few step diffusion modeling

Rabiul Awal
Apr 09, 2026

plan for today

diffusion (we start here)

probability flow ODE (arguably most influential paper in generative modeling in the last 5 years)

distilling pf-ode for fast sampling

consistency models

flow matching

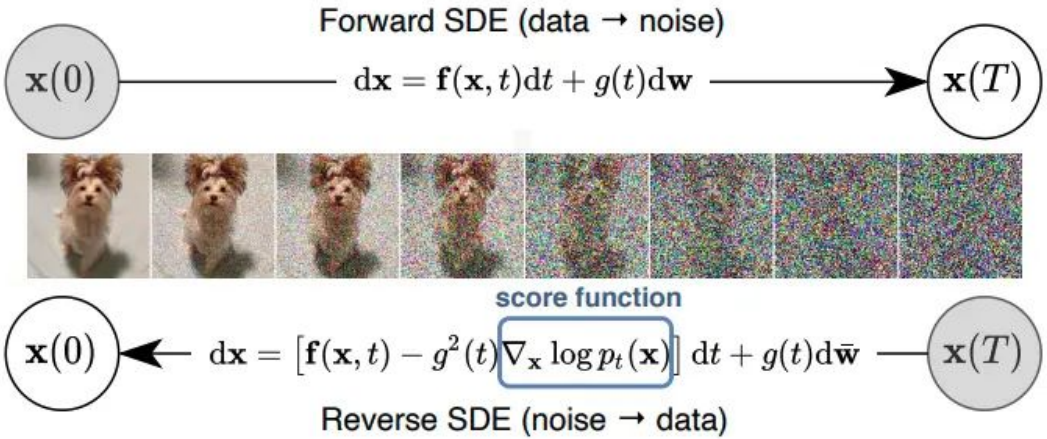
stochastic interpolants (an unifying framework)

flow maps (where things are headed)

diffusion models

Gradually add normal noise to data

Reverse the diffusion process (generative models)



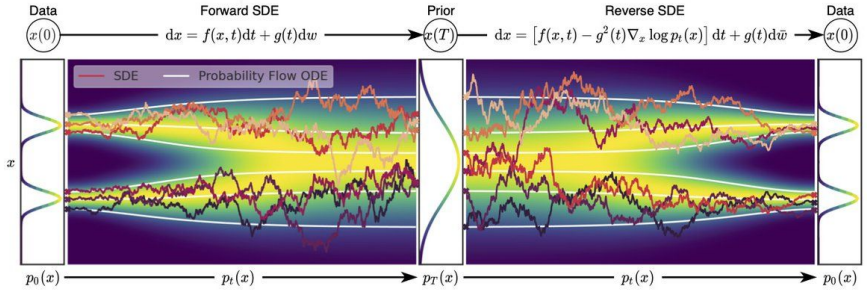
Learn the score function

$$\nabla_x \log p_t(x)$$

Denoising Diffusion Probabilistic Models, Ho et al., 2020
Score-Based Generative Modeling through Stochastic Differential Equations, Song et al., 2021

probability flow ode (pf-ode)

Every SDE has an associated probability flow ODE, which yields deterministic processes that sample from the same distribution as the SDE at each timestep.



pf-ode $$\mathrm{d}x = -\frac{\sigma(t)^2}{2}\nabla_x \log p_t(x)\,\mathrm{d}t$$

Derivation: SDE → Fokker–Planck → rewrite diffusion via score ($\nabla \log p$) → match continuity equation → get ODE with drift correction ($-\frac{1}{2}\, g^2\, \nabla \log p$)

SCORE-BASED GENERATIVE MODELING THROUGH STOCHASTIC DIFFERENTIAL EQUATIONS

Yang Song* Stanford University yangsong@cs.stanford.edu

Jascha Sohl-Dickstein Google Brain jaschasd@google.com

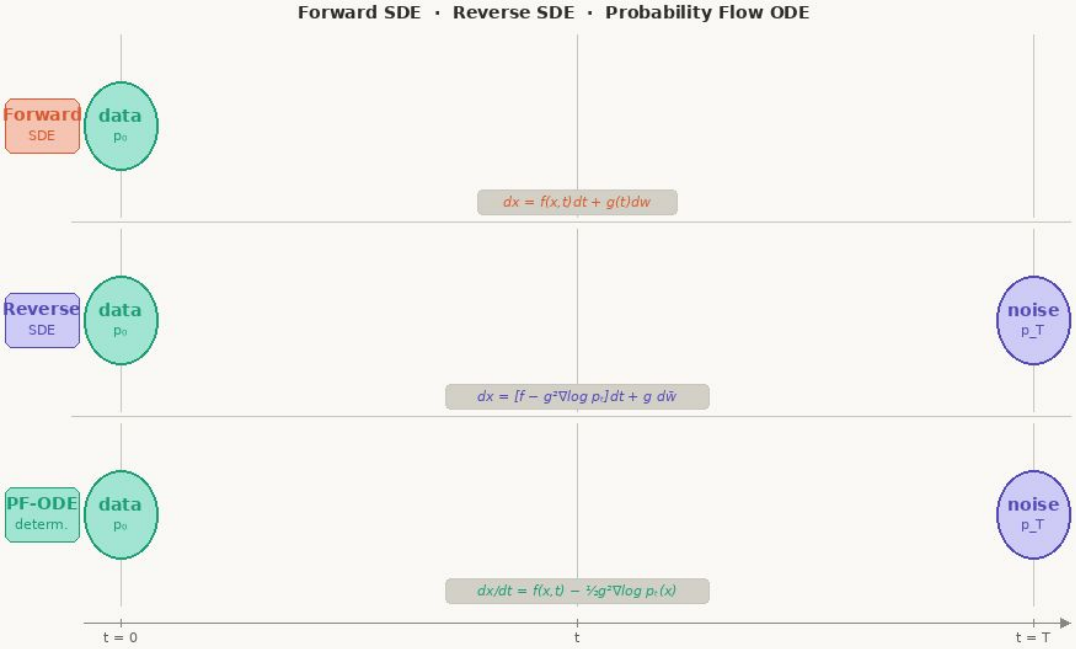
Diederik P. Kingma Google Brain durk@google.com

Abhishek Kumar Google Brain abhishk@google.com

Stefano Ermon Stanford University ermon@cs.stanford.edu

Ben Poole Google Brain pooleb@google.com

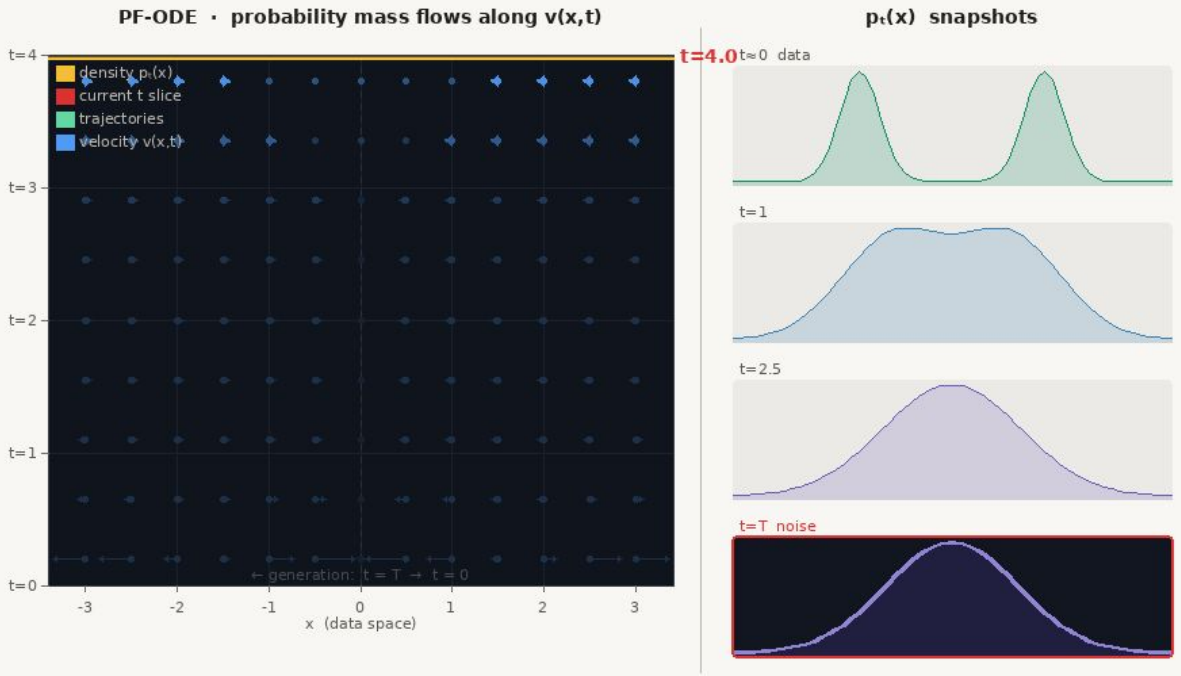
Score-Based Generative Modeling through Stochastic Differential Equations, Song et al., 2021
Elucidating the Design Space of Diffusion-Based Generative Models, Karras et al., 2023


Forward · Reverse · Probability Flow ODE
Forward
SDE
data
p₀
dx = f(x,t)dt + g(t)dw
Reverse
SDE
data
p₀
noise
p_T
dx = [f − g²∇log pₜ]dt + g dw̄
PF-ODE
determ.
data
p₀
noise
p_T
dx/dt = f(x,t) − ½g²∇log pₜ(x)
t = 0
t
t = T

probability mass ~= fluid

ODE is the velocity field that moves this fluid

this is literally called a *flow*



The PF-ODE converts the learned score into a **velocity field dx/dt,** shown as arrows guiding movement at each point.

