

Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow

Sina Namazi, Patrick Chen

<https://arxiv.org/abs/2209.03003>



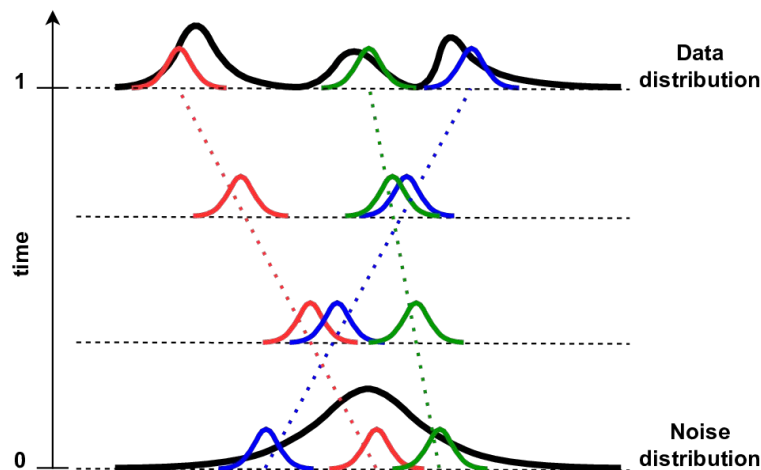
Outline

- Problem statement
- Motivation
- Method
 - Training
 - Reflow
- Properties
 - Non-Crossing Flow
 - Marginal Preserving
 - Distillation
- Comparison
- Experiments & Results
- Strengths & Weaknesses
- Question



The Core Challenge: The Transport Mapping Problem

- **The Mathematical Definition:** Given empirical observations of $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ on \mathbb{R}^d , find a transport map $T : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that $Z_1 := T(Z_0) \sim \pi_1$ when $Z_0 \sim \pi_0$.
- **Generative Modeling:**
 - π_0 : Elementary, tractable distribution
 - π_1 : Complex data distribution
- **Domain Transfer:**
 - π_0 and π_1 : Empirically observed data distributions.



Transport Between Distributions

Key idea

We want to transport samples from one distribution to another.
Like from standard normalization to complex distribution

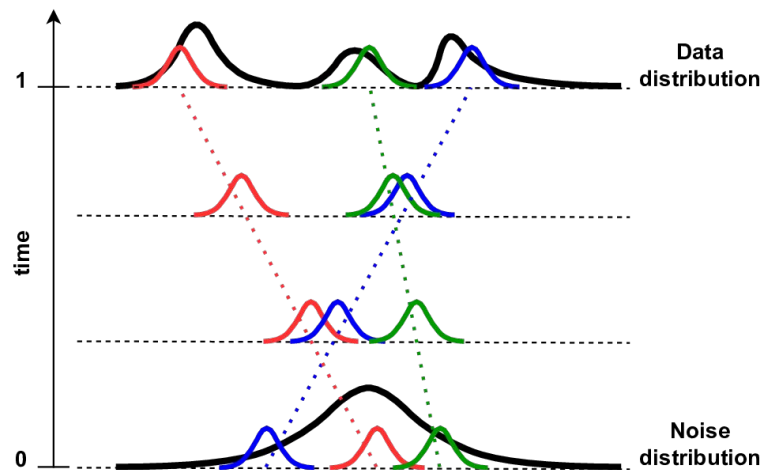
$$X_0 \sim \pi_0, \quad X_1 \sim \pi_1$$

Find transport map (Learning flows)

$$T : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

Such that,

$$Z_1 = T(Z_0), \quad Z_0 \sim \pi_0, \quad Z_1 \sim \pi_1$$



The Landscape of One-Step Models (And Their Flaws)

- **Generative Adversarial Networks (GANs):**
 - **Mechanism:** Parameterize the transport map T with a neural network and train via minimax optimization.
 - **Bottlenecks:**
 - Notorious numerical instability.
 - Severe mode collapse.
 - Require heavy domain-specific engineering.



The Landscape of One-Step Models (And Their Flaws)

- **Maximum Likelihood Estimation (MLE) / VAEs / Normalizing Flows:**
 - **Mechanism:** Optimize explicit or variational approximations of likelihood.
 - **Bottlenecks:**
 - Intractable likelihoods for complex models.
 - Normalizing flows demand strictly invertible architectures and computationally expensive Jacobian matrix calculations.



