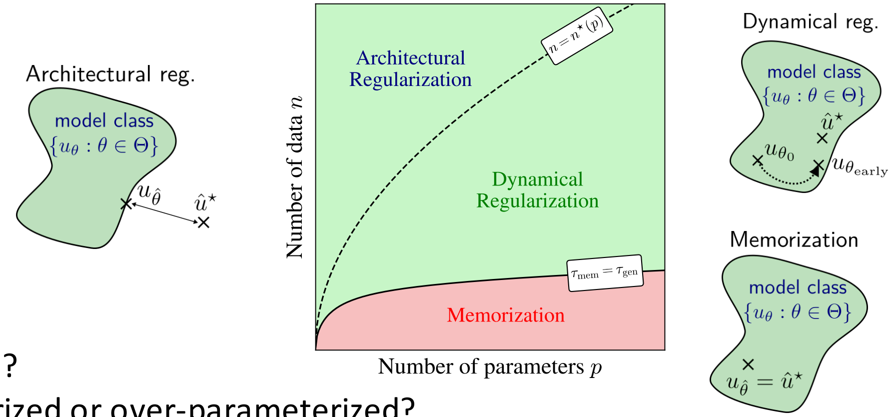






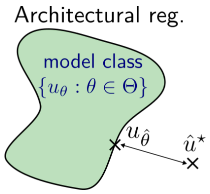
- Which regime are we in?
  - Under-parameterized or over-parameterized?

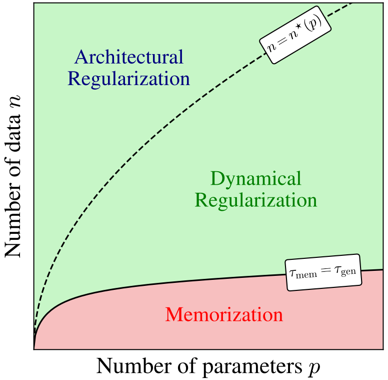
Over- vs under-parameterized models

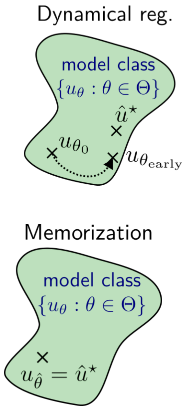


- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?

Over- vs under-parameterized models

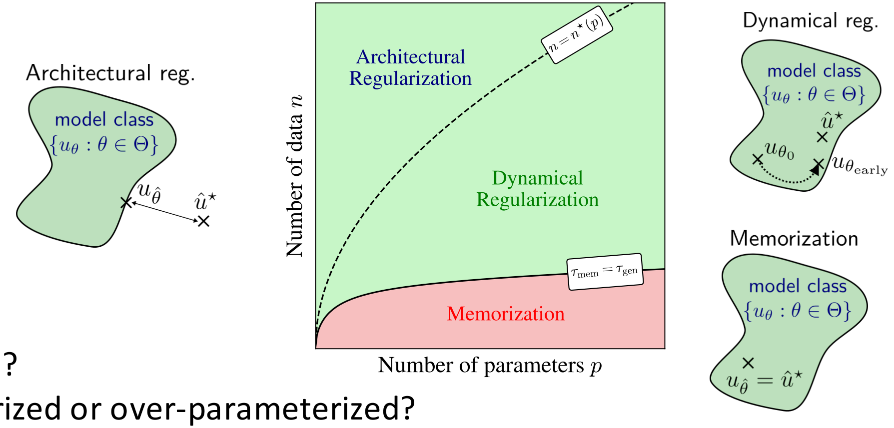






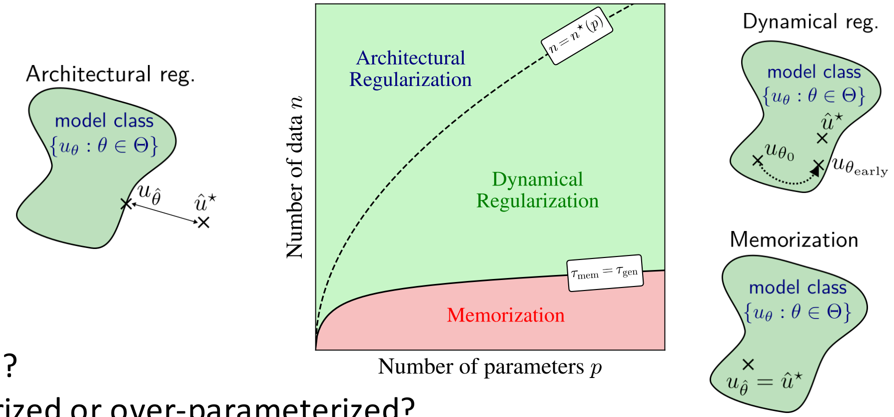
- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?