An ODE solver follows these directions over time, producing the trajectories in the figure that map noise to data.

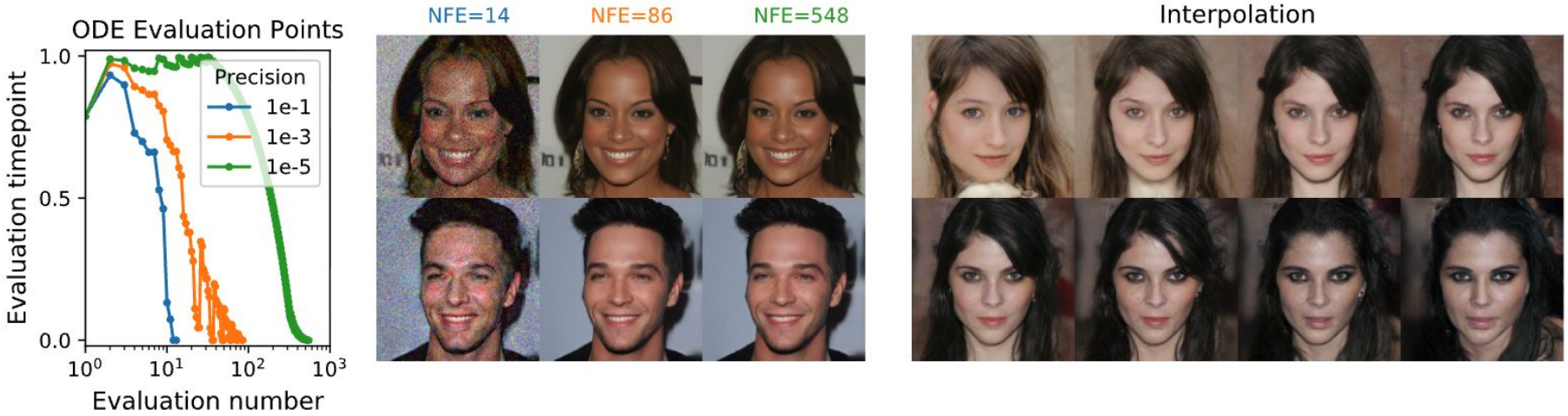
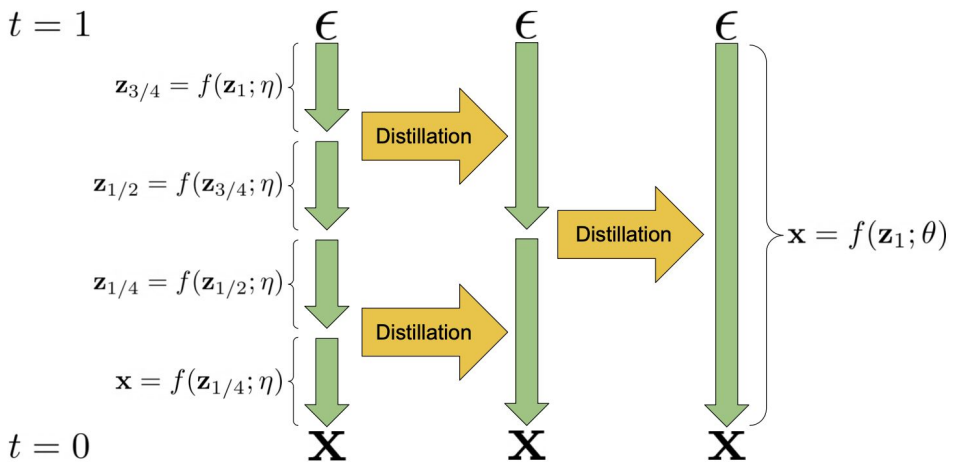


Figure 3: **Probability flow ODE enables fast sampling** with adaptive stepsizes as the numerical precision is varied (*left*), and reduces the number of score function evaluations (NFE) without harming quality (*middle*). The invertible mapping from latents to images allows for interpolations (*right*).

progressive distillation (pf-ode application)

Student model trained to do in 1 step what **teacher** achieves in 2 steps

Applied recursively to drastically reduce the number of steps required



Progressive Distillation for Fast Sampling of Diffusion Models, Salimans & Ho, 2022

progressive distillation (pf-ode application)

Algorithm 1 Standard diffusion training

Require: Model $\hat{\mathbf{x}}_\theta(\mathbf{z}_t)$ to be trained
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$

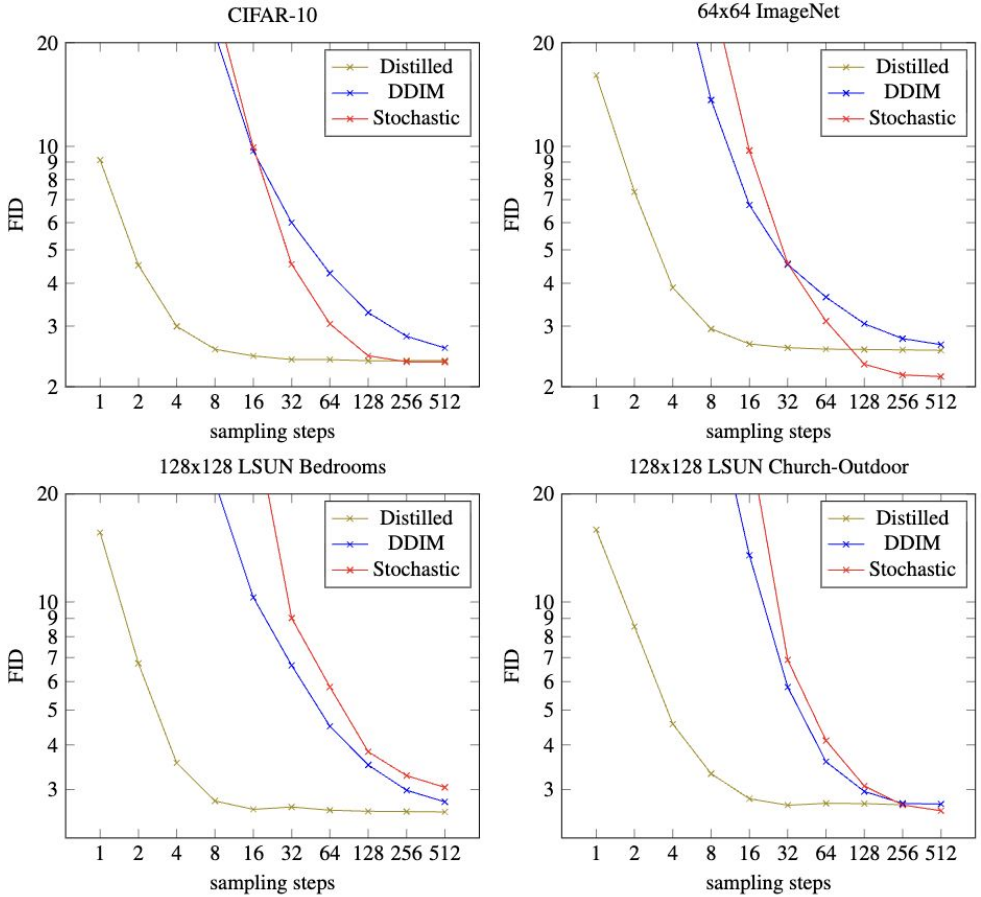
while not converged **do**
  $\mathbf{x} \sim \mathcal{D}$ ▷ Sample data
  $t \sim U[0, 1]$ ▷ Sample time
  $\epsilon \sim N(0, I)$ ▷ Sample noise
  $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$ ▷ Add noise to data