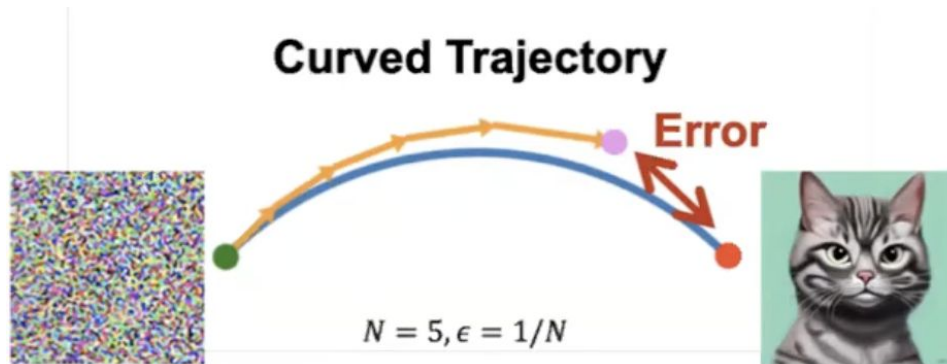
The Promise and Bottleneck of Continuous-Time Models

- **The Paradigm Shift:** Represent the transport plan implicitly as a continuous-time process using Stochastic Differential Equations (SDEs).
- **The Advantages:** Stable optimization without minimax, no strict architectural constraints, surpass GANs in both image quality and diversity.
- **The Critical Bottleneck:** High computational cost at inference.
 - Drawing a single point requires solving the SDE/ODE using a numerical solver (e.g., Euler, Runge-Kutta).
 - Requires repeatedly evaluating an expensive neural network drift function for hundreds or thousands of discretization steps.



Motivation I: Rethinking the Trajectory

- **The SDE Baggage:** Existing ODE models (like VP-ODE, sub-VP ODE) are mathematically derived from SDEs.
- **Sub-optimal Geometries:**
 - **Curved Paths:** Derivation forces the transport trajectories to be curved.
 - **Non-Uniform Speed:** They move exceptionally slowly in the early phase and aggressively speed up later.
- **The Discretization Problem:** Massive truncation errors in numerical solvers unless the step size is tiny, preventing coarse, fast inference.

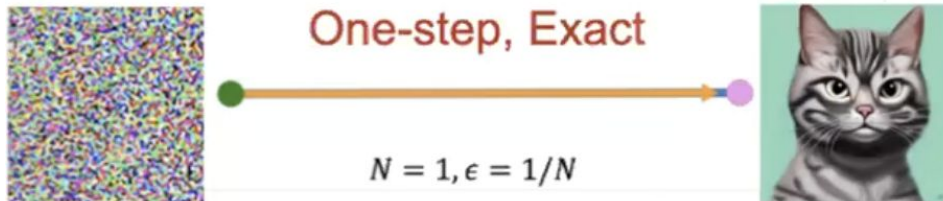


Motivation II: The Rectified Flow Solution

- **The Core Hypothesis:** What if we explicitly train an ODE to transport π_0 to π_1 by following straight-line paths from the beginning?
- **Why Straight Paths?**
 - **Theoretical Optimality:** Shortest, most efficient path between two points.
 - **Computational Superiority:** Simulated without time discretization error.
- **Bridging the Gap:** A straight-path ODE bridges the gap between continuous-time models and fast one-step models.
- **Simplicity:** Solved via an unconstrained least-squares regression