Over- vs under-parameterized models



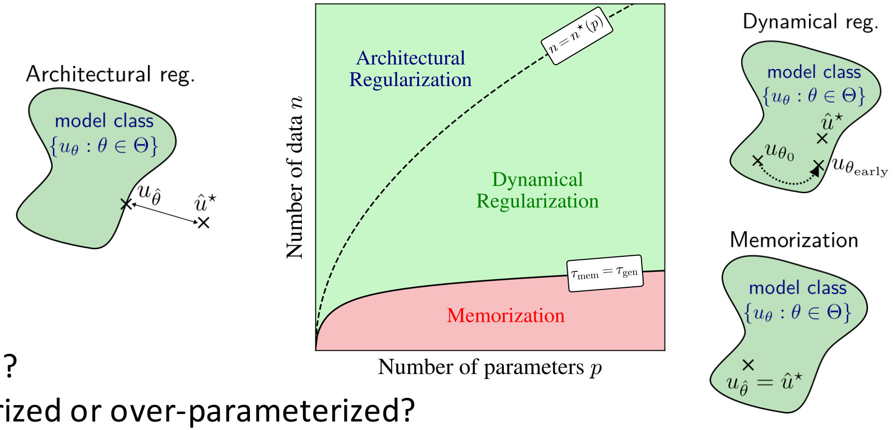
- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:

Over- vs under-parameterized models



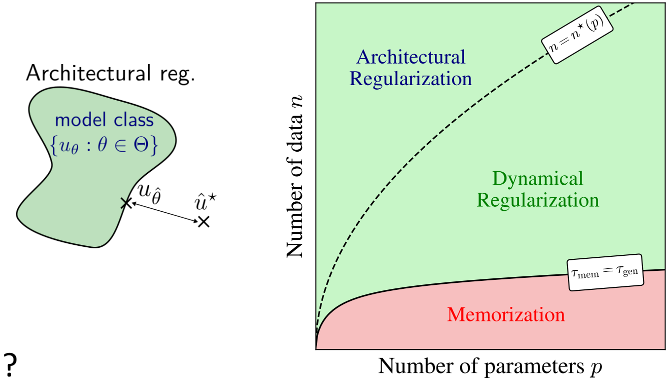
- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:
  - In practice:
    - We do not observe 0 loss

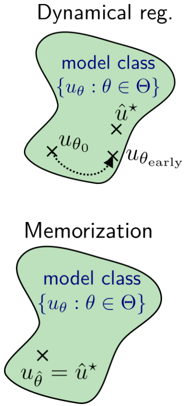
Over- vs under-parameterized models



- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:
  - In practice:
    - We do not observe 0 loss
  - Would the error vanish if we train for longer?

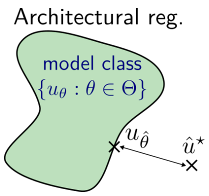
Over- vs under-parameterized models

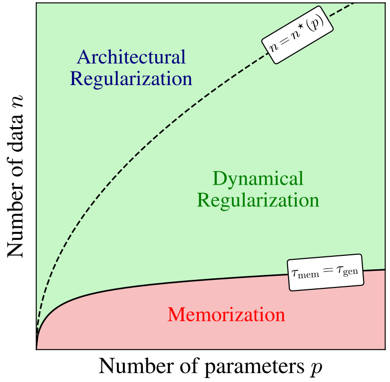


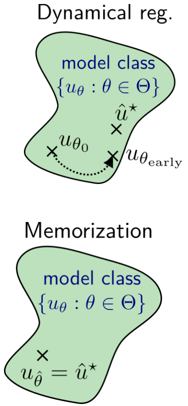


- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:
  - In practice:
    - We do not observe 0 loss
  - Would the error vanish if we train for longer?
- Can we combine the 2 points of view/analyses?

