Understanding Generative AI Through Optimal Transport and PDEs

ACM26 Southeast Applied & Computational Math Student Workshop

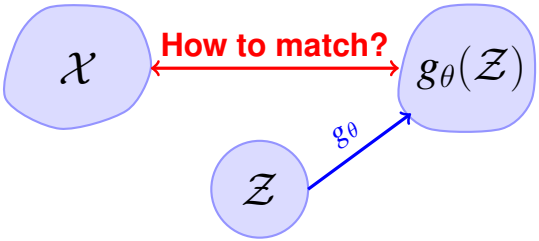
Lars Ruthotto

Departments of Mathematics and Computer Science
Emory University

lruthotto@emory.edu
larsruthotto

Generative AI Builds on Centuries of Mathematics

Generative Modeling as Distribution Matching



Mathematical Framework

- **Goal**: Learn generator $g_\theta : \mathbb{R}^q \to \mathbb{R}^n$ that transforms latent $\mathcal{Z}$ to match data $\mathcal{X}$
- **Challenges**:
  - n typically large (high-dimensional)
  - $\mathcal{X}$ complicated (multimodal, disjoint support)
- **Core Problem**: Match distributions

$$p_{g_\theta(\mathcal{Z})}(x) \approx p_{\mathcal{X}}(x)$$

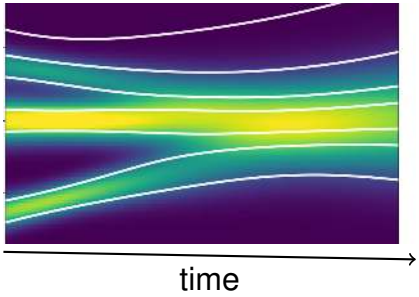
- **Today's focus**: Distribution Matching with PDEs

generative modeling = matching high-dimensional distributions

The Master Equation for Generative Modeling

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0, \quad \rho_0 = p_{\mathcal{X}}, \quad \rho_1 = p_{\mathcal{Z}}$$

Given velocity v_t, determines the density path ρ_t through probability space



Four Strategies to Construct Feasible (ρ_t, v_t) Pairs

1. **Continuous Normalizing Flows**: Optimize v_t so that $\rho_1 \approx p_{\mathcal{Z}}$
2. **Optimal Transport**: Regularize v_t by adding kinetic energy
3. **Flow Matching**: Construct (ρ_t, v_t) from conditional flows (superposition)
4. **Score-based Diffusion**: Fokker-Planck $\rightarrow$ log transform $\rightarrow$ another (ρ_t, v_t) pair

Joint work with: Katie Keegan, Rishi Leburu, Levon Nurbekyan

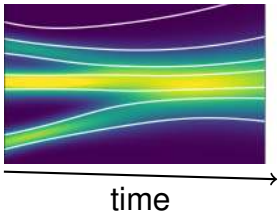
Continuous Normalizing Flows

Strategy 1: Method of Characteristics

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0, \quad t \in (0,1], \quad \rho_0 = p_{\mathcal{X}}$$

- Define **characteristic curves** (particle trajectories):

$$\frac{dx}{dt} = v_t(x,t), \quad x(0) \sim p_{\mathcal{X}}$$



- Along these curves (log density evolution):

$$\frac{d \log \rho_t(x(t))}{dt} = -\nabla \cdot v_t$$

Key Insight

- **PDE** (continuity equation) $\iff$ **System of ODEs** (particle trajectories)
- If particles follow ODEs, density automatically satisfies PDE!

CNF Idea Chen et al. 2018: Parameterize velocity field v_t as neural network $v_\theta(z,t)$

PDE transport $\Rightarrow$ Method of characteristics $\Rightarrow$ Neural ODE

CNF Training

Neural ODE

- Velocity field: $\frac{dx}{dt} = v_\theta(x(t), t)$ with $x(0) \sim p_0$
- **Advantage**: Invertible and tractable log-density for any reasonable v_θ
- Density p_t satisfies continuity equation automatically (method of characteristics)

Likelihood Computation (integrate instantaneous change of variables)

$$\log p_\theta(x) = \log p_0(x(1)) + \int_0^1 \nabla \cdot v_\theta(x(t), t) dt$$