Straight Trajectory

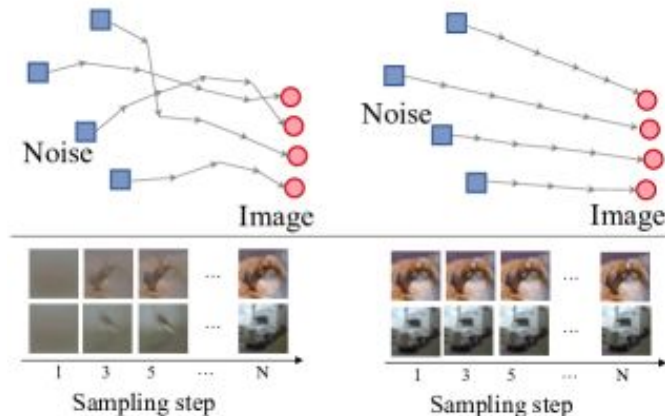


Model Comparison



Diffusion

- Forward + backward paths
- Noisy / zigzag trajectories

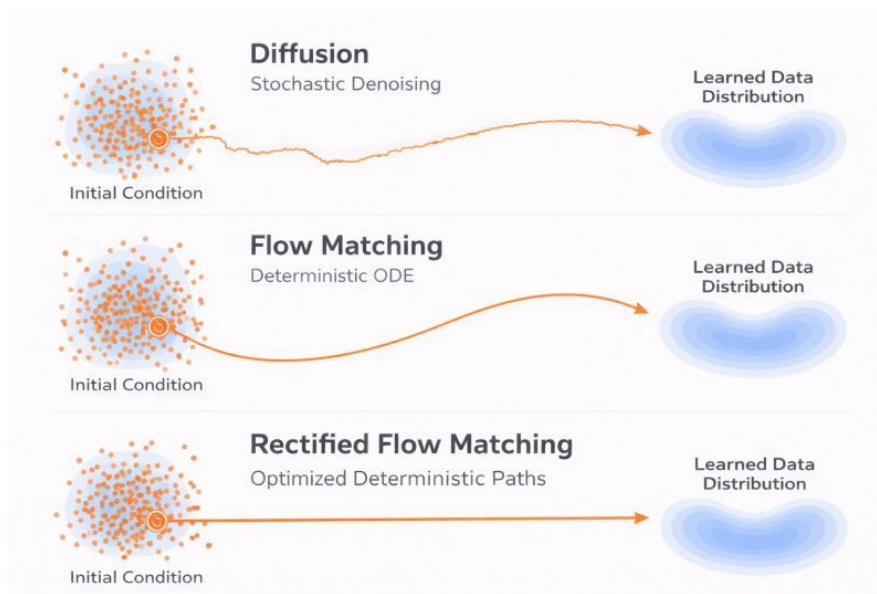


Flow Matching

- Smooth but curved paths
- Possible trajectory crossing

Rectified Flow

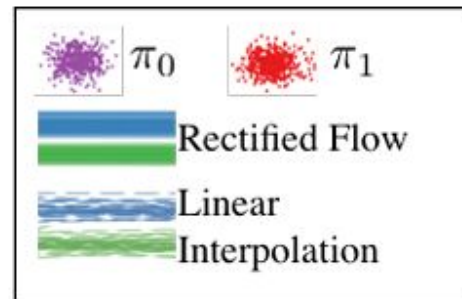
- Single-valued velocity field
- Straight, non-crossing trajectories



Iterative refinement process



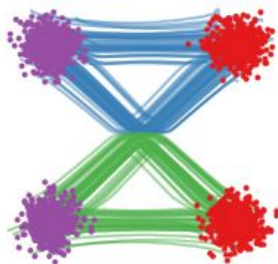
1. Training
2. Sampling
3. Reflow
4. Distillation



Training

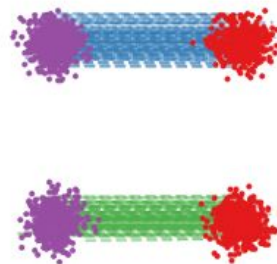
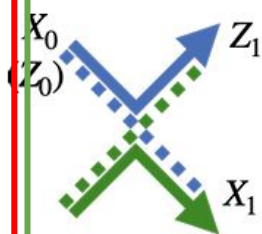


(a) Linear interpolation



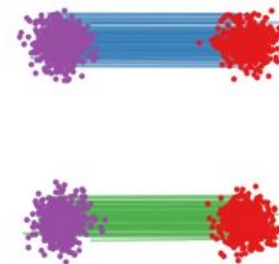
(b) Rectified flow Z_t

Sampling



(c) Linear interpolation

Reflow



(d) Rectified flow Z'_t

Algorithm

Training

1. Sample pair $x_0 \sim p_0, \quad x_1 \sim p_{\text{data}}$
2. Sample time $t \sim U[0, 1]$
3. Interpolate $x_t = (1 - t)x_0 + tx_1$
4. Target Velocity $u_t = x_1 - x_0$
5. Train Velocity field $\mathcal{L} = \mathbb{E}[\|v_\theta(x_t, t) - u_t\|^2]$

Sampling

1. Start from noise: $x(0) \sim p_0$
2. Solve ODE: $\frac{dx}{dt} = v_\theta(x, t)$
3. Obtain: $x(1) \sim p_{\text{data}}$



Algorithm 1 Rectified Flow: Main Algorithm

Procedure: $Z = \text{RectFlow}((X_0, X_1))$:

Inputs: Draws from a coupling (X_0, X_1) of π_0 and π_1 ; velocity model $v_\theta: \mathbb{R}^d \rightarrow \mathbb{R}^d$ with parameter θ .

