Categorical Iterative Proportional Fitting via Discrete Diffusion Models. SMILES Summer School Project.

Supervisors: S. Kholkin[1], A. Korotin [1, 2]
Students: Ilya Sudakov, Ekaterina Grishina, Artem Basharin, Ivan Gurev, Nikita Kurdiukov
[1] Skoltech
[2] AIRI

Abstract

Transporting between arbitrary distributions is a fundamental goal in generative modeling. Recently proposed diffusion-bridge and flow-matching models offer a solution by recovering the Schrödinger Bridge (SB) or Optimal Transport (OT) formulation. These frameworks enjoy desirable properties—such as minimal-energy translations between distributions—and have been widely applied to unpaired domain-translation problems. However, they rely on continuous domains, limiting their applicability to discrete settings (e.g., text or graphs). To address this gap, we propose Categorical Iterative Proportional Fitting (C-IPF), a novel framework that leverages discrete diffusion models to solve the SB problem in high-dimensional discrete state spaces. Our approach builds on recent advances in discrete diffusion modeling to construct a sequence of models that converges to the discrete-state-space SB solution. The code is available at this repository.

Index Terms: Schrodinger Bridges, Optimal Transport, Discrete Diffusions, Continuous Time Markov Chains.

1 Introduction

The Schrödinger Bridge (SB) problem has recently attracted attention from the machine learning community due to its relevance for modern challenges in generative modeling and unpaired learning. The SB solution can translate between two arbitrary distributions, p_0 and p_1 , while minimizing the energy required for that translation. This contrasts with standard diffusion models, which always fix one distribution (the noise distribution).

The recent advantages in generative modeling on discrete state spaces include discrete diffusions, or Continuous in Time Markov Chains (CTMC) [1], which have shown great potential in text generation [2, 3], image generation [4] and graph generation [5]. Extending this success to unpaired learning problem is another challenge. Only a few theoretically justified approaches have been developed, such as diffusion models [6] and SB [7, 5] on discrete state spaces.

This project explores the potential of Iterative Proportional Fitting (IPF) [8, 9, 10] and Discrete Diffusion Models [2, 3, 1] combination for finding the solution to SB problems on discrete state spaces [7, 5]. Most of the work related to Schrodinger Bridges is concentrated on the continuous state space formulation, sacrificing the applicability of SB on the discrete state spaces. In this project we propose to construct the discrete state space analog to the IPF diffusion algorithms presented in [8, 9]. In detail, we propose to utilize the reverse discrete diffusion learning procedure from [3], build a sequence of reverse discrete diffusions that in the limit converges to the SB. We call this algorithm Categorical Iterative Proportional Fitting (**C-IPF**).

Our contributions are as follows:

- We implement the learning of discrete diffusion process and C-IPF procedure.
- We validate the performance of this algorithm on quantized Gaussian → Swiss-Roll transport (Sec 4.2 [7]) and on Colored MNIST "3" → "2" transport in the pixel space (Sec 4.3 [7]).
- The converged C-IPF procedure successfully translate between distributions p_0 and p_1 in sense of the diversity and fidelity of the resulting samples.

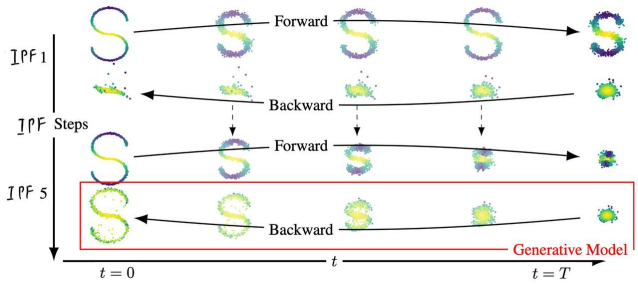


Figure 1. IPF procedure visualization. Credits to Figure 1 [8].

