Schrödinger Bridge Flow for Unpaired Data Translation

Faissal Izermine Dhia Garbaya

Deep Learning ▪ Master MVA

The Problem: Unpaired Data Translation

Transport unpaired distributions π_0 (Source) to π_1 (Target) via optimal paths without paired training data.

Challenges:

- Standard OT requires many training iterations
- Computational cost scales poorly with dimension
- Need for bidirectional consistency

Entropic Kantorovich Formulation

We use **Entropic Optimal Transport (EOT)** to find the coupling Π^*:

$$\Pi^* = \arg\min_{\Pi} \left\{ \int \frac{1}{2} \|x - y\|^2 d\Pi(x, y) - \epsilon H(\Pi) \right\}$$

The dynamic equivalent is the **Schrödinger Bridge (SB)**:

$$\mathbb{P}^* = \arg\min_{\mathbb{P}} \{KL(\mathbb{P} \| \mathbb{Q})\}$$

subject to marginal constraints $\mathbb{P}_0 = \pi_0, \mathbb{P}_1 = \pi_1$, where $\mathbb{Q}$ is the $(\sqrt{\epsilon} B_t)$.

Parameter ϵ regulates the trade-off between shortest path and diffusion smoothness.

Theoretical Projections

The SB solution alternates between two projection spaces:

Reciprocal Projection (proj$_{\mathcal{R}}$): Ensures path belongs to reciprocal class $\mathbb{P} = \mathbb{P}_{0,1}\mathbb{Q}_{|0,1}$. This fixes the bridge structure:

$$\hat{X}_t = \text{Interp}_t(X_0, X_1, Z) = (1-t)X_0 + tX_1 + \sqrt{\epsilon t(1-t)} Z$$

where $Z \sim \mathcal{N}(0, I)$.

Markovian Projection (proj$_{\mathcal{M}}$): Finds drift field $v_t^*(x_t)$ preserving marginals:

$$v_t^*(x_t) = \frac{\mathbb{E}_{\mathbb{P}}[X_1 | X_t = x_t] - x_t}{1 - t}$$

The process follows: $dX_t^* = v_t^*(X_t^*)dt + \sqrt{\epsilon} dB_t$.

We thus define the empirical training objectives:

$$\ell^{\rightarrow}(\theta; t, \mathbf{X}_1, \hat{\mathbf{X}}_t) = \frac{1}{B} \sum_{i=1}^{B} \left\| v_\theta^{\rightarrow}(t^i, \hat{\mathbf{X}}_t^i) - (\mathbf{X}_1^i - \hat{\mathbf{X}}_t^i)/(1 - t^i) \right\|^2$$

$$\ell^{\leftarrow}(\theta; t, \mathbf{X}_0, \hat{\mathbf{X}}_t) = \frac{1}{B} \sum_{i=1}^{B} \left\| v_\theta^{\leftarrow}(1 - t^i, \hat{\mathbf{X}}_t^i) - (\mathbf{X}_0^i - \hat{\mathbf{X}}_t^i)/t^i \right\|^2$$

The α-DSBM Algorithm

Key Innovation: α-DSBM introduces a flow with step size $\alpha \in (0, 1]$:

$$\hat{\mathbb{P}}^{n+1} = (1 - \alpha)\hat{\mathbb{P}}^n + \alpha \text{proj}_{\mathcal{R}}(\text{proj}_{\mathcal{M}}(\hat{\mathbb{P}}^n))$$

Two-Phase Training Strategy:

1. **Pretraining Phase:** Optimize bridge matching on raw samples $(X_0, X_1) \sim \pi_0 \otimes \pi_1$:
$$\mathcal{L}(\theta) = \frac{1}{2} \left[\ell^{\rightarrow}(t^{1:b}, \mathbf{X}_1^{1:b}, \hat{\mathbf{X}}_t^{1:b}) + \ell^{\leftarrow}(t^{b+1:B}, \mathbf{X}_0^{b+1:B}, \hat{\mathbf{X}}_t^{b+1:B}) \right]$$
2. **Online Finetuning:** Incremental updates on θ and EMA parameters θ^{EMA}:
   - **SDE Sampling:** Generate endpoints $\hat{\mathbf{X}}_1$ and $\hat{\mathbf{X}}_0$ using $v_{\theta^{\text{EMA}}}$:
   $$dX_t = v_{\theta^{\text{EMA}}}^{\rightarrow}(t, X_t)dt + \sqrt{\epsilon} dB_t, \quad X_0 \sim \pi_0 \implies \hat{\mathbf{X}}_1$$
   $$dY_t = v_{\theta^{\text{EMA}}}^{\leftarrow}(t, Y_t)dt + \sqrt{\epsilon} dB_t, \quad Y_0 \sim \pi_1 \implies \hat{\mathbf{X}}_0$$
   - **Bridge Interpolation (Batch):** Sample $t^{\rightarrow} \sim \text{Unif}([0,1])^{\otimes b}, \mathbf{Z}^{\rightarrow} \sim \mathcal{N}(0, I)^{\otimes b} \rightarrow \hat{\mathbf{X}}_t^{\rightarrow} = \text{Interp}_{t^{\rightarrow}}(\hat{\mathbf{X}}_0, \mathbf{X}_1, \mathbf{Z}^{\rightarrow})$ Sample $t^{\leftarrow} \sim \text{Unif}([0,1])^{\otimes b}, \mathbf{Z}^{\leftarrow} \sim \mathcal{N}(0, I)^{\otimes b} \rightarrow \hat{\mathbf{X}}_t^{\leftarrow} = \text{Interp}_{t^{\leftarrow}}(\mathbf{X}_0, \hat{\mathbf{X}}_1, \mathbf{Z}^{\leftarrow})$
   - **Parameter Update:** Update θ with step size α and update EMA:
   $$\theta \leftarrow \theta - \alpha \nabla_\theta \frac{1}{2} \left[\ell^{\rightarrow}(t^{\rightarrow}, \mathbf{X}_1, \hat{\mathbf{X}}_t^{\rightarrow}) + \ell^{\leftarrow}(t^{\leftarrow}, \mathbf{X}_0, \hat{\mathbf{X}}_t^{\leftarrow}) \right]$$
   $$\theta^{\text{EMA}} = \gamma \theta^{\text{EMA}} + (1 - \gamma)\theta$$

Advantage: The online gradient update avoids expensive repeated Markovian projections and ensures stability via EMA.

Bidirectional Network Trick

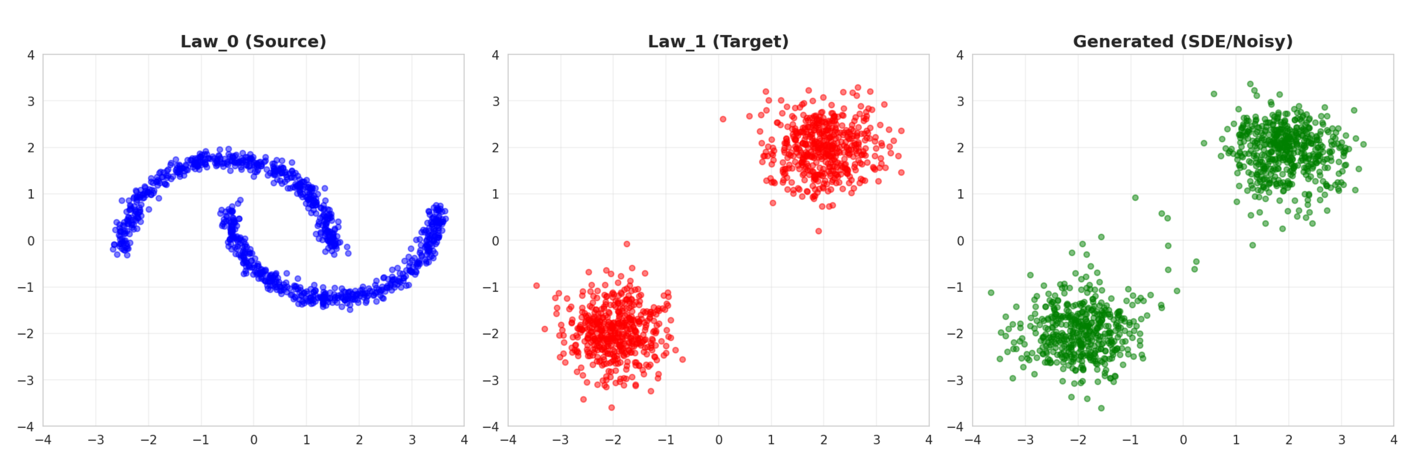
Problem: Traditional methods require two separate networks.