Training: $\hat{\theta} = \arg \min_{\theta} \mathbb{E}[\|X_1 - X_0 - v(tX_1 + (1-t)X_0, t)\|^2]$, with $t \sim \text{Uniform}([0, 1])$.

Sampling: Draw (Z_0, Z_1) following $dZ_t = v_{\hat{\theta}}(Z_t, t)dt$ starting from $Z_0 \sim \pi_0$ (or backwardly $Z_1 \sim \pi_1$).

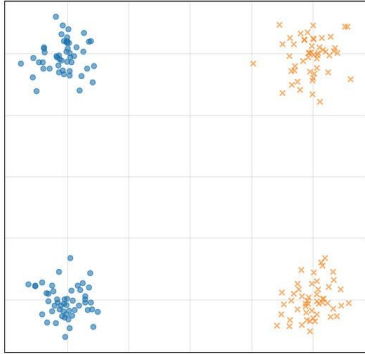
Return: $Z = \{Z_t: t \in [0, 1]\}$.

Reflow (optional): $Z^{k+1} = \text{RectFlow}((Z_0^k, Z_1^k))$, starting from $(Z_0^0, Z_1^0) = (X_0, X_1)$.

Distill (optional): Learn a neural network \hat{T} to distill the k -rectified flow, such that $Z_1^k \approx \hat{T}(Z_0^k)$.

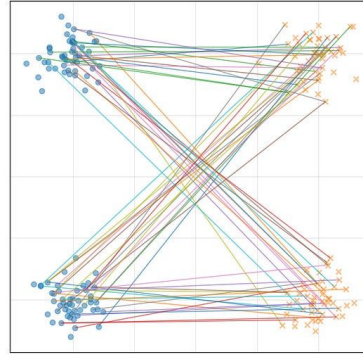
Training

Straight-line ordinary differential equation (ODE):



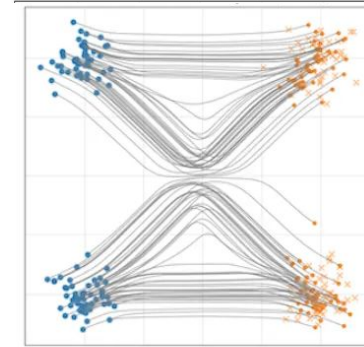
Distributions

- Blue = $x_0 \sim p_{data}$
- Orange = $x_1 \sim p_{noise}$



Lines = trajectories

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



Linear interpolation

(training targets)

$$v_\theta(x_t, t) = \mathbb{E}[x_1 - x_0 \mid x_t, t]$$



Non-Crossing Flow Property

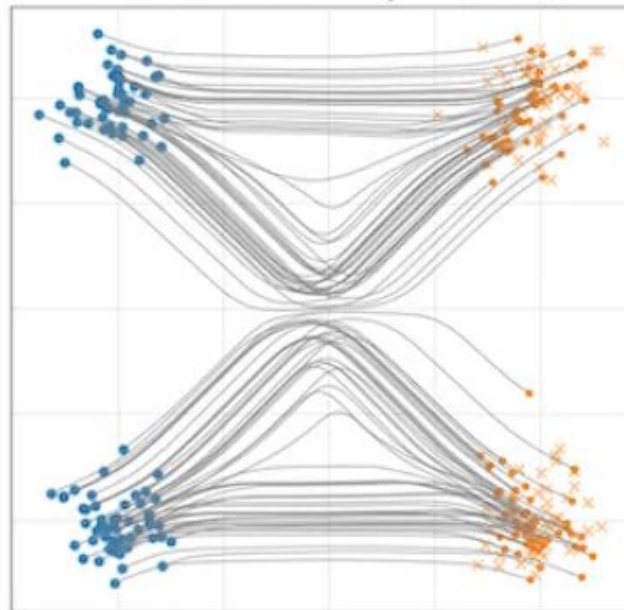
$$v_{\theta}(x, t) = \mathbb{E}[x_1 - x_0 \mid x_t = x]$$

- ODE flows have **unique solutions**
- Trajectories **cannot cross**
- Straight paths from pairs **do cross**

Rectified Flow eliminates crossings by converting inconsistent pairwise velocities into a single-valued velocity field.