  $\tilde{\mathbf{x}} = \mathbf{x}$ ▷ Clean data is target for $\hat{\mathbf{x}}$
  $\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$ ▷ log-SNR
  $L_\theta = w(\lambda_t) \|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$ ▷ Loss
  $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$ ▷ Optimization
end while

Algorithm 2 Progressive distillation

Require: Trained teacher model $\hat{\mathbf{x}}_\eta(\mathbf{z}_t)$
Require: Data set $\mathcal{D}$
Require: Loss weight function $w()$
Require: Student sampling steps N
for K iterations **do**
  $\theta \leftarrow \eta$ ▷ Init student from teacher
  while not converged **do**
    $\mathbf{x} \sim \mathcal{D}$
    $t = i/N, \;\; i \sim Cat[1, 2, \ldots, N]$
    $\epsilon \sim N(0, I)$
    $\mathbf{z}_t = \alpha_t \mathbf{x} + \sigma_t \epsilon$
    `# 2 steps of DDIM with teacher`
    $t' = t - 0.5/N, \;\; t'' = t - 1/N$
    $\mathbf{z}_{t'} = \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_t) + \frac{\sigma_{t'}}{\sigma_t}(\mathbf{z}_t - \alpha_t \hat{\mathbf{x}}_\eta(\mathbf{z}_t))$
    $\mathbf{z}_{t''} = \alpha_{t''} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}) + \frac{\sigma_{t''}}{\sigma_{t'}}(\mathbf{z}_{t'} - \alpha_{t'} \hat{\mathbf{x}}_\eta(\mathbf{z}_{t'}))$
    $\tilde{\mathbf{x}} = \frac{\mathbf{z}_{t''} - (\sigma_{t''}/\sigma_t)\mathbf{z}_t}{\alpha_{t''} - (\sigma_{t''}/\sigma_t)\alpha_t}$ ▷ Teacher $\hat{\mathbf{x}}$ target
    $\lambda_t = \log[\alpha_t^2 / \sigma_t^2]$
    $L_\theta = w(\lambda_t) \|\tilde{\mathbf{x}} - \hat{\mathbf{x}}_\theta(\mathbf{z}_t)\|_2^2$
    $\theta \leftarrow \theta - \gamma \nabla_\theta L_\theta$
  end while
  $\eta \leftarrow \theta$ ▷ Student becomes next teacher
  $N \leftarrow N/2$ ▷ Halve number of sampling steps
end for

Progressive Distillation for Fast Sampling of Diffusion Models, Salimans & Ho, 2022

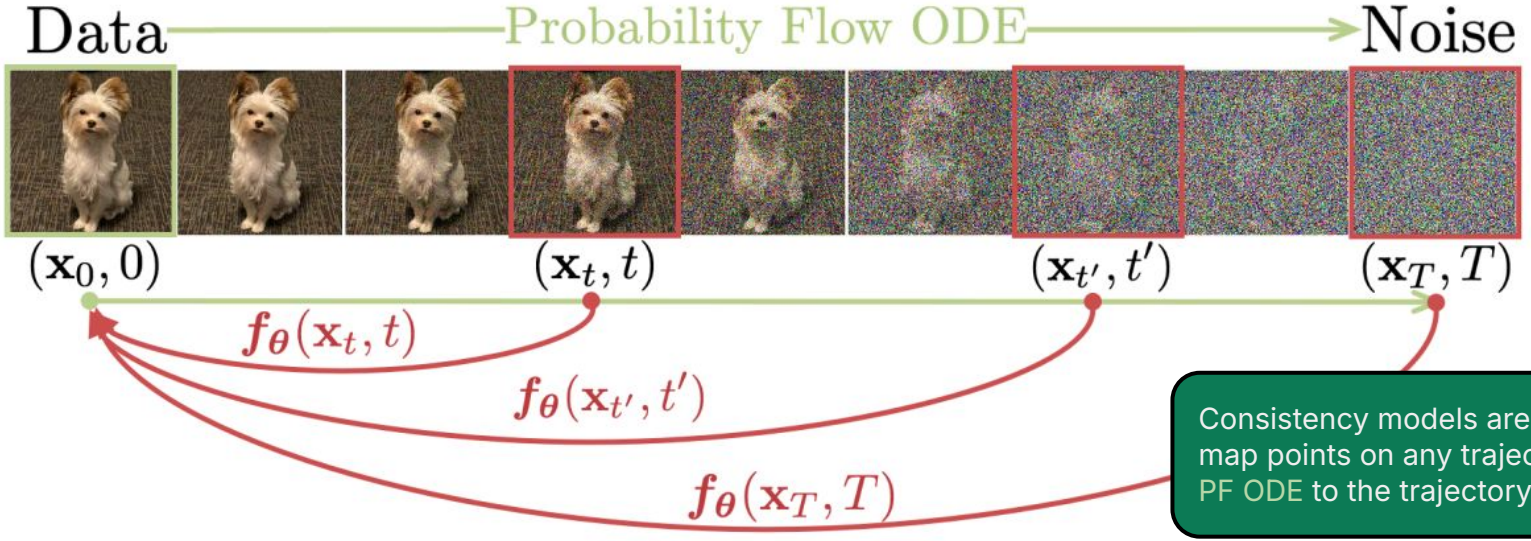


Progressive Distillation for Fast Sampling of Diffusion Models, Salimans & Ho, 2022

properties of pf-ode

1. **invertibility:** mapping from noise → data is one-to-one, unique trajectory (no randomness)
2. **exact likelihood:** process is deterministic, exact data likelihoods (useful for evaluation)
3. **distillation:** a student model can learn to match multi-step trajectories in fewer steps (faster generation)
4. **Trajectory curvature:** probability flow follows curved paths. a single linear step deviates from the true trajectory → leads to large errors in one-step generation

consistency models



Diffusion models takes 100s of function evaluation during sampling

Distill a teacher diffusion model to a few-step student model via endpoint consistency.

Consistency Models, Song et al., 2023

consistency distillation and training

Sample two nearby times *t, t+Δt* on the same PF-ODE trajectory and train the model to output identical predictions for both points

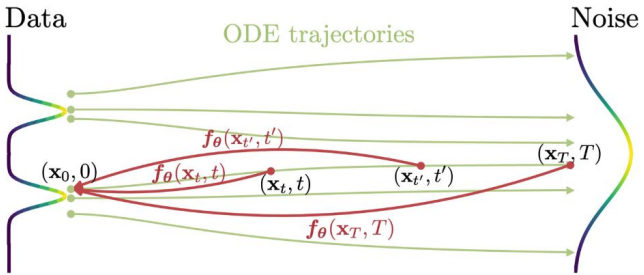


Figure 2: Consistency models are trained to map points on any trajectory of the PF ODE to the trajectory's origin.

In distillation, generate these paired states using a pretrained diffusion/score model (teacher) and match the student's outputs across them

Algorithm 2 Consistency Distillation (CD)

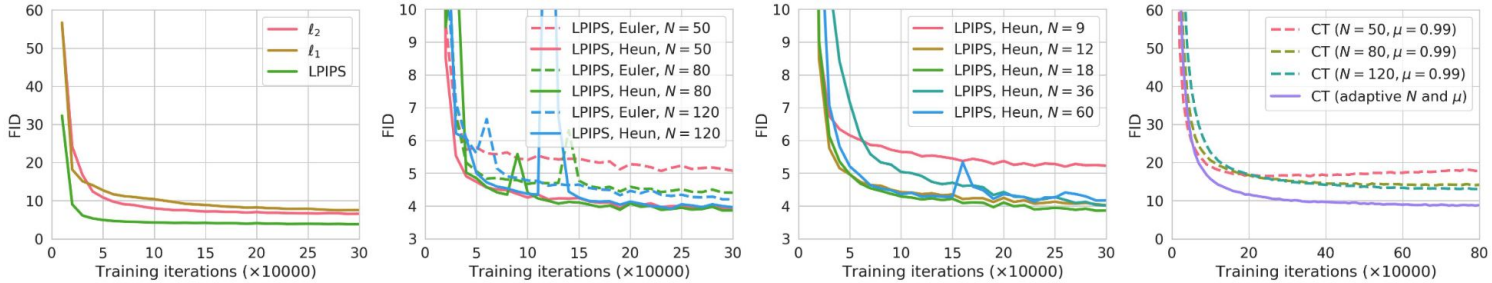
Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, ODE solver $\Phi(\cdot,\cdot;\boldsymbol{\phi})$, $d(\cdot,\cdot)$, $\lambda(\cdot)$, and μ
$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$
repeat
  Sample $\mathbf{x} \sim \mathcal{D}$ and $n \sim \mathcal{U}[\![1, N-1]\!]$
  Sample $\mathbf{x}_{t_{n+1}} \sim \mathcal{N}(\mathbf{x}; t_{n+1}^2 \boldsymbol{I})$
  $\hat{\mathbf{x}}^{\boldsymbol{\phi}}_{t_n} \leftarrow \mathbf{x}_{t_{n+1}} + (t_n - t_{n+1})\Phi(\mathbf{x}_{t_{n+1}}, t_{n+1}; \boldsymbol{\phi})$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi}) \leftarrow$
    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x}_{t_{n+1}}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\hat{\mathbf{x}}^{\boldsymbol{\phi}}_{t_n}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-; \boldsymbol{\phi})$
  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu \boldsymbol{\theta}^- + (1-\mu)\boldsymbol{\theta})$
until convergence