Over- vs under-parameterized models

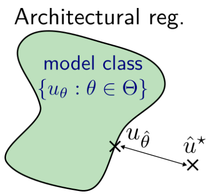


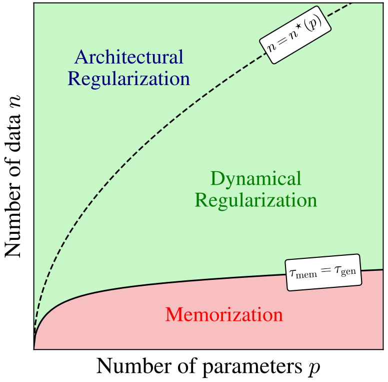


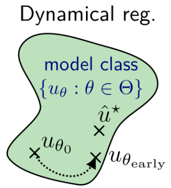


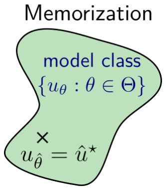
- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:
  - In practice:
    - We do not observe 0 loss
  - Would the error vanish if we train for longer?
- Can we combine the 2 points of view/analyses?
  - Interaction between architectural inductive bias + dynamical regularization of the optimization?

Over- vs under-parameterized models

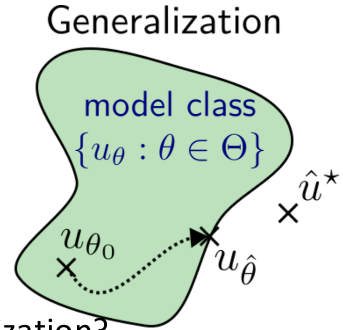




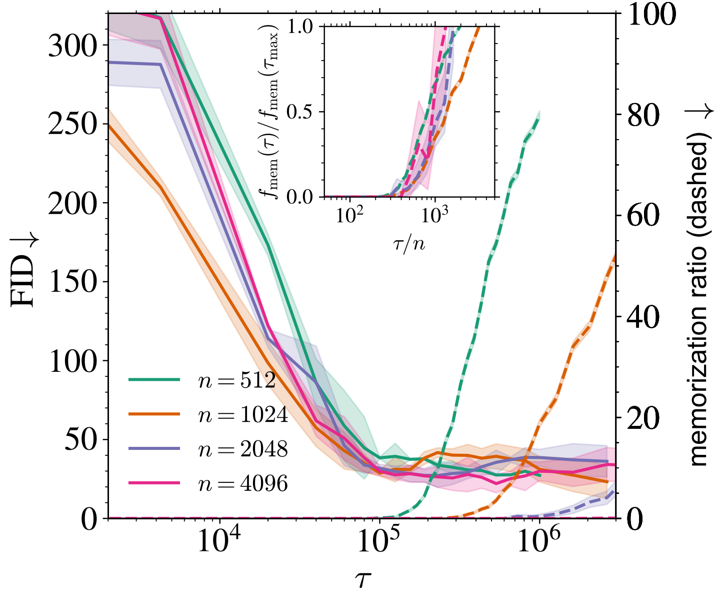




- Which regime are we in?
  - Under-parameterized or over-parameterized?
  - **I.e.** do neural network used **in practice** have enough capacity to achieve 0 training loss?
  - Any empirical evidence?
- Personal experience:
  - In practice:
    - We do not observe 0 loss
  - Would the error vanish if we train for longer?
- Can we combine the 2 points of view/analyses?
  - Interaction between architectural inductive bias + dynamical regularization of the optimization?



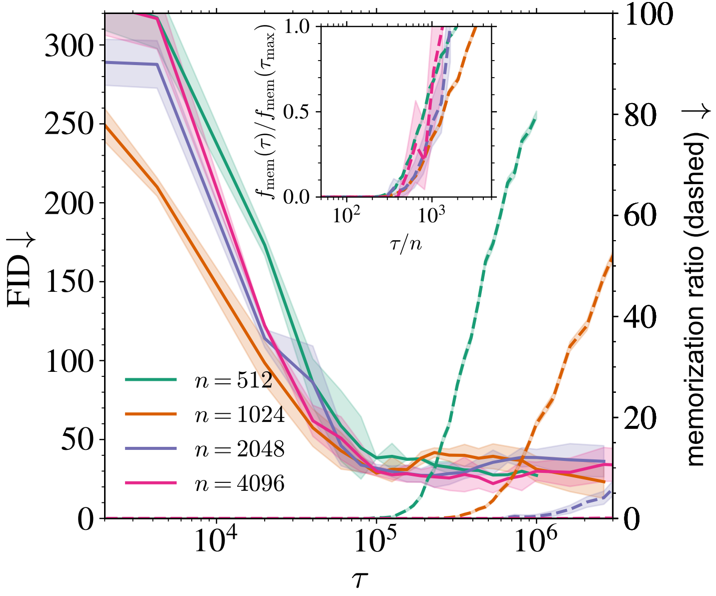
Memorization/Generalization dichotomy?