- Conflicting velocities are **averaged**
- Each point has **one velocity**
- Valid flow field
- **No crossing trajectories**

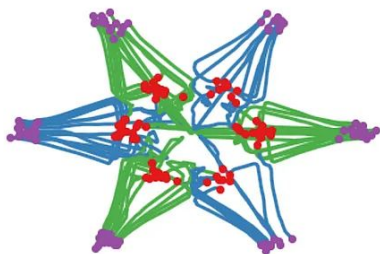
Learned velocity field (Rectified flow)



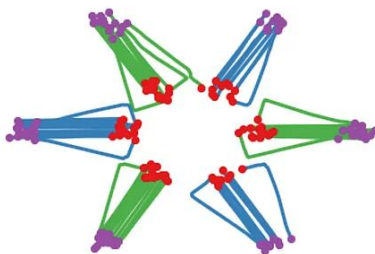
Reflow: Reduced Transport Cost

Process of Reflow:

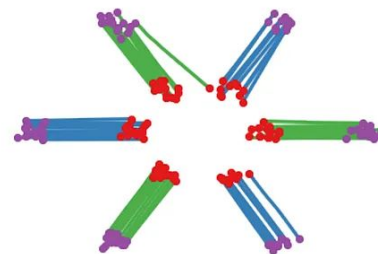
1. **Stage One Training:** Train the model on the original dataset until convergence.
2. **Generate New Pairs:** Use the stage one model with an ODE sampler to collect new (x_0, x_1) pairs.
3. **Reflow Training:** Retrain the stage one model using only the newly collected data pairs.
4. **Iterative Refinement (Optional):** Repeat the sampling and retraining steps to further straighten the velocity field and reduce transport cost.



(a) The 1st rectified flow Z^1
 $Z^1 = \text{RectFlow}((X_0, X_1))$



(b) Reflow Z^2
 $Z^2 = \text{RectFlow}((Z_0^1, Z_1^1))$



(c) Reflow Z^3
 $Z^3 = \text{RectFlow}((Z_0^2, Z_1^2))$

ReFlow Convergence

Straightness Error

$$S(Z) = \int_0^1 \mathbb{E} \left[\|(Z_1 - Z_0) - \dot{Z}_t\|^2 \right] dt$$

- Measures deviation from constant velocity (straight line)
- $S(Z) = 0 \rightarrow$ perfectly straight flow



Theoretical convergence bound

$$\begin{aligned} \sum_{k=0}^K S(Z^{k+1}) + V((Z_0^k, Z_1^k)) &\leq \mathbb{E}[\|X_1 - X_0\|^2] \\ \sum_{k=0}^K S(Z^{k+1}) &\leq \mathbb{E}[\|X_1 - X_0\|^2] \\ \min_{k \leq K} S(Z^k) &\leq \frac{\mathbb{E}[\|X_1 - X_0\|^2]}{K} \end{aligned}$$

- Straightness improves as K increases
- Converges at rate $O(1/K)$, K = number of ReFlow iterations

Distillation

Goal:

- Produces a faster model
- Generation with fewer steps / 1-step model



Many small steps (teacher ODE)

$$x_1 \rightarrow x_{0.9} \rightarrow \dots \rightarrow x_0$$

One direct mapping

$$x_0 = f_\phi(x_1)$$

Straight trajectories from rectification make the teacher mapping easier to approximate, enabling high-quality one-step distillation.

Marginal Preserving

Rectified flow preserves the marginal distributions at every time t

Marginal law of Z_t : $\text{Law}(Z_t) = p_t^{(Z)}(x)$



What it means

$$\text{Law}(Z_t) = \text{Law}(X_t), \quad \forall t$$

- Even though trajectories are changed
- The distribution at each time stays the same

$$\text{Flow in \& out} \left(\begin{array}{c} \text{Blue arrows} \\ \text{Green arrows} \\ \text{Red dots} \end{array} \right) = \text{Flow in \& out} \left(\begin{array}{c} \text{Green arrows} \\ \text{Blue arrows} \\ \text{Red dots} \end{array} \right)$$

Nonlinear Extension

Linear path

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

Nonlinear path

$$x_t = \phi_t(x_0, x_1)$$

Training loss

$$\min_v \int_0^1 \mathbb{E} \left[w_t \left\| v(X_t, t) - \dot{X}_t \right\|^2 \right] dt$$