Solution: Single bidirectional network $v_\theta(s, t, x)$

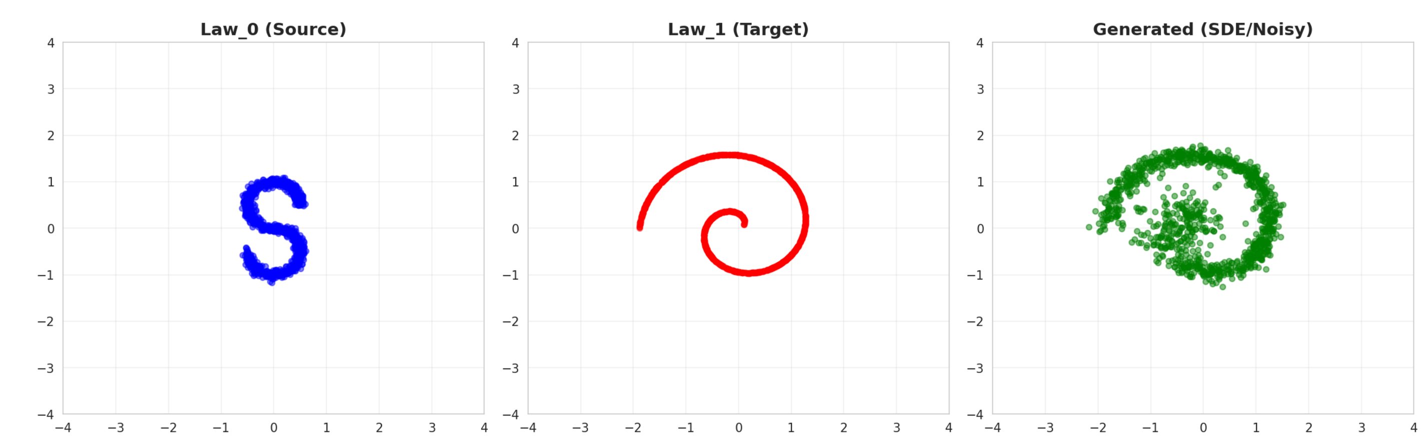
- **Direction Bit:** Binary input $s \in \{0, 1\}$
- $v_\theta^{\rightarrow}(\cdot) = v_\theta(1, \cdot)$ handles forward drift
- $v_\theta^{\leftarrow}(\cdot) = v_\theta(0, \cdot)$ handles backward drift

Experimental Results: 2D Toy Datasets

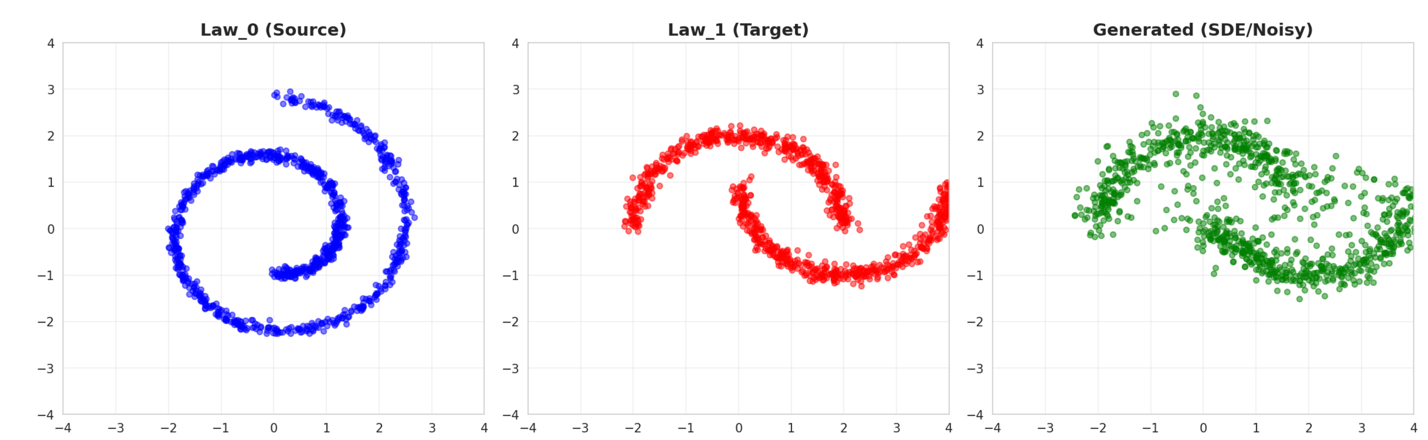
Test Scenarios: Complex topology mappings



Moons to Blobs



S-Curve to Spiral



Swiss Roll to Moons

Key Observations

- **Topology Preservation:** Successfully recovers complex geometric structures
- **Robustness:** Handles high-variance initial distributions
- **Quality:** Generated samples match target distribution characteristics

Conclusion & Future Directions

Summary: α-DSBM provides computationally efficient Schrödinger Bridge solutions without multiple full retraining iterations.

Future Work:

- Apply it to **Black-Box finetuning** of diffusion models
- Develop sampling-free training procedure

De Bortoli et al., "Schrödinger Bridge Flow for Unpaired Data Translation", September 2024.