Bonnaire et al. (2025)

Memorization/Generalization dichotomy?

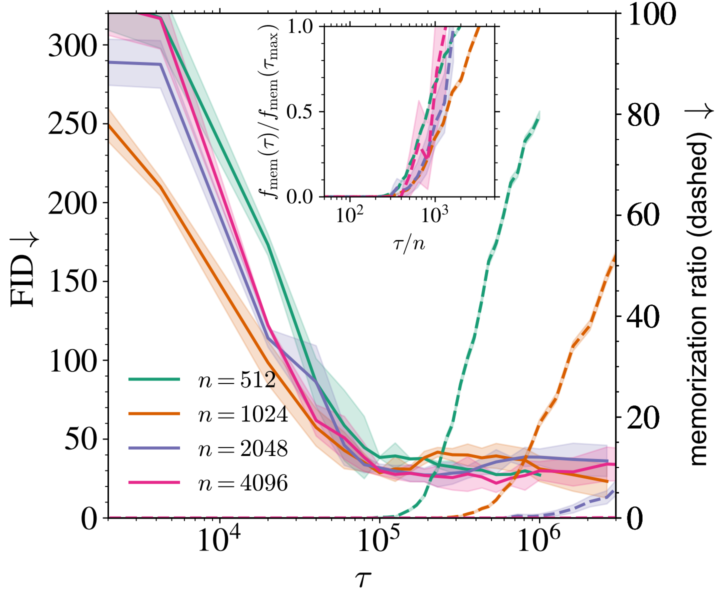
- Currently: transition phase memorization vs generalization



Bonnaire et al. (2025)

Memorization/Generalization dichotomy?

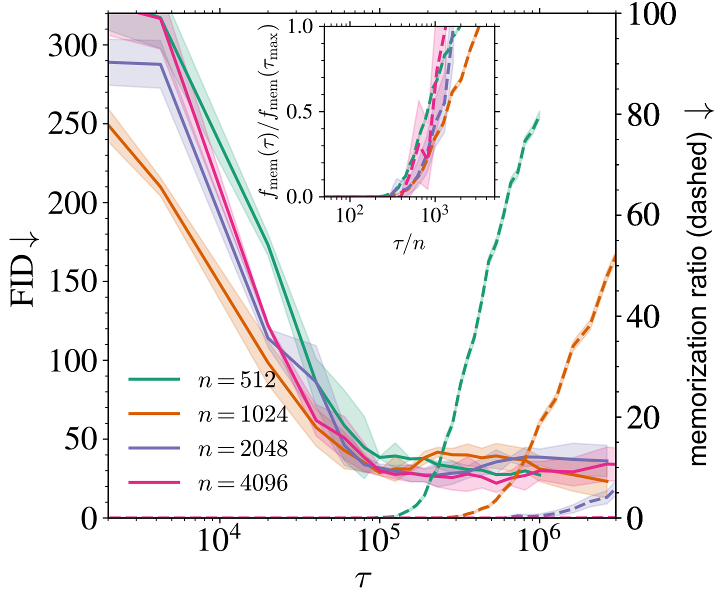
- Currently: transition phase memorization vs generalization
  - In practice? How fast is the transition?



Bonnaire et al. (2025)

Memorization/Generalization dichotomy?