Algorithm 3 Consistency Training (CT)

Input: dataset $\mathcal{D}$, initial model parameter $\boldsymbol{\theta}$, learning rate η, step schedule $N(\cdot)$, EMA decay rate schedule $\mu(\cdot)$, $d(\cdot,\cdot)$, and $\lambda(\cdot)$
$\boldsymbol{\theta}^- \leftarrow \boldsymbol{\theta}$ and $k \leftarrow 0$
repeat
  Sample $\mathbf{x} \sim \mathcal{D}$, and $n \sim \mathcal{U}[\![1, N(k)-1]\!]$
  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
  $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-) \leftarrow$
    $\lambda(t_n) d(\boldsymbol{f}_{\boldsymbol{\theta}}(\mathbf{x} + t_{n+1}\mathbf{z}, t_{n+1}), \boldsymbol{f}_{\boldsymbol{\theta}^-}(\mathbf{x} + t_n \mathbf{z}, t_n))$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\theta}^-)$
  $\boldsymbol{\theta}^- \leftarrow \text{stopgrad}(\mu(k)\boldsymbol{\theta}^- + (1-\mu(k))\boldsymbol{\theta})$
  $k \leftarrow k+1$
until convergence

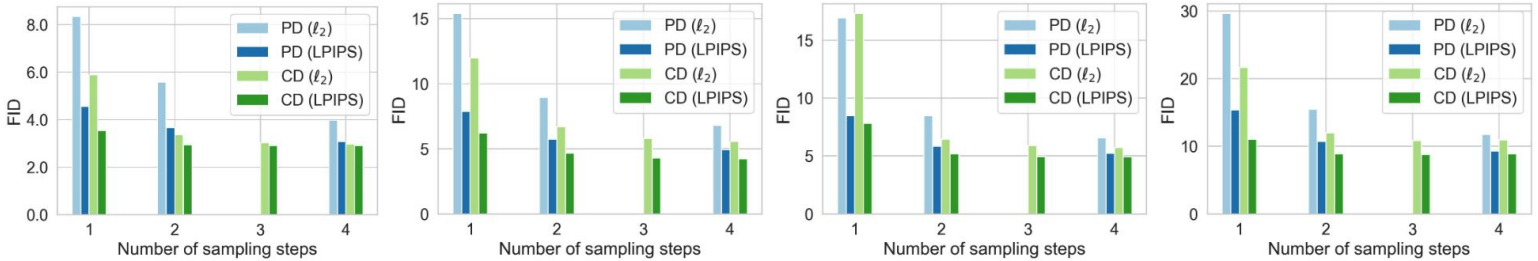
In training, approximate the same pairs by perturbing data with noise and enforce consistency directly, without a teacher

Consistency Models, Song et al., 2023



(a) Metric functions in CD. (b) Solvers and N in CD. (c) N with Heun solver in CD. (d) Adaptive N and μ in CT.

Figure 3: Various factors that affect consistency distillation (CD) and consistency training (CT) on CIFAR-10. The best configuration for CD is LPIPS, Heun ODE solver, and $N = 18$. Our adaptive schedule functions for N and μ make CT converge significantly faster than fixing them to be constants during the course of optimization.



(a) CIFAR-10 (b) ImageNet 64 × 64 (c) Bedroom 256 × 256 (d) Cat 256 × 256

Figure 4: Multistep image generation with consistency distillation (CD). CD outperforms progressive distillation (PD) across all datasets and sampling steps. The only exception is single-step generation on Bedroom 256 × 256.

Table 1: Performance comparisons on CIFAR-10[9].

Model	NFE	Unconditional		Conditional
		FID↓	NLL↓	FID↓
GAN Models				
BigGAN (Brock et al., 2018)	1	8.51	✗	-
StyleGAN-Ada (Karras et al., 2020)	1	2.92	✗	2.42
StyleGAN-D2D (Kang et al., 2021)	1	-	✗	2.26
StyleGAN-XL (Sauer et al., 2022)	1	-	✗	1.85
Diffusion Models – Score-based Sampling				
DDPM (Ho et al., 2020)	1000	3.17	3.75	-
DDIM (Song et al., 2020a)	100	4.16	-	-
	10	13.36	-	-
Score SDE (Song et al., 2020a)	2000	2.20	3.45	-
VDM (Kingma et al., 2021)	1000	7.41	2.49	-
LSGM (Vahdat et al., 2021)	138	2.10	3.43	-
EDM (Karras et al., 2022)	35	2.01	2.56	1.82
Diffusion Models – Distillation Sampling				
KD (Luhman & Luhman, 2021)	1	9.36	✗	-
DFNO (Zheng et al., 2023)	1	3.78	✗	-
2-Rectified Flow (Liu et al., 2022)	1	4.85	✗	-
PD (Salimans & Ho, 2021)	1	9.12	✗	-
CD (official report) (Song et al., 2023)	1	3.55	✗	-
CD (retrained)	1	10.53	✗	-
CD + GAN (Lu et al., 2023)	1	2.65	✗	-
CTM (ours)	1	1.98	**2.43**	1.73
PD (Salimans & Ho, 2021)	2	4.51	-	-
CD (Song et al., 2023)	2	2.93	-	-
CTM (ours)	2	**1.87**	**2.43**	**1.63**
Models without Pre-trained DM – Direct Generation				
CT	1	8.70	✗	-
CTM (ours)	1	2.39	-	-

Table 2: Performance comparisons on ImageNet 64 × 64.

Model	NFE	FID↓	IS↑	Rec↑
Validation Data		1.41	64.10	0.67
ADM (Dhariwal & Nichol, 2021)	250	2.07	-	0.63
EDM (Karras et al., 2022)	79	2.44	48.88	**0.67**
BigGAN-deep (Brock et al., 2018)	1	4.06	-	0.48
StyleGAN-XL (Sauer et al., 2022)	1	2.09	**82.35**	0.52
Diffusion Models – Distillation Sampling				
PD (Salimans & Ho, 2021)	1	15.39	-	0.62
BOOT (Gu et al., 2023)	1	16.3	-	0.36
CD (Song et al., 2023)	1	6.20	40.08	0.63
CTM (ours)	1	1.92	70.38	0.57
PD (Salimans & Ho, 2021)	2	8.95	-	0.65
CD (Song et al., 2023)	2	4.70	-	0.64
CTM (ours)	2	**1.73**	64.29	0.57

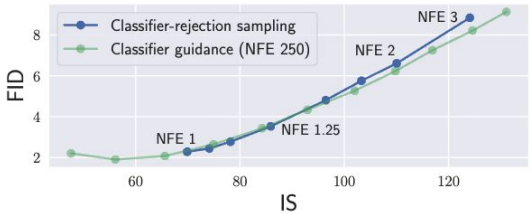


Figure 8: FID-IS curve on ImageNet.