✗ Problem 1: No transport cost reduction

- Nonlinear case \rightarrow may not be shortest path

✗ Problem 2: ReFlow stops working

- paths are already arbitrary
- no guarantee they become straighter



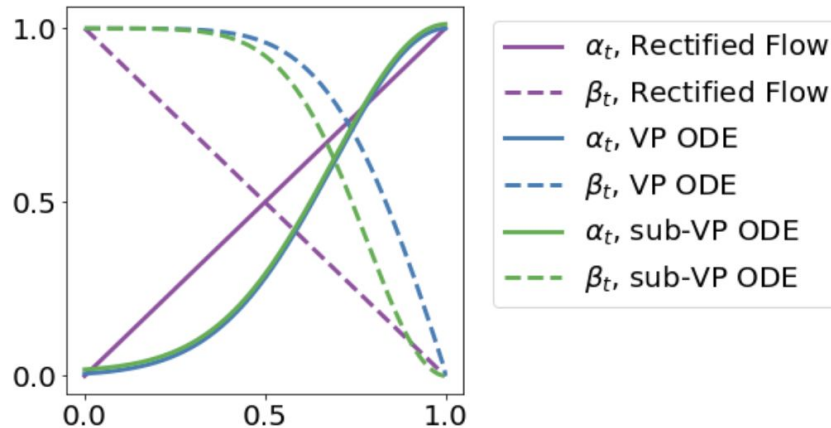
Comparison - Probability Flow ODEs

- **The Origin:** Diffusion models (DDPM) use noisy Stochastic Differential Equations (SDEs).
- **The Conversion:** Fokker-Planck equations convert noisy SDEs into deterministic ODEs.
- **PF-ODEs:** Same marginal distributions as SDEs, but with zero injected noise.
- **Key Baselines:**
 - **VP ODE:** Continuous-time DDIM.
 - **sub-VP ODE:** Stricter variance bound.



Comparison - Unifying the Models

- **The Nonlinear Framework:** PF-ODEs, DDIM, and Rectified Flow all share the same interpolation curve: $X_t = \alpha_t X_1 + \beta_t \xi$ $\xi \sim \mathcal{N}(0, I)$
- **The SDE Baggage:** VP and sub-VP ODEs rely on complex schedules inherited from SDEs.
 - *Exponential Decay:* $\alpha_t = \exp(-\frac{1}{4}a(1-t)^2 - \frac{1}{2}b(1-t))$
 - *Mismatched Variance:* VP sets $\beta_t = \sqrt{1 - \alpha_t^2}$, sub-VP sets $\beta_t = 1 - \alpha_t^2$



Comparison - Two Fatal Flaws for Fast Simulation

- Flaw 1: Inherently Curved Paths:** Because $\beta_t \neq 1 - \alpha_t$, trajectories are forced to curve.
- Flaw 2: Non-Uniform Speed:** The exponential α_t schedule causes sluggish early movement ($t \leq 0.5$).



Time-Discretization Steps N	Rectified Flow $\alpha_t = t, \beta_t = 1 - t$	VP ODE α_t in (7), $\beta_t = \sqrt{1 - \alpha_t^2}$	sub-VP ODE α_t in (7), $\beta_t = 1 - \alpha_t^2$
$N = 1$			
$N = 2$			
$N = 5$			
$N = 100$			

Comparison - The Rectified Flow Solution

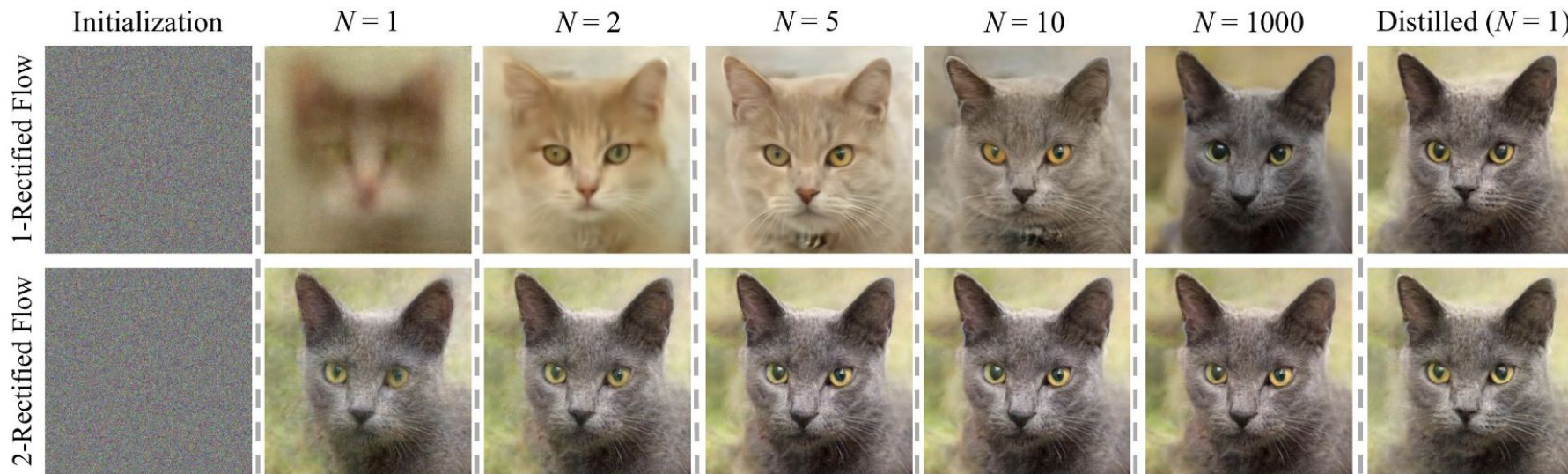
- **The Linear Fix:** Drop the SDE derivation entirely. Set $\alpha_t = t$ and $\beta_t = 1 - t$.
- **The Result:** Perfectly straight paths.
 - Uniform, constant progression in time.
 - Stable generation with a single Euler step.