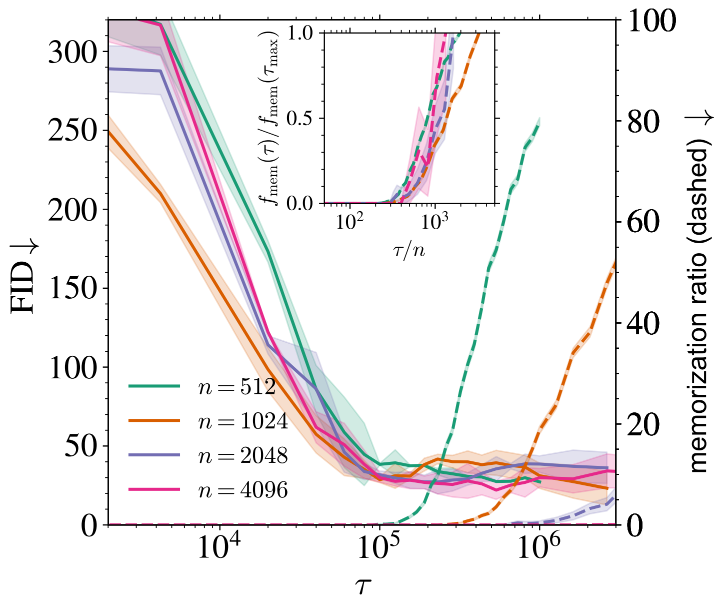
- Currently: transition phase memorization vs generalization
  - In practice? How fast is the transition?
  - More complex than a transition phase?



Bonnaire et al. (2025)

Memorization/Generalization dichotomy?

- Currently: transition phase memorization vs generalization
  - In practice? How fast is the transition?
  - More complex than a transition phase?
  - Which samples are memorized? Easy to memorize?
    - Sample level memorization?
    - What about classes with a few examples?[1]



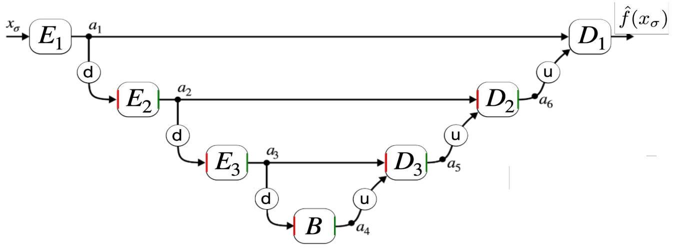
Bonnaire et al. (2025)

[1] C. Merger et al., **Local Coverage Governs Memorization in Diffusion Models,** In: arXiv preprint arXiv:2606.14390, 2026.

Beyond U-Nets?

Beyond U-Nets?

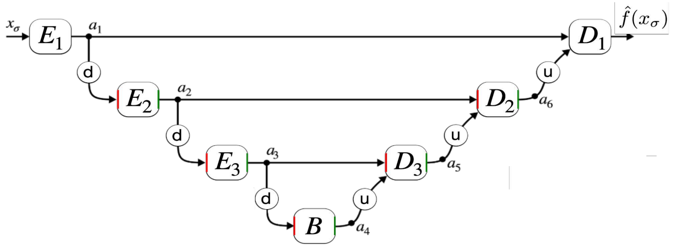
- U-Nets play a crucial in diffusion models for images



Kadkhodaie et al. (2026)

Beyond U-Nets?

- U-Nets play a crucial in diffusion models for images
  - What makes U-Net so special?[2]

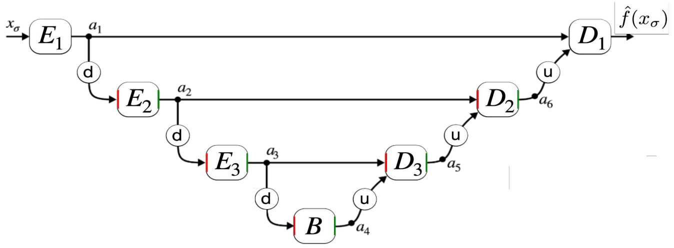


Kadkhodaie et al. (2026)

[2] C. Williams et al., **A unified framework for U-Net design and analysis,** In: NeurIPS, 2023.
[3] Z. Kadkhodaie, **How do diffusion models learn and encode global structure,** In: CVPR Tutorial on Analytical Diffusion Models, 2026.
[4] Z. Kadkhodaie et al., **Learning multi-scale local conditional probability models of images,** In: ICLR, 2023.
[5] W. Peebles et al., **Scalable diffusion models with transformers,** In: ICCV, 2023.
[6] N. Ma et al., **Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers,** In: ECCV, 2024.
[7] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: , 2024.
[8] T. Li et al., **Back to basics: Let denoising generative models denoise,** In: CVPR, 2026.
[9] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: 2nd Workshop on Deep Generative Model in Machine Learning: Theory , 2026.