(a) EDM (79 NFE) (b) CTM w/o GAN (1 NFE) (c) CTM w/ GAN (1 NFE)

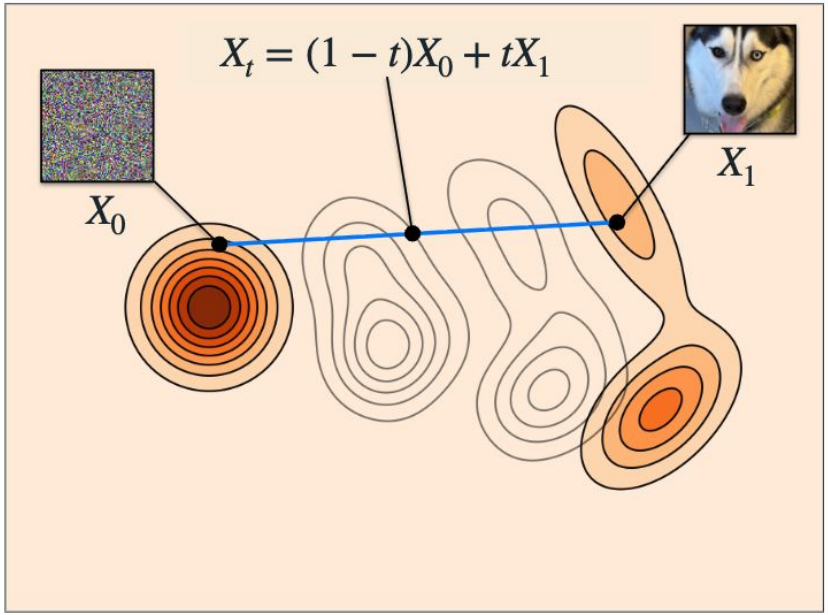
Consistency Trajectory Models: Learning Probability Flow ODE Trajectory of Diffusion, Kim et al., 2023

story so far

- Diffusion learns to reverse noise → data via many small steps
- PF-ODE turns this into a **deterministic flow (velocity field)**
- Distillation + consistency reduce steps → **few-step generation**
- But still rely on **predefined trajectories / teacher models**

Can we instead *directly learn the vector field* that defines these trajectories?

flow matching



$$\mathbb{E}_{t,X_0,X_1}\left\|u_t^{\theta}(X_t) - (X_1 - X_0)\right\|^2$$

Learn a velocity field that transports samples between distributions without relying on precomputed trajectories.

Just sample a pair (x_0, x_1), pick a time t, and train the model to predict the direction from x_t toward x_1

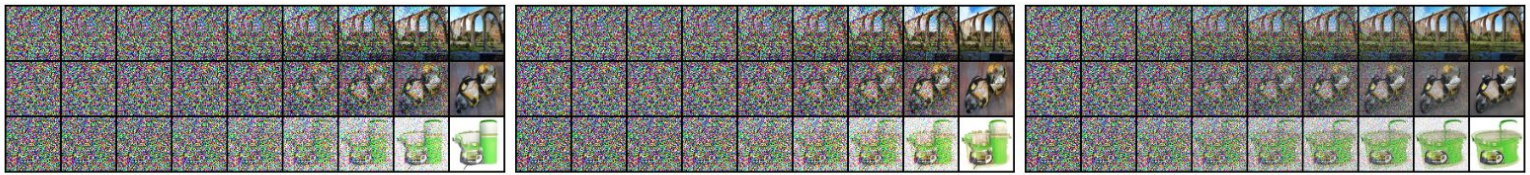
Flow Matching for Generative Modeling, Lipman et al., 2022

generation as a continuous transport

Model	CIFAR-10 NLL↓	CIFAR-10 FID↓	CIFAR-10 NFE↓	ImageNet 32×32 NLL↓	ImageNet 32×32 FID↓	ImageNet 32×32 NFE↓	ImageNet 64×64 NLL↓	ImageNet 64×64 FID↓	ImageNet 64×64 NFE↓
Ablations									
DDPM	3.12	7.48	274	3.54	6.99	262	3.32	17.36	264
Score Matching	3.16	19.94	242	3.56	5.68	178	3.40	19.74	441
ScoreFlow	3.09	20.78	428	3.55	14.14	195	3.36	24.95	601
Ours									
FM ^{w}/ Diffusion	3.10	8.06	183	3.54	6.37	193	3.33	16.88	187
FM ^{w}/ OT	**2.99**	**6.35**	**142**	**3.53**	**5.02**	**122**	**3.31**	**14.45**	**138**

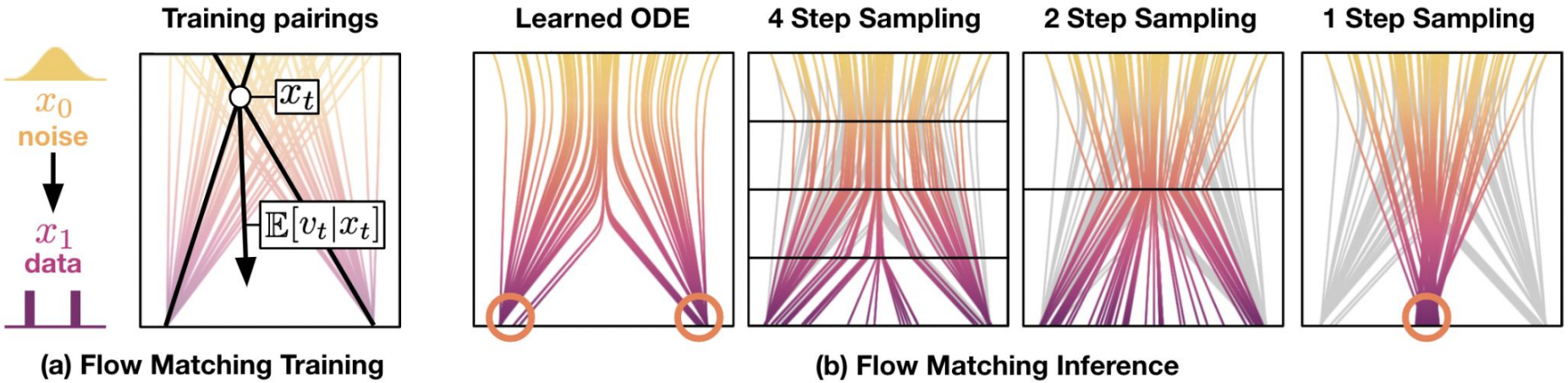
Model	ImageNet 128×128 NLL↓	ImageNet 128×128 FID↓
MGAN (Hoang et al., 2018)	–	58.9
PacGAN2 (Lin et al., 2018)	–	57.5
Logo-GAN-AE (Sage et al., 2018)	–	50.9
Self-cond. GAN (Lučić et al., 2019)	–	41.7
Uncond. BigGAN (Lučić et al., 2019)	–	25.3
PGMGAN (Armandpour et al., 2021)	–	21.7
FM ^{w}/ OT	**2.90**	**20.9**

Table 1: Likelihood (BPD), quality of generated samples (FID), and evaluation time (NFE) for the same model trained with different methods.



Score Matching w/ Diffusion | Flow Matching w/ Diffusion | Flow Matching w/ OT

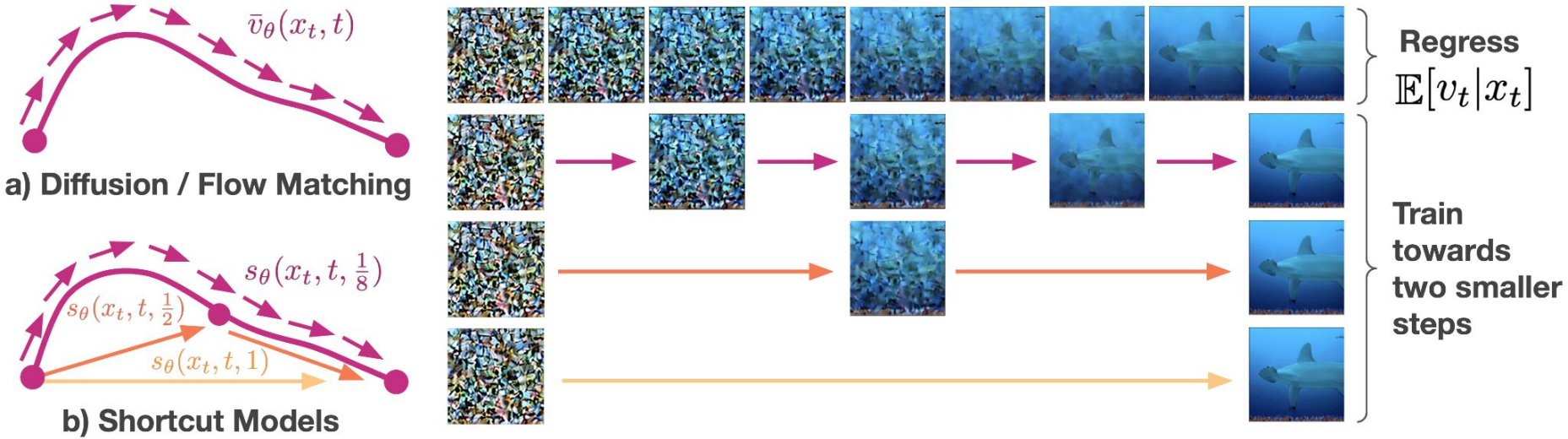
Figure 6: Sample paths from the same initial noise with models trained on ImageNet 64×64. The OT path reduces noise roughly linearly, while diffusion paths visibly remove noise only towards the end of the path. Note also the differences between the generated images.



flow matching inference in few-step has a problem

One-Step Diffusion via Shortcut Models, Frans et al., 2024

shortcut models



Shortcut models enable end-to-end training with step-size and implicit teacher-student distillation.