Training

- Maximize $\mathbb{E}_{x \sim p_\mathcal{X}}[\log p_\theta(x)]$ (maximum likelihood)
- **Requirements**: ODE solve + trace computation Grathwohl et al. 2018 at every training step

Sampling: Draw $z(1) \sim p_\mathcal{Z}$, solve ODE from $t = 1$ to $t = 0$ with $v_\theta(z, t)$

elegant theory, but ODE solving + trace at every step

CNF Limitations

Computational Bottlenecks

1. **Trace computation** Grathwohl et al. 2018: $O(n)$ per evaluation (or high-variance stochastic)
2. **ODE solving**: Many function evaluations needed for adaptive time-stepping
3. **Training time**: Significantly slower than standard architectures

CNF Problem is Under-Determined

- Only map matters, no control over **trajectory shape**
- Maximum likelihood $\neq$ minimize path energy
- Can produce **complex, curved, high-energy paths**
- More function evaluations needed in training and sampling

Scale Challenge

- Doesn't scale well to high-dimensional imaging applications
- Both trace and ODE bottlenecks compound

Using CNFs for Optimal Transport

Strategy 2: Optimal Transport

Idea: Add kinetic energy penalty to MLE loss with trade-off parameter α

$$\mathcal{J}_{\text{OT-CNF}}(\boldsymbol{\theta}) = \mathbb{E}_{x\sim p_{\mathcal{X}}}[-\log p_\theta(x)] + \frac{\alpha}{2}\mathbb{E}\left[\int_0^1 \|v_\theta(x(t),t)\|^2 dt\right]$$

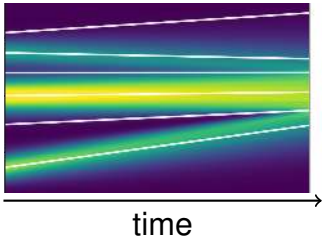
Variational Perspective (Benamou–Brenier, 2000)

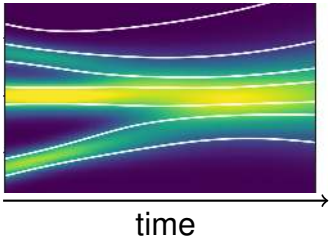
$$\begin{aligned} &\text{minimize} && \mathcal{J}(\rho_t, v_t) = \int_0^1 \int \tfrac{1}{2}\|v_t(x)\|^2 \rho_t(x)\, dx\, dt + \lambda D(\rho_1, p_{\mathcal{Z}}) \\ &\text{subject to} && \frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0, \quad \rho_0 = p_{\mathcal{X}} \end{aligned}$$

Structure from Transport Costs (when optimal)

- Optimality condition Villani 2008: $v_t = -\nabla \Phi_t$ (conservative), $\nabla \cdot v_t = -\Delta \Phi_t$
- Value function Φ_t satisfies Hamilton-Jacobi-Bellman equation
- **Key insight**: Transport cost $\rightarrow$ structure $\rightarrow$ simplified computation

OT-Flow Onken et al. 2020 – Benefits and Limitations





Benefits Over Vanilla CNF

Theoretical:

- Unique solution (OT map)
- Min kinetic energy $\rightarrow$ straighter paths

Computational:

- Explicit Laplacian: $\nabla \cdot v = -\Delta\Phi$
- More sampling efficient
- Fewer steps than vanilla CNF

Why Not Used for Imaging?

MLE Framework Limitations:

- time integration in training
- Likelihood computation: expensive and unstable (manifold hypothesis)

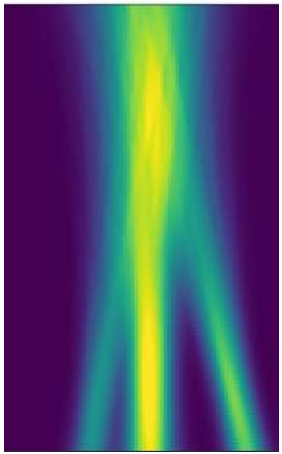
Next: Flow matching and diffusion

- no time integration in training
- simpler, faster training, SOTA sample quality

Flow Matching

Strategy 3: Feasible Paths via Superposition

$$\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0, \quad \rho_0 = p_{\mathcal{X}}, \quad \rho_1 = p_{\mathcal{Z}}$$



Special Case: Two Dirac Deltas

For point pair $p_{\mathcal{X}} = \delta(x_0)$ and $p_{\mathcal{Z}} = \delta(x_1)$, OT map is

- Conditional path: $\psi_t(x_0, x_1) = (1-t)x_0 + tx_1$
- Conditional density: $\rho_t(\cdot|x_0, x_1) = \delta(x - \psi_t(x_0, x_1))$
- Conditional velocity: $u_t(x|x_0, x_1) = \frac{d\psi_t}{dt} = x_1 - x_0$

Superposition via Linearity

Sample $x_0 \sim p_{\mathcal{X}}$ and $x_1 \sim p_{\mathcal{Z}}$ independently.
By linearity of the PDE, the marginal density is:

$$\rho_t(x) = \int \delta(x - \psi_t(x_0, x_1)) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1)\, dx_0\, dx_1 = \mathbb{E}_{x_0, x_1}[\rho_t(x|x_0, x_1)]$$

This gives a **feasible** probability path!

Question: How do we get the marginal velocity v_t?

From Conditional to Marginal Velocity

For each (x_0, x_1) the conditional density and conditional velocity satisfy

$$\frac{\partial \rho_t(x|x_0,x_1)}{\partial t} + \nabla \cdot (\rho_t(x|x_0,x_1) u_t(x|x_0,x_1)) = 0$$

Step 1: Take expectation over $(x_0, x_1) \sim p_{\mathcal{X}} \times p_{\mathcal{Z}}$

$$\int \frac{\partial \rho_t(x|x_0,x_1)}{\partial t} p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1 + \int \nabla \cdot (\rho_t(x|x_0,x_1) u_t(x|x_0,x_1)) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1 = 0$$

Step 2: Interchange differentiation and integration

$$\frac{\partial}{\partial t}\left[\int \rho_t(x|x_0,x_1) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1\right] + \nabla \cdot \left[\int \rho_t(x|x_0,x_1) u_t(x|x_0,x_1) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1\right] = 0$$

Step 3: Identify Coefficients. This is $\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0$ with:

$$v_t(x) = \frac{\int u_t(x|x_0,x_1) \rho_t(x|x_0,x_1) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1}{\int \rho_t(x|x_0,x_1) p_{\mathcal{X}}(x_0) p_{\mathcal{Z}}(x_1) dx_0 dx_1}$$

marginal velocity is weighted average of conditional velocities

Conditional Expectation Interpretation

From Integral to Conditional Expectation

The marginal velocity can be written as:

$$v_t(x) = \frac{\int u_t(x|x_0,x_1)\rho_t(x|x_0,x_1)p(x_0,x_1)dx_0dx_1}{\int \rho_t(x|x_0,x_1)p(x_0,x_1)dx_0dx_1} = \mathbb{E}[u_t(x|x_0,x_1) \mid \psi_t(x_0,x_1) = x]$$

where $p(x_0,x_1) = p_{\mathcal{X}}(x_1)\mathcal{N}(x_0)$

Problem: Computing this conditional expectation in high dimensions is **intractable**!

Reminder: Conditional Expectation

General form: $\mathbb{E}[Y \mid X = x] = \frac{\int y\, p(y|x)dy}{\int p(y|x)dy}$

Here: $Y = u_t(x|x_0,x_1)$, condition on $\psi_t(x_0,x_1) = x$

next: How to compute this expectation?

From Expectation to Function Approximation

Idea: Regression to Compute Expectation Use a neural network $v_\theta(x,t)$ and minimize:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,x_0,x_1}[\|v_\theta(\psi_t(x_0,x_1),t) - u_t(x|x_0,x_1)\|^2]$$

where $u_t(x|x_0,x_1) = x_1 - x_0$ is known analytically!

Why This Works: A Simple Example

Two data points with the same x but different y values: (x,y_1) and (x,y_2)

Minimize: $L(v) = |v(x)-y_1|^2 + |v(x)-y_2|^2$

Optimality: $\frac{\partial L}{\partial v(x)} = 2(v(x)-y_1) + 2(v(x)-y_2) = 0 \quad \Rightarrow \quad v^*(x) = \frac{y_1+y_2}{2}$

Key insight: $v^*(x)$ is the **average** of y-values at x = conditional expectation!

Conditional Flow Matching Training

Training Objective Lipman et al. 2023

$$\mathcal{J}_{\text{CFM}}(\boldsymbol{\theta}) = \mathbb{E}_{t,x_0,x_1}\left[\|v_\theta(\psi_t, t) - (x_1 - x_0)\|^2\right]$$

where the target velocity $u_t = x_1 - x_0$ is known analytically!

Training Procedure

1. Sample $x_0 \sim p_{\mathcal{X}}$, $x_1 \sim p_{\mathcal{Z}}$, $t \sim U[0,1]$
2. Compute path location: $\psi_t = (1-t)x_0 + tx_1$
3. Compute target velocity: $u_t = x_1 - x_0$
4. Compute prediction: $v_\theta(\psi_t, t)$
5. Minimize squared error with stochastic gradient descent

Advantages

- No ODE solve during training
- No trace computation
- Simple supervised learning (not adversarial or variational)

Sampling: Solve ODE with learned v_θ from $t = 1$ to $t = 0$ (same as in CNF)

supervised learning on analytically known conditional velocities

Discussion – Flow Matching

The Flow Matching Recipe

1. Construct conditional flows from point pairs (Dirac deltas)
2. Use superposition via linearity $\rightarrow$ marginal densities
3. Fit neural network to match conditional velocities (supervised learning)

Computational Advantages

- **Training**: No ODE solves, no trace estimation $\rightarrow$ significantly faster than CNF
- **Sampling**: Linear interpolation $\rightarrow$ fewer function evaluations
- **Simplicity**: Convexity in v_t, supervised learning

Trade-offs

- Produces **feasible** pairs (satisfies continuity equation) ✓
- Does NOT minimize Benamou-Brenier kinetic energy (not optimal)
- Construction beats optimization in high dimensions!

State-of-the-Art: Stable Diffusion 3, Sora, AlphaFold 3

next: stochastic alternative via Fokker-Planck

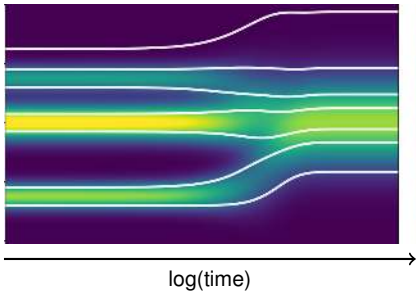
Score-Based Diffusion

Strategy 4: Fokker-Planck PDE

Idea Song et al. 2021**:** Use second-order PDE to map data to Gaussian

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot (v_t p_t) + \frac{g^2(t)}{2} \nabla \cdot (\nabla p_t), \quad t > 0, \quad p_0 = p_{\mathcal{X}}$$

Choose v_t so that $p_T(x) \to \mathcal{N}(\mathbf{0}, \mathbf{I})$ as $T \to \infty$.



Example: Variance-Preserving (Mean Reversal)

$$v_t(x) = -\frac{1}{2}\beta(t)x, \quad g^2(t) = \beta(t) \quad \Rightarrow \quad p_T \to \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Common $\beta(t)$**: Linear:** $\beta(t) = \beta_{\min} + (\beta_{\max} - \beta_{\min})t/T$ **Cosine:** $\alpha_t = \cos(\frac{\pi t}{2T})$, $\sigma_t = \sin(\frac{\pi t}{2T})$

diffusion term causes asymptotic convergence to Gaussian

Log Transform Reveals Score Function

The Score Function

$$s_t(x) := \nabla_x \log p_t(x) = \frac{\nabla p_t(x)}{p_t(x)}$$

Substituting Score into FP-PDE

Recall divergence-gradient form from previous slide. Use $\nabla p_t = p_t s_t$ to rewrite:

$$\frac{\partial p_t}{\partial t} = -\nabla \cdot \left[v_t p_t - \frac{g^2(t)}{2} \nabla p_t \right]$$

Score Matching – Conditional Construction

Idea (similar to flow matching): Start simple and use linearity of PDE!

Construction Strategy

Step 1: Pick $x_0 \sim p_{\mathcal{X}}$, solve FP-PDE with Dirac initial condition $p_0(x) = \delta(x - x_0)$

Solution is Gaussian (from variance-preserving SDE):

$$p_t(x|x_0) = \mathcal{N}(x; \alpha_t x_0, \sigma_t^2 \mathbf{I})$$

where α_t, σ_t are known analytically for the variance-preserving schedule

Step 2: Conditional score (differentiate log pdf of Gaussian)
For $x = \alpha_t x_0 + \sigma_t \varepsilon$ with $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$:

$$s_t(x|x_0) = \nabla_x \log p_t(x|x_0) = -\frac{x - \alpha_t x_0}{\sigma_t^2} = -\frac{\varepsilon}{\sigma_t}$$

Next Step: Marginalize

- Similar to flow matching: $p_t(x) = \int p_t(x|x_0) p_{\mathcal{X}}(x_0)\, dx_0 = \mathbb{E}_{x_0}[p_t(x|x_0)]$
- New problem: Computing score is more difficult because log is nonlinear!

next: how to get marginal score from conditional scores

From Conditional to Marginal Score

Marginal density from superposition: $p_t(x) = \int p_t(x|x_0) p_{\mathcal{X}}(x_0)\, dx_0$

Computing the Score (note: cannot simply average!)

Chain rule for positive functions a_i: $\nabla \log \sum_i a_i = \frac{\sum_i \nabla a_i}{\sum_i a_i}$
Apply to marginal:

$$s_t(x) = \nabla_x \log p_t(x) = \frac{\nabla_x \int p_t(x|x_0) p_{\mathcal{X}}(x_0)\, dx_0}{\int p_t(x|x_0) p_{\mathcal{X}}(x_0)\, dx_0}$$

Score-Based Diffusion Training

Training Objective

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{t,x_0,\varepsilon}[\|s_\theta(x_t,t) - s_t(x_t|x_0)\|^2]$$

where $s_t(x_t|x_0) = -\frac{\varepsilon}{\sigma_t}$ (known analytically!)

Training Procedure

1. Sample $x_0 \sim p_{\mathcal{X}}$ (data), $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \sim U[0,T]$
2. Compute forward diffusion: $x_t = \alpha_t x_0 + \sigma_t \varepsilon$
3. Compute target score: $s_t(x_t|x_0) = -\frac{\varepsilon}{\sigma_t}$
4. Compute prediction: $s_\theta(x_t,t)$
5. Minimize squared error with gradient descent

Advantages

- No ODE solve during training
- No trace computation
- Trivial forward process (just add noise)

supervised learning on analytically known conditional scores

Time Reversal and Sampling

Reverse SDE (stochastic, white trajectories)

$$dx = [f(x,t) - g^2(t)s_\theta(x,t)]\,dt + g(t)\,d\bar{W}$$

Time-reversed SDE: samples from $p_0 = p_{\mathcal{X}}$ starting from $p_T \approx \mathcal{N}(\mathbf{0}, \mathbf{I})$.

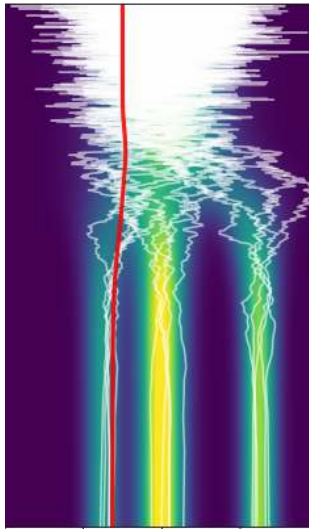
Probability Flow ODE (deterministic, red trajectory)

$$\frac{dx}{dt} = f(x,t) - \frac{g^2(t)}{2} s_\theta(x,t)$$

Same marginals as SDE, but deterministic.

Advantages of ODE

- Faster sampling (adaptive step sizes)
- Exact likelihood computation
- Latent space interpolation



A Family of Samplers (Adding a Smart Zero)

Starting point: the continuity equation from the log transform Song et al. 2021

Add and subtract $\frac{\widetilde{g}^2}{2} s_t$ inside the bracket for any $\widetilde{g}(t)$:

$$\partial_t p = -\nabla \cdot \left[\left(v_t - \tfrac{g^2}{2} s_t \right) p \right]$$

same density, your choice of noise — just add a smart zero

Training and Sampling Characteristics

Training Advantages

- No ODE solving ✓
- No trace computation ✓
- Forward diffusion is trivial (just add noise)
- **Extremely stable** – regression on Gaussian noise

Sampling

- More steps than flow matching (100–1000 vs 20–100 NFE)
- But very high sample quality (SOTA FID scores)
- Extremely robust across different architectures

Trade-off

- Slower sampling, but simpler training
- Stochastic spreading vs deterministic transport
- Entropy-regularized OT interpretation (Schrödinger bridge)

State-of-the-Art: DALL-E 2, Imagen, Stable Diffusion (1-2)

extreme training stability, SOTA quality, more sampling steps

Diffusion on Manifolds

When Data Lives on a Manifold

Setup: data satisfies constraints $h(x) = 0$, $h\colon \mathbb{R}^d \to \mathbb{R}^{d-m}$
$\Rightarrow p_0$ supported on m-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$ with $m < d$

Problem: p_0 is singular w.r.t. Lebesgue measure in $\mathbb{R}^d$

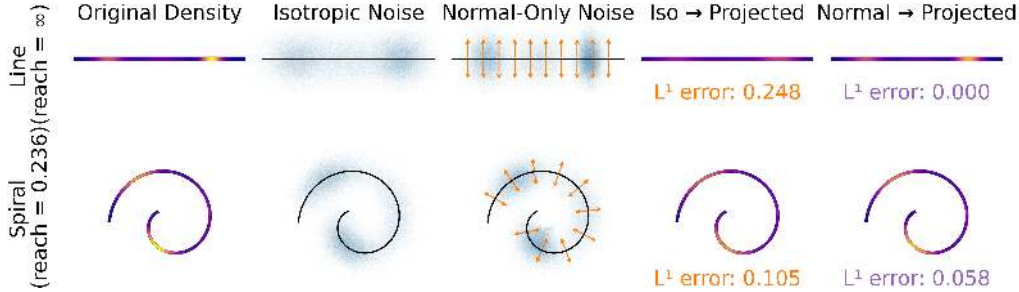
- Score explosion: $\|\nabla_x \log p_t(x)\| \to \infty$ as $t \to 0$ near $\mathcal{M}$
- NF breakdown: Jacobian determinant $= 0$ (no invertible map $p_0 \leftrightarrow p_T$)
- Standard generative models assume full-dimensional support

Examples:

- Protein backbone geometry (bond length/angle constraints)
- Images with fixed total intensity
- Physics simulations with conservation laws

standard diffusion fails when data has lower-dimensional support

Manifold-Aware Noising: (Keegan and Ruthotto 2026)



Key idea: perturb data only in normal directions to $\mathcal{M}$
Lifted distribution: $n \,|\, z \sim \mathcal{N}\big(0,\ \sigma^2 P_{N_z\mathcal{M}}\big), \quad x = z + n$

Algorithm:

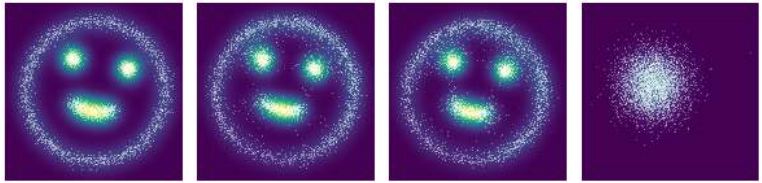
1. Lift: perturb into normal bundle $\rightarrow$ samples from p_σ
2. Train: any generative model (DDPM, NF, FM) on p_σ
3. Project: $\Pi(x) = \arg\min_{z\in\mathcal{M}} \|x - z\|$

Theory: $d_{\mathrm{TV}}(\Pi_{\#}p_\sigma, p_0) \le C_1 \exp(-C_2 r^2/\sigma^2)$ (general $\mathcal{M}$, $r < \mathrm{reach}(\mathcal{M})$)

extrinsic, model-agnostic: only 2 manifold-aware ops (lift + project)

Numerical Results

Smiley on plane ($\mathcal{M}$ = hyperplane in $\mathbb{R}^3$, linear constraint)



Data p_0 | $\Pi_{\#}p_\sigma$ (ours) | DDPM + proj. | PIDM

Gaussian on Stanford bunny (complex mesh, non-uniform curvature)

Data p_0 | $\Pi_{\#}p_\sigma$ (ours) | DDPM | PIDM

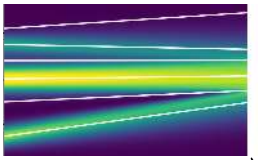
p_σ achieves best fidelity with guaranteed constraint satisfaction

Summary

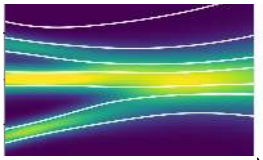
The Master Equation Unifies Everything

The Continuity Equation $\frac{\partial \rho_t}{\partial t} + \nabla \cdot (\rho_t v_t) = 0, \quad \rho_0 = p_{\mathcal{X}}, \quad \rho_1 = p_{\mathcal{Z}}$

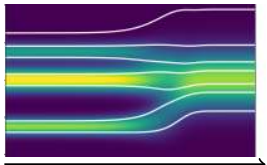
Optimal Transport



Flow Matching



Diffusion



Method	Strategy	Optimal?	Train	Sample	Key Differentiator
CNF Chen et al. 2018	Direct v_θ	No	Slow	Slow	MLE with trace bottleneck
OT Flow Onken et al. 2020	Min. energy	Yes	Medium	**Fast**	Theory: $v = -\nabla\Phi$
Flow Match Lipman et al. 2023	Superposition	No	Fast	Slow	no time integration, arbitrary $p_{\mathcal{Z}}$
Diffusion Song et al. 2021	FP $\rightarrow$ continuity	No	Fast	Slow	no time integration, SDE sampling

all roads lead back to the continuity equation

Outlook: Generative AI Through OT and PDEs

Open Research Directions

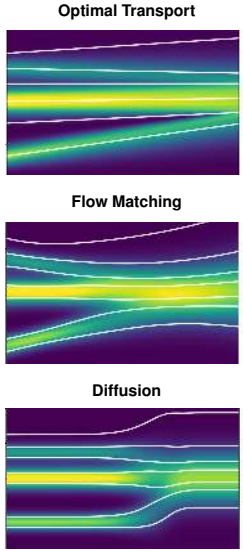
- The holy grail: fast + optimal + simple
- Beyond continuity: telegrapher, Poisson, ...
- Connections to optimal control, MFG, Wasserstein flows

Related Talks

- **P. Wang** (Auburn) — diffusion error estimates
- **C. Fikes** (Emory) — score-based ptychography
- **H. Yan** (Georgia Tech) — denoising & manifolds
- **X. Wang** (Emory) — MFG & equilibrium pricing

Want to Learn More?

Recordings from our 2025 Emory Winter School on Computational Math & AI — ten lecture series on optimization/generalization, inverse problems, scientific ML, and AI for algorithmic discovery



Joint work with K. Keegan, R. Leburu, L. Nurbekyan (Emory)

Generative AI Builds on Centuries of Mathematics

References

- Chen, R. T. Q. et al. (2018). “Neural Ordinary Differential Equations”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31.
- Grathwohl, Will et al. (2018). “Ffjord: Free-form continuous dynamics for scalable reversible generative models”. In: *arXiv preprint arXiv:1810.01367*.
- Keegan, Katherine and Lars Ruthotto (2026). “Manifold-Aware Perturbations for Constrained Generative Modeling”. In: arXiv: 2601.23151 [cs.LG].
- Lipman, Y. et al. (2023). “Flow Matching for Generative Modeling”. In: *International Conference on Learning Representations (ICLR)*.
- Onken, Derek et al. (May 2020). “OT-Flow: Fast and Accurate Continuous Normalizing Flows via Optimal Transport”. In: *arXiv.org*. arXiv: 2006.00104v1 [cs.LG].
- Song, Y. et al. (2021). “Score-Based Generative Modeling through Stochastic Differential Equations”. In: *International Conference on Learning Representations (ICLR)*.
- Villani, Cédric (2008). *Optimal Transport: Old and New*. Vol. 338. Springer Science & Business Media.