Beyond U-Nets?

- U-Nets play a crucial in diffusion models for images
  - What makes U-Net so special?[2]
  - Is it possible to characterize more finely the inductive bias of U-Net?[3]

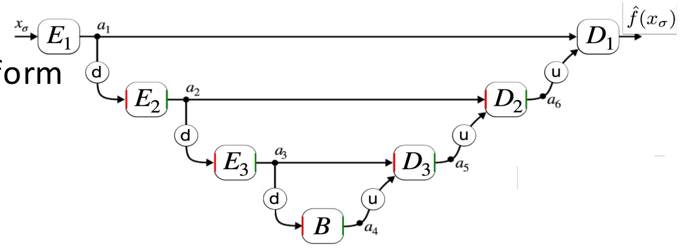


Kadkhodaie et al. (2026)

[2] C. Williams et al., **A unified framework for U-Net design and analysis,** In: NeurIPS, 2023.
[3] Z. Kadkhodaie, **How do diffusion models learn and encode global structure,** In: CVPR Tutorial on Analytical Diffusion Models, 2026.
[4] Z. Kadkhodaie et al., **Learning multi-scale local conditional probability models of images,** In: ICLR, 2023.
[5] W. Peebles et al., **Scalable diffusion models with transformers,** In: ICCV, 2023.
[6] N. Ma et al., **Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers,** In: ECCV, 2024.
[7] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: , 2024.
[8] T. Li et al., **Back to basics: Let denoising generative models denoise,** In: CVPR, 2026.
[9] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: 2nd Workshop on Deep Generative Model in Machine Learning: Theory , 2026.

Beyond U-Nets?

- U-Nets play a crucial in diffusion models for images
  - What makes U-Net so special?[2]
  - Is it possible to characterize more finely the inductive bias of U-Net?[3]
  - How to include the "coarse-to-fine conditional locality" in the closed-form models?[4]

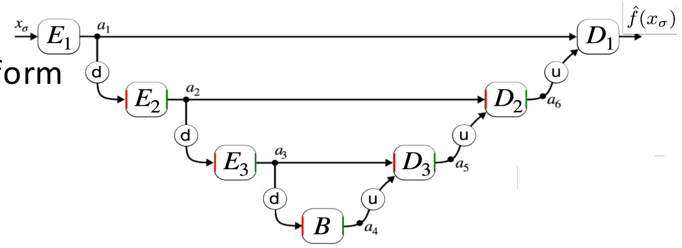


Kadkhodaie et al. (2026)

[2] C. Williams et al., **A unified framework for U-Net design and analysis,** In: NeurIPS, 2023.
[3] Z. Kadkhodaie, **How do diffusion models learn and encode global structure,** In: CVPR Tutorial on Analytical Diffusion Models, 2026.
[4] Z. Kadkhodaie et al., **Learning multi-scale local conditional probability models of images,** In: ICLR, 2023.
[5] W. Peebles et al., **Scalable diffusion models with transformers,** In: ICCV, 2023.
[6] N. Ma et al., **Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers,** In: ECCV, 2024.
[7] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: , 2024.
[8] T. Li et al., **Back to basics: Let denoising generative models denoise,** In: CVPR, 2026.
[9] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: 2nd Workshop on Deep Generative Model in Machine Learning: Theory , 2026.

Beyond U-Nets?

- U-Nets play a crucial in diffusion models for images
  - What makes U-Net so special?[2]
  - Is it possible to characterize more finely the inductive bias of U-Net?[3]
  - How to include the "coarse-to-fine conditional locality" in the closed-form models?[4]
  - What about the inductive bias of transformers, ViT/DiT/SiT?[5,6]

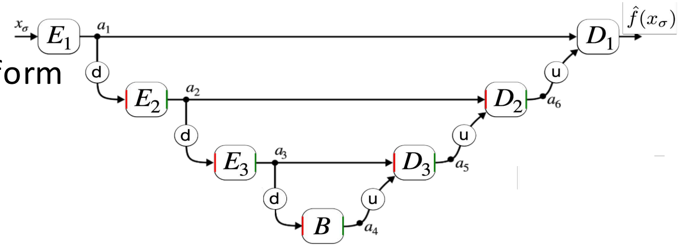


Kadkhodaie et al. (2026)