Time-Discretization Steps N	Rectified Flow $\alpha_t = t, \beta_t = 1 - t$	VP ODE α_t in (7), $\beta_t = \sqrt{1 - \alpha_t^2}$	sub-VP ODE α_t in (7), $\beta_t = 1 - \alpha_t^2$
$N = 1$			
$N = 2$			
$N = 5$			
$N = 100$			

Experiments - Unconditioned Image Generation

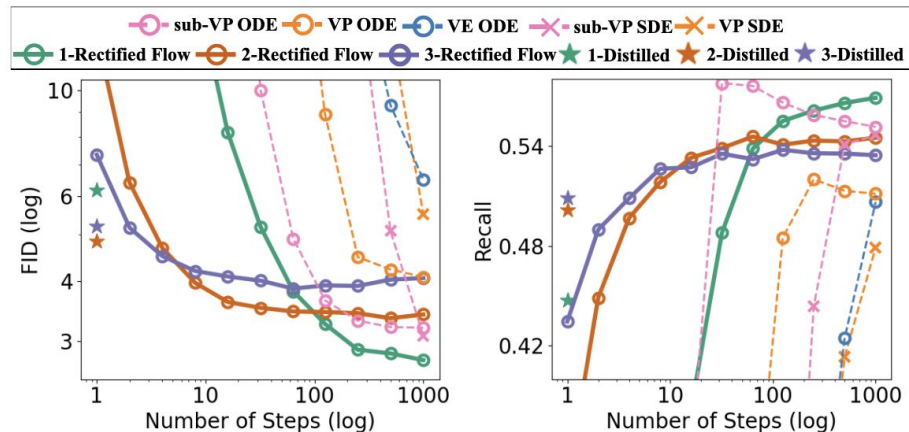
- **Task:** Generating images from standard Gaussian noise ($\pi_0 \rightarrow \pi_1$).
- **Setup:** Evaluated on CIFAR-10.



Experiments - Unconditioned Image Generation

Method	NFE(↓)	IS (↑)	FID (↓)	Recall (↑)
<i>ODE</i>				
<i>One-Step Generation (Euler solver, N=1)</i>				
1-Rectified Flow (+Distill)	1	1.13 (9.08)	378 (6.18)	0.0 (0.45)
2-Rectified Flow (+Distill)	1	8.08 (9.01)	12.21 (4.85)	0.34 (0.50)
3-Rectified Flow (+Distill)	1	8.47 (8.79)	8.15 (5.21)	0.41 (0.51)
VP ODE [73] (+Distill)	1	1.20 (8.73)	451 (16.23)	0.0 (0.29)
sub-VP ODE [73] (+Distill)	1	1.21 (8.80)	451 (14.32)	0.0 (0.35)
<i>ODE</i>				
<i>Full Simulation (Runge-Kutta (RK45), Adaptive N)</i>				
1-Rectified Flow	127	9.60	2.58	0.57
2-Rectified Flow	110	9.24	3.36	0.54
3-Rectified Flow	104	9.01	3.96	0.53
VP ODE [73]	140	9.37	3.93	0.51
sub-VP ODE [73]	146	9.46	3.16	0.55
<i>SDE</i>				
<i>Full Simulation (Euler solver, N=2000)</i>				
VP SDE [73]	2000	9.58	2.55	0.58
sub-VP SDE [73]	2000	9.56	2.61	0.58

(a) Results using the DDPM++ architecture.



(a) FID and Recall vs. Number of Euler discretization steps N

Experiments - Image-to-Image Translation

- **Task:** Unpaired style/domain transfer



(A) Cat \rightarrow Wild Animals



(B) Wild Animals \rightarrow Cat



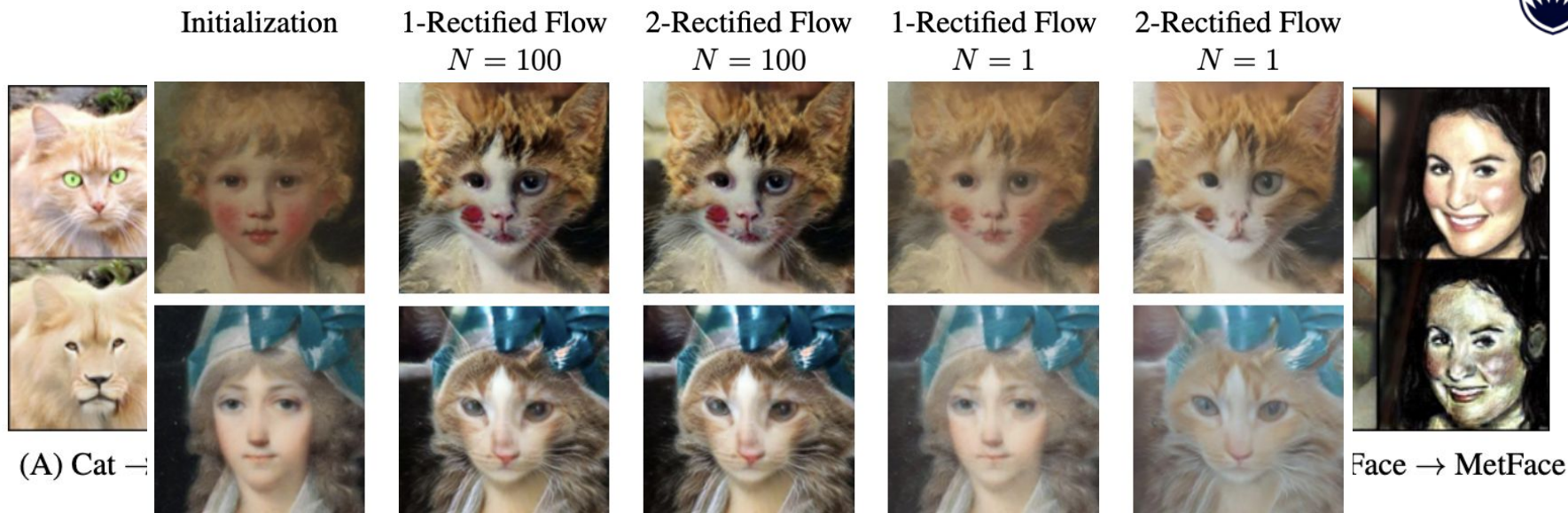
(C) MetFace \rightarrow CelebA Face



(D) CelebA Face \rightarrow MetFace

Experiments - Image-to-Image Translation

- **Task:** Unpaired style/domain transfer



Experiments - Domain Adaptation

- **The Problem:** Machine learning models degrade when test data (novel domain) differs from training data.
- **The Strategy:** Use 1-Rectified Flow to transport the novel test domain (π_0) back into the original training domain (π_1).



Method	ERM	IRM	ARM	Mixup	MLDG	CORAL	Ours
OfficeHome	66.5 \pm 0.3	64.3 \pm 2.2	64.8 \pm 0.3	68.1 \pm 0.3	66.8 \pm 0.6	68.7 \pm 0.3	69.2 \pm 0.5
DomainNet	40.9 \pm 0.1	33.9 \pm 2.8	35.5 \pm 0.2	39.2 \pm 0.1	41.2 \pm 0.1	41.5 \pm 0.2	41.4 \pm 0.1

Strengths & Weaknesses

- **Strengths:**

- Mathematical Simplicity
- Unified Framework
- Fast Inference

- **Weaknesses:**

- Training Compute Bottleneck
- Error Accumulation



Open Questions & Future Directions

- **Open Questions:**

- How does this framework scale to discrete or sequential data spaces (like text)?



- **Proposed Improvements:**

- **Dynamic Reflow Routing:** Instead of reflowing the entire dataset uniformly, selectively reflow only the trajectories with the highest curvature to drastically save on compute costs.