2 Background & Literature Review

2.1 Continuous state spaces.

Schrödinger Bridge Problem. The Schrödinger Bridge problem for a pair of distributions p_0, p_1 and volatility coefficient ϵ consists of finding a stochastic process $\mathbb{P}^{\text{SB}}$ such that:

$$\mathbb{P}^{\text{SB}} = \text{argmin}_{\mathbb{P}}\{KL(\mathbb{P}\|W^{\epsilon}), \quad \mathbb{P}_{t=0} = p_0, \mathbb{P}_{t=1} = p_1\},$$

where $\mathbb{P}$ is a diffusion process on real state space.

In other words find such a process $\mathbb{P}^{\text{SB}}$ that starts and ends at p_0 and p_1, correspondingly, while being as close as possible to Wiener process W^{ϵ}, which can be treated as *energy minimization*.

Recently, a variety of methods have been proposed to solve SB in continuous spaces; see [11] for a recent survey. Contemporary approaches include mini-batch optimal transport [12], stochastic optimal control [13], iterative procedures [14, 7, 15].

Iterative Proportional Fitting. One of the most celebrated approaches is Iterative Proportional Fitting (IPF) [10, 9, 8], which constructs a sequence of diffusion models where each n + 1-th model is trained to reverse the n-th. Specifically, the IPF procedure begins by learning a diffusion that maps noisy data p_0 to its clean counterpart, then training its reverse. Next, this reverse diffusion

is initialized at another distribution p_1 and its reverse is learned as well. The cycle continues until convergence to a diffusion that transforms p_0 into p_1. The algorithm appears in Algorithm 1, and its visualization is shown in Figure 1.

Algorithm 1: Iterative Proportional Fitting

Input: Samples from p_0: $\{p_0^{(i)}\}_{i=1}^n$, Samples from p_1: $\{p_1^{(i)}\}_{i=1}^n$
Output: SBP solution $\mathbb{P}^{\text{SB}}$

$\{x_0^{(i)}\}_{i=1}^m \leftarrow$ Draw minibatch from p_1;
$W^\epsilon(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start W^ϵ noising process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
$\mathbb{Q}^{\text{backward}} \leftarrow$ Learn a reverse for a process $W^\epsilon(\{x_0^{(i)}\}_{i=1}^m)$;
while *not converged* **do**
  $\{x_0^{(i)}\}_{i=1}^m \leftarrow$ Draw minibatch from p_0;
  $\mathbb{Q}^{\text{backward}}(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start $\mathbb{Q}^{\text{backward}}$ process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
  $\mathbb{Q}^{\text{forward}} \leftarrow$ Learn a reverse for a process $\mathbb{Q}^{\text{backward}}(\{x_0^{(i)}\}_{i=1}^m)$;
  $\{x_0^{(i)}\}_{i=1}^m \leftarrow$ Draw minibatch from p_1;
  $\mathbb{Q}^{\text{forward}}(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start $\mathbb{Q}^{\text{forward}}$ process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
  $\mathbb{Q}^{\text{backward}} \leftarrow$ Learn a reverse for a process $\mathbb{Q}^{\text{forward}}(\{x_0^{(i)}\}_{i=1}^m)$;
$\mathbb{P}^{\text{SB}} \leftarrow \mathbb{Q}^{\text{forward}}$ **return** $\hat{y}$;

2.2 Discrete state spaces.

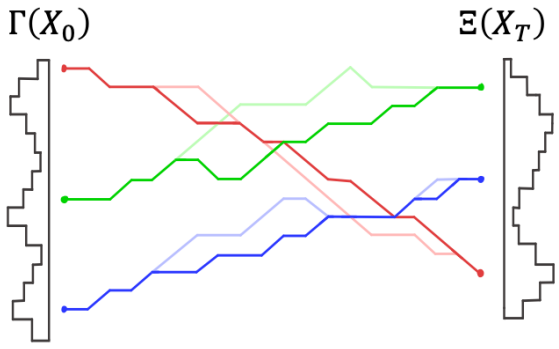
Schrödinger Bridge Problem.

The Schrödinger Bridge problem in discrete state space is defined similarly to continuous state space one. For a pair of distributions p_0, p_1 and prior "noising" process $\mathbb{Q}$ consists of finding