[2] C. Williams et al., **A unified framework for U-Net design and analysis,** In: NeurIPS, 2023.
[3] Z. Kadkhodaie, **How do diffusion models learn and encode global structure,** In: CVPR Tutorial on Analytical Diffusion Models, 2026.
[4] Z. Kadkhodaie et al., **Learning multi-scale local conditional probability models of images,** In: ICLR, 2023.
[5] W. Peebles et al., **Scalable diffusion models with transformers,** In: ICCV, 2023.
[6] N. Ma et al., **Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers,** In: ECCV, 2024.
[7] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: , 2024.
[8] T. Li et al., **Back to basics: Let denoising generative models denoise,** In: CVPR, 2026.
[9] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: 2nd Workshop on Deep Generative Model in Machine Learning: Theory , 2026.

Beyond U-Nets?

- U-Nets play a crucial in diffusion models for images
  - What makes U-Net so special?[2]
  - Is it possible to characterize more finely the inductive bias of U-Net?[3]
  - How to include the "coarse-to-fine conditional locality" in the closed-form models?[4]
  - What about the inductive bias of transformers, ViT/DiT/SiT?[5,6]
- Impact of parameterization on generalization?[7,8,9]



Kadkhodaie et al. (2026)

[2] C. Williams et al., **A unified framework for U-Net design and analysis,** In: NeurIPS, 2023.
[3] Z. Kadkhodaie, **How do diffusion models learn and encode global structure,** In: CVPR Tutorial on Analytical Diffusion Models, 2026.
[4] Z. Kadkhodaie et al., **Learning multi-scale local conditional probability models of images,** In: ICLR, 2023.
[5] W. Peebles et al., **Scalable diffusion models with transformers,** In: ICCV, 2023.
[6] N. Ma et al., **Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers,** In: ECCV, 2024.
[7] R. Gao et al., **Diffusion Meets Flow Matching: Two Sides of the Same Coin,** In: , 2024.
[8] T. Li et al., **Back to basics: Let denoising generative models denoise,** In: CVPR, 2026.
[9] A. Gagneux et al., **Training flow matching: The role of weighting and parameterization,** In: 2nd Workshop on Deep Generative Model in Machine Learning: Theory , 2026.

Beyond unconditional image generation?

Beyond unconditional image generation?

- What about class/text-conditional generative models?[10]

[10] A. Ramesh et al., **Hierarchical text-conditional image generation with clip latents,** In: arXiv preprint arXiv:2204.06125, 2022.
[11] C. Schuhmann et al., **Laion-5b: An open large-scale dataset for training next generation image-text models,** In: NeurIPS, 2022.
[12] S. Dieleman, **The geometry of diffusion guidance,** In: , 2023.
[13] P. Dhariwal et al., **Diffusion Models Beat GANs on Image Synthesis,** In: Advances in Neural Information Processing Systems, 2021.
[14] J. Ho et al., **Classifier-free diffusion guidance,** In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications:, 2021.
[15] N. Boffi et al., **Flow map matching with stochastic interpolants: A mathematical framework for consistency models,** In: TMLR, 2024.
[16] Z. Geng et al., **Mean flows for one-step generative modeling,** In: NeurIPS, 2025.
[17] A. Müller et al., **Support Before Frequency in Discrete Diffusion,** In: arXiv preprint arXiv:2605.13999, 2026.

Beyond unconditional image generation?

- What about class/text-conditional generative models?[10]
    - Enough capacity to memorize LAION-5B?[11]

[10] A. Ramesh et al., **Hierarchical text-conditional image generation with clip latents,** In: arXiv preprint arXiv:2204.06125, 2022.
[11] C. Schuhmann et al., **Laion-5b: An open large-scale dataset for training next generation image-text models,** In: NeurIPS, 2022.
[12] S. Dieleman, **The geometry of diffusion guidance,** In: , 2023.
[13] P. Dhariwal et al., **Diffusion Models Beat GANs on Image Synthesis,** In: Advances in Neural Information Processing Systems, 2021.
[14] J. Ho et al., **Classifier-free diffusion guidance,** In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications:, 2021.
[15] N. Boffi et al., **Flow map matching with stochastic interpolants: A mathematical framework for consistency models,** In: TMLR, 2024.
[16] Z. Geng et al., **Mean flows for one-step generative modeling,** In: NeurIPS, 2025.
[17] A. Müller et al., **Support Before Frequency in Discrete Diffusion,** In: arXiv preprint arXiv:2605.13999, 2026.