One-Step Diffusion via Shortcut Models, Frans et al., 2024

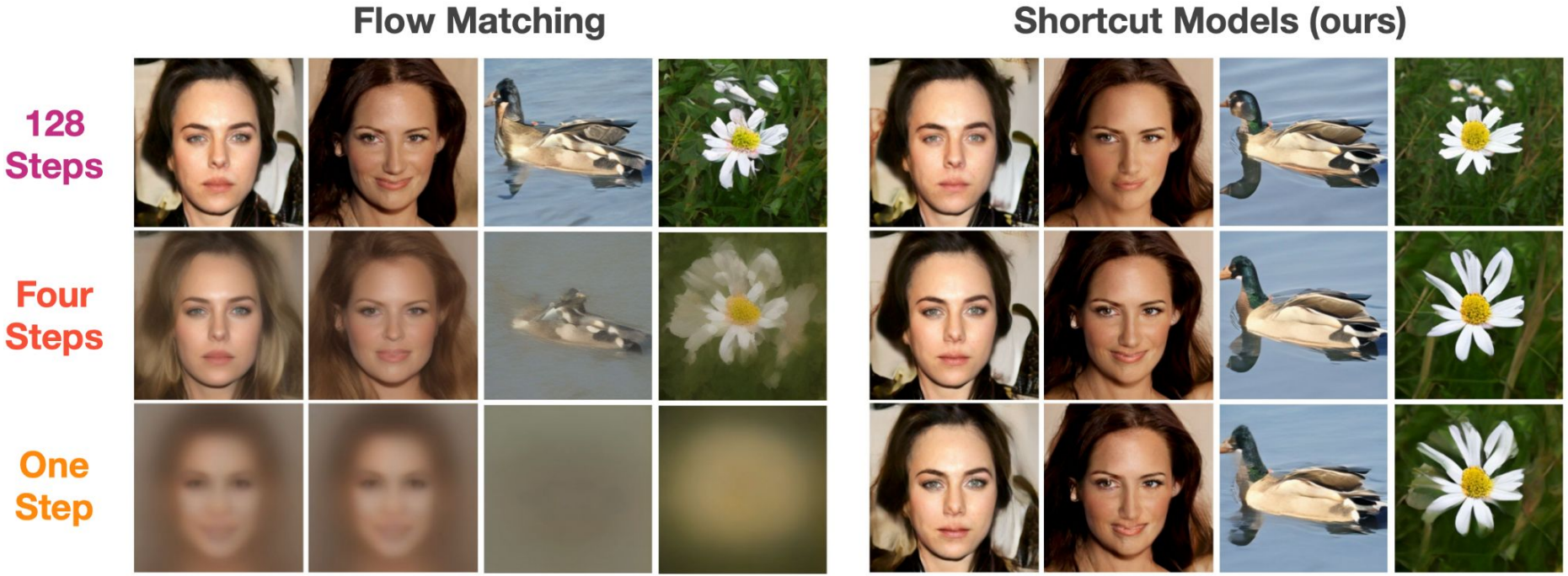
shortcut models

Algorithm 1 Shortcut Model Training

while not converged **do**
  $x_0 \sim \mathcal{N}(0, I),\ x_1 \sim D,\ (d, t) \sim p(d, t)$
  $x_t \leftarrow (1 - t)\, x_0 + t\, x_1$ — Noise data point
  for first k batch elements **do**
    $s_{\text{target}} \leftarrow x_1 - x_0$ — Flow-matching target
    $d \leftarrow 0$
  for other batch elements **do**
    $s_t \leftarrow s_\theta(x_t, t, d)$ — First small step
    $x_{t+d} \leftarrow x_t + s_t\, d$ — Follow ODE
    $s_{t+d} \leftarrow s_\theta(x_{t+d}, t + d, d)$ — Second small step
    $s_{\text{target}} \leftarrow \text{stopgrad}(s_t + s_{t+d})/2$ — Self-consistency target
  $\theta \leftarrow \nabla_\theta ||s_\theta(x_t, t, 2d) - s_{\text{target}}||^2$

Algorithm 2 Sampling

$x \sim \mathcal{N}(0, I)$
$d \leftarrow 1/M$
$t \leftarrow 0$
for $n \in [0, \ldots, M - 1]$ **do**
  $x \leftarrow x + s_\theta(x, t, d)\, d$
  $t \leftarrow t + d$
return x



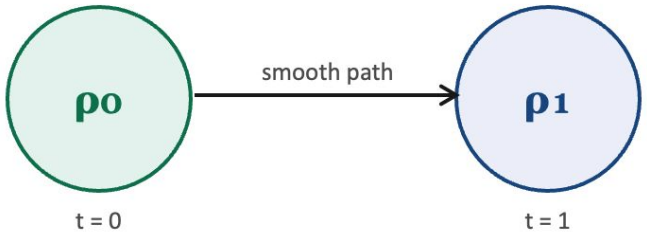
few-step generation is REALLLLL!!!!

One-Step Diffusion via Shortcut Models, Frans et al., 2024

stochastic interpolants (an unifying framework)

Given two arbitrary distributions ρ_0 and ρ_1, **how do we build a smooth path between them?**

ρ_0 can be anything — not just Gaussian noise.



Construct a **random process** x_t that:

$$x_t = \alpha_t x_1 + \beta_t x_0 + \sigma_t z$$

- starts at $x_0 \sim \rho_0$ at $t = 0$
- ends at $x_1 \sim \rho_1$ at $t = 1$
- varies smoothly in between
- can optionally add noise along the path

The choice of path determines everything: the ODE, the training loss, the model family.

Flow Matching, DDPM are all special cases — same framework, different path.

Stochastic interpolants: A unifying framework for flows and diffusions, Albergo et al., 2025

stochastic interpolants (an unifying framework)

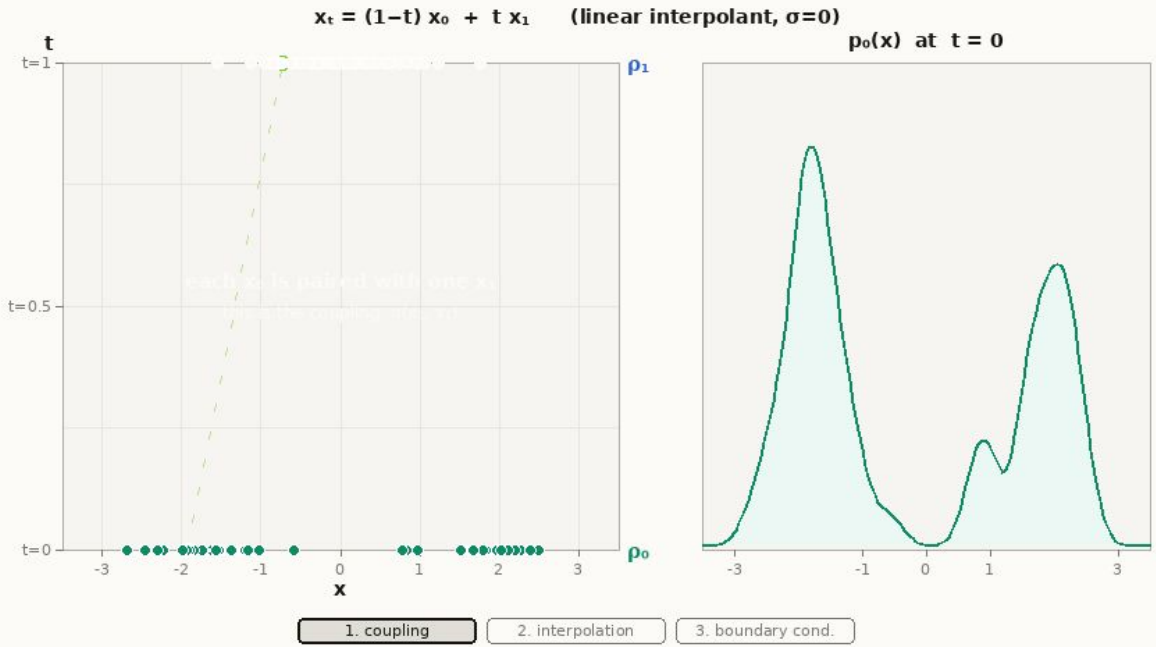
$$x_t = \alpha_t x_1 + \beta_t x_0 + \sigma_t z$$

$z \sim N(0,I)$ · $\alpha_1 = \beta_0 = 1$ · $\alpha_0 = \beta_1 = \sigma_0 = \sigma_1 = 0$ *(boundary conditions)*

Different choices of $(\alpha_t, \beta_t, \sigma_t)$ *recover all known methods as special cases*

method	α_t	β_t	σ_t	source ρ_0	key property
Flow matching	t	1 – t	**0**	any	*straight paths, no noise*
DDPM / VP-SDE	$\sqrt{\bar{\alpha}_t}$	**0**	$\sqrt{(1-\bar{\alpha}_t)}$	N(0,I) fixed	*Gaussian source only*
Schrödinger bridge	t	1 – t	**t(1–t)**	any	*entropy-regularized OT*

Stochastic interpolants: A unifying framework for flows and diffusions, Albergo et al., 2025



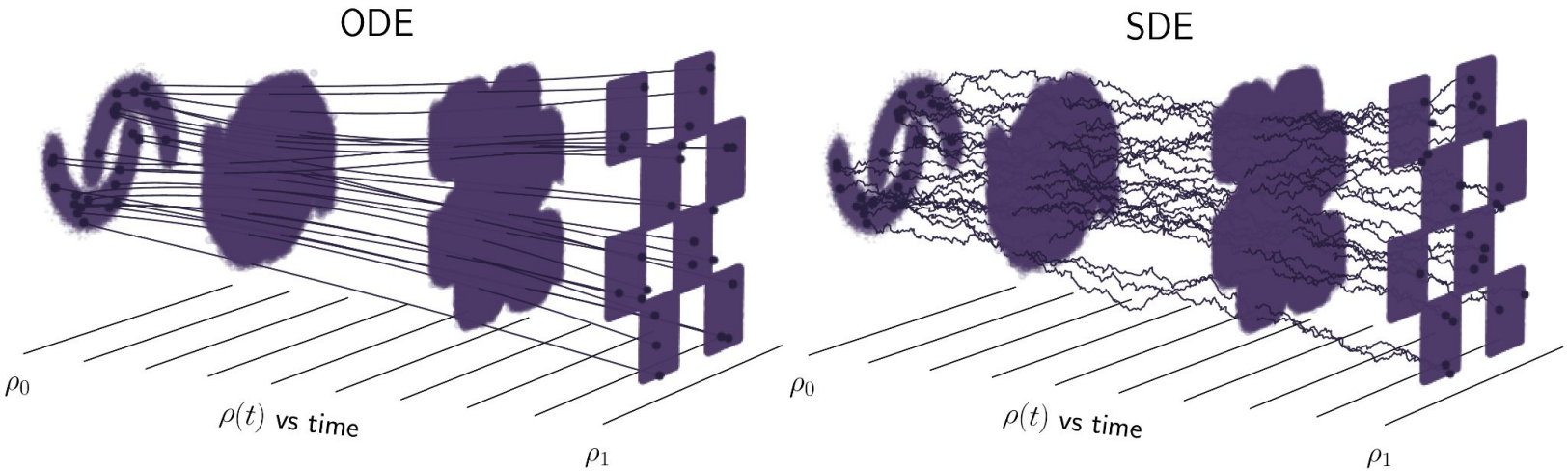
If the path is chosen as linear interpolation, stochastic interpolants generalizes flow-matching!

Stochastic interpolants: A unifying framework for flows and diffusions, Albergo et al., 2025

stochastic interpolants (an unifying framework)

$$x_\square = \alpha_\square x_1 + \beta_\square x_0 + \sigma_\square z$$

$z \sim N(0,I)$ · $\alpha_1 = \beta_0 = 1$ · $\alpha_0 = \beta_1 = \sigma_0 = \sigma_1 = 0$ (boundary conditions)



Both objectives are simulation-free and require no pretrained model — this is what makes the framework practical.

Stochastic interpolants: A unifying framework for flows and diffusions, Albergo et al., 2025

stochastic interpolants (an unifying framework)

$$x_t = \alpha_t x_1 + \beta_t x_0 + \sigma_t z$$

$z \sim N(0,I)$ · $\alpha_1 = \beta_0 = 1$ · $\alpha_0 = \beta_1 = \sigma_0 = \sigma_1 = 0$ *(boundary conditions)*

Velocity field $b_t(x)$ → generates a flow ODE

Regression target is the time-derivative of x_t :

$$\min \; E\| b(x_t, t) - \dot{x}_t \|^2 \quad \text{where} \quad \dot{x}_t = \dot{\alpha}_t x_1 + \dot{\beta}_t x_0 + \dot{\sigma}_t z$$

Target is exact — computed from the pair (x_0, x_1, z).

This is precisely the conditional flow matching loss.

Score field $s_t(x)$ → generates a diffusion SDE

Regression target is the conditional score:

$$\min \; E\| s(x_t, t) + z/\sigma_t \|^2 \quad \text{target:} \; -z/\sigma_t$$

$p(x_t \mid x_0, x_1) = N(\alpha_t x_1 + \beta_t x_0, \; \sigma_t^2 I)$ — Gaussian, so

the conditional score $-z/\sigma_t$ is known in closed form.

Both objectives are simulation-free and require no pretrained model — this is what makes the framework practical.

Stochastic interpolants: A unifying framework for flows and diffusions, Albergo et al., 2025

Fig. 1: Selected samples from SiT-XL models trained on ImageNet [55] at 512×512 and 256×256 resolution with cfg $= 4.0$, respectively.

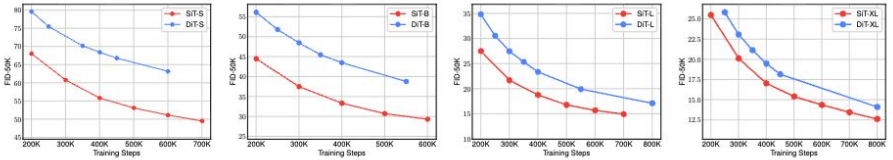
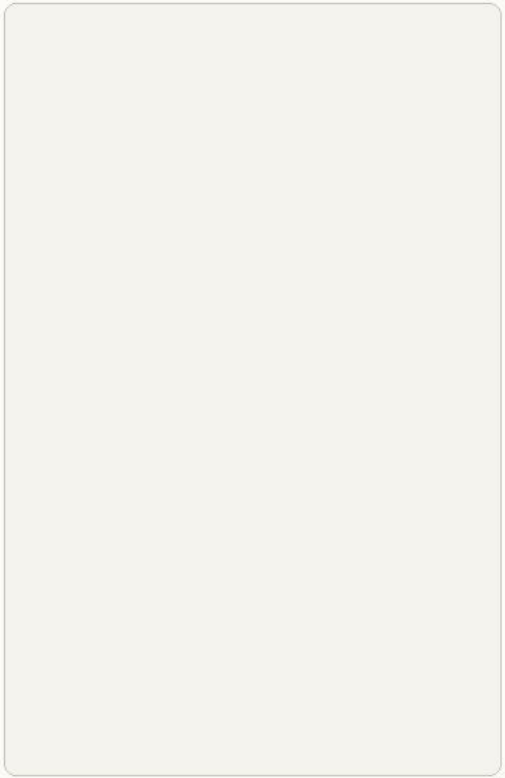


Fig. 2: SiT improves FID across all model sizes. FID-50K over training iterations for both DiT and SiT. All results are produced by a Euler-Maruyama sampler using 250 integration steps. Across all model sizes, SiT converges much faster.

story so far

From stochastic interpolant to flow matching to flow map

concrete example



stochastic interpolant → flow matching → correct transport

flow maps (finally we reached here :)

Probability flow-ode: $\frac{dx}{dt} = b^*(x, t)$

Given a starting point x_s at time s, this ODE has a unique solution at every later (or earlier) time t. You just follow the velocity field forward or backward.

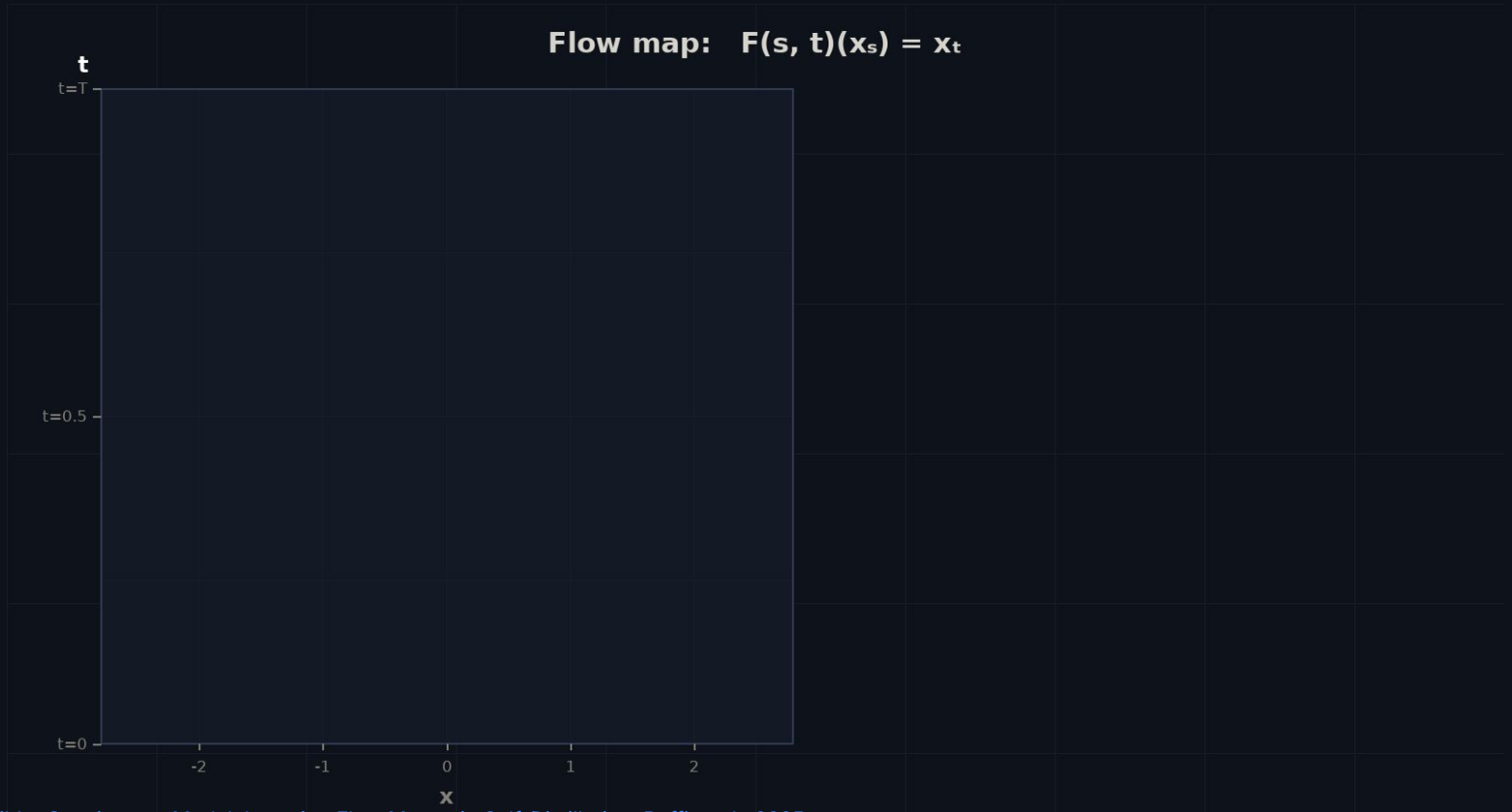
The **flow map** is simply the name for that solution: $F(s,t)(x_s) = x_t$

The flow map $F^*(s,t)$ is the solution operator of the PF-ODE that comes from the interpolant:

F*(s, t)(x□) = x□ ← follow the PF-ODE from time s to time t (jump condition)

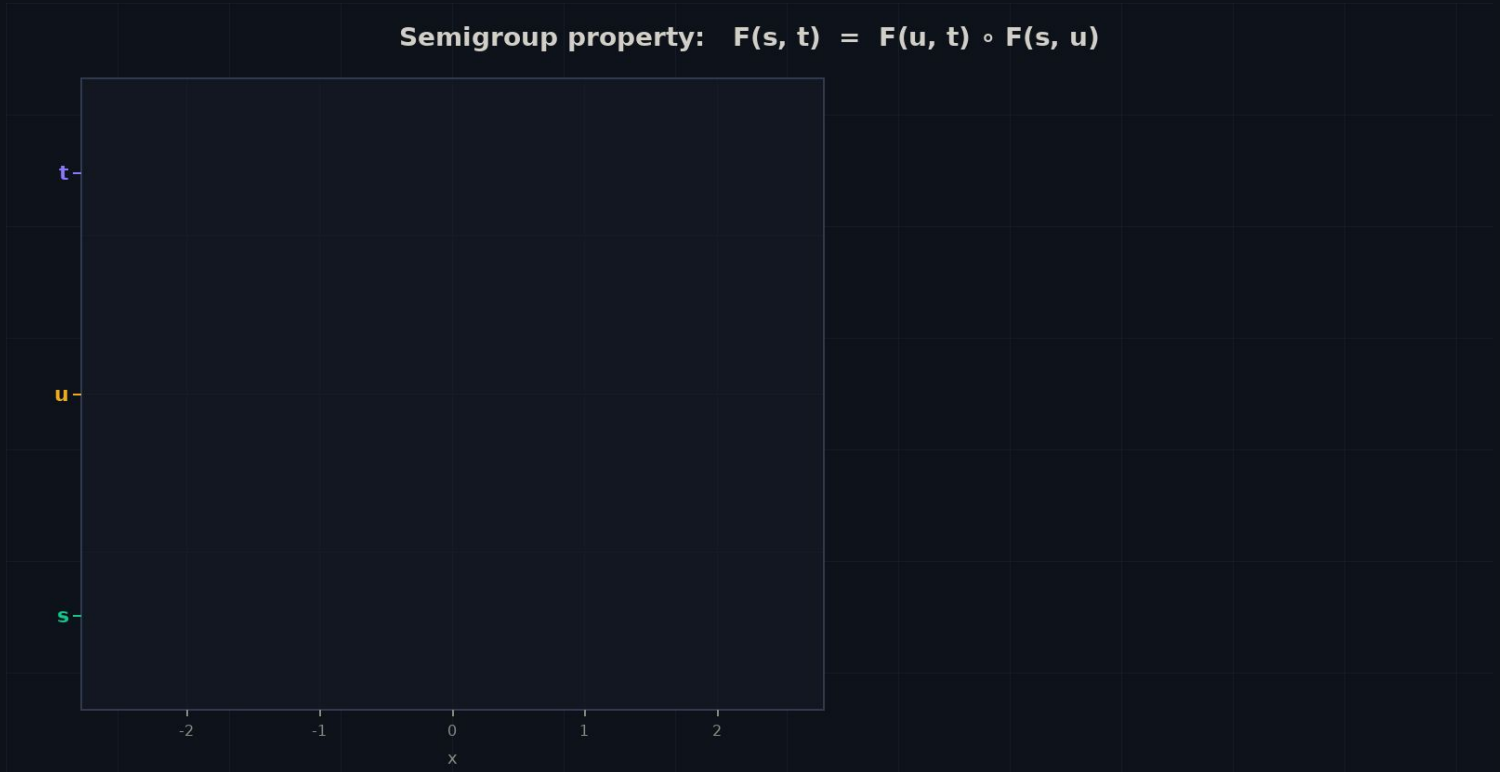
How to Build a Consistency Model: Learning Flow Maps via Self-Distillation, Boffi et al., 2025

flow maps visualization



How to Build a Consistency Model: Learning Flow Maps via Self-Distillation, Boffi et al., 2025

flow maps semigroup condition (self supervised learning)



How to Build a Consistency Model: Learning Flow Maps via Self-Distillation, Boffi et al., 2025

flow maps for few-step generation

Dataset	Method	Step Count				
		1	2	4	8	16
Checker (KL ↓)	LSD	**0.086**	**0.077**	**0.071**	**0.070**	0.071
	ESD	0.098	0.092	0.083	0.082	0.075
	PSD-M	0.146	0.089	0.081	0.072	0.069
	PSD-U	0.111	0.107	0.075	0.073	**0.068**
CIFAR-10 (FID ↓)	LSD	**8.100**	**4.370**	**3.340**	**3.330**	**3.570**
	PSD-M	12.810	8.430	5.960	5.070	4.640
	PSD-U	13.610	7.950	6.030	5.320	5.160
CelebA-64 (FID ↓)	LSD	**12.220**	**5.740**	**3.180**	**2.180**	**1.960**
	PSD-M	19.640	11.750	7.890	6.060	5.090
	PSD-U	18.810	11.020	7.470	6.000	5.630
AFHQ-64 (FID ↓)	LSD	**11.190**	**7.780**	**7.000**	**5.890**	**5.610**
	PSD-M	18.860	14.750	14.400	13.260	11.070
	PSD-U	14.500	10.730	10.990	12.020	11.470

Table 1: Benchmark results. Performance across sampling step counts for the low-dimensional checker dataset (KL divergence) and natural image datasets (FID). Best method per dataset and step count shown in **bold**.

How to Build a Consistency Model: Learning Flow Maps via Self-Distillation, Boffi et al., 2025

Flow maps unify diffusion, flow matching, and consistency models.

Direct, few-step generation by learning the full transport between noise and data.