Beyond unconditional image generation?

- What about class/text-conditional generative models?[10]
  - Enough capacity to memorize LAION-5B?[11]
  - Impact of guidance?[12] (classifier-based[13] and classifier-free[14]?)
    - On memorization?
    - Diversity?

[10] A. Ramesh et al., **Hierarchical text-conditional image generation with clip latents,** In: arXiv preprint arXiv:2204.06125, 2022.
[11] C. Schuhmann et al., **Laion-5b: An open large-scale dataset for training next generation image-text models,** In: NeurIPS, 2022.
[12] S. Dieleman, **The geometry of diffusion guidance,** In: , 2023.
[13] P. Dhariwal et al., **Diffusion Models Beat GANs on Image Synthesis,** In: Advances in Neural Information Processing Systems, 2021.
[14] J. Ho et al., **Classifier-free diffusion guidance,** In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications:, 2021.
[15] N. Boffi et al., **Flow map matching with stochastic interpolants: A mathematical framework for consistency models,** In: TMLR, 2024.
[16] Z. Geng et al., **Mean flows for one-step generative modeling,** In: NeurIPS, 2025.
[17] A. Müller et al., **Support Before Frequency in Discrete Diffusion,** In: arXiv preprint arXiv:2605.13999, 2026.

Beyond unconditional image generation?

- What about class/text-conditional generative models?[10]
  - Enough capacity to memorize LAION-5B?[11]
  - Impact of guidance?[12] (classifier-based[13] and classifier-free[14]?)
    - On memorization?
    - Diversity?
- Interaction with one-step/flow maps models[15,16]?

---

[10] A. Ramesh et al., **Hierarchical text-conditional image generation with clip latents,** In: arXiv preprint arXiv:2204.06125, 2022.
[11] C. Schuhmann et al., **Laion-5b: An open large-scale dataset for training next generation image-text models,** In: NeurIPS, 2022.
[12] S. Dieleman, **The geometry of diffusion guidance,** In: , 2023.
[13] P. Dhariwal et al., **Diffusion Models Beat GANs on Image Synthesis,** In: Advances in Neural Information Processing Systems, 2021.
[14] J. Ho et al., **Classifier-free diffusion guidance,** In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications:, 2021.
[15] N. Boffi et al., **Flow map matching with stochastic interpolants: A mathematical framework for consistency models,** In: TMLR, 2024.
[16] Z. Geng et al., **Mean flows for one-step generative modeling,** In: NeurIPS, 2025.
[17] A. Müller et al., **Support Before Frequency in Discrete Diffusion,** In: arXiv preprint arXiv:2605.13999, 2026.

Beyond unconditional image generation?

- What about class/text-conditional generative models?[10]
    - Enough capacity to memorize LAION-5B?[11]
    - Impact of guidance?[12] (classifier-based[13] and classifier-free[14]?)
        - On memorization?
        - Diversity?
- Interaction with one-step/flow maps models[15,16]?
- Memorization/Generalization for discrete diffusion?[17]

---

[10] A. Ramesh et al., **Hierarchical text-conditional image generation with clip latents,** In: arXiv preprint arXiv:2204.06125, 2022.
[11] C. Schuhmann et al., **Laion-5b: An open large-scale dataset for training next generation image-text models,** In: NeurIPS, 2022.
[12] S. Dieleman, **The geometry of diffusion guidance,** In: , 2023.
[13] P. Dhariwal et al., **Diffusion Models Beat GANs on Image Synthesis,** In: Advances in Neural Information Processing Systems, 2021.
[14] J. Ho et al., **Classifier-free diffusion guidance,** In: NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications:, 2021.
[15] N. Boffi et al., **Flow map matching with stochastic interpolants: A mathematical framework for consistency models,** In: TMLR, 2024.
[16] Z. Geng et al., **Mean flows for one-step generative modeling,** In: NeurIPS, 2025.
[17] A. Müller et al., **Support Before Frequency in Discrete Diffusion,** In: arXiv preprint arXiv:2605.13999, 2026.

Hiring students and postdocs in Lyon, France!

Hiring students and postdocs in Lyon, France!

Thank you for your attention!

Interested in discrete diffusion for LLMs? Just released speedrun!

github.com/agonon/speedrun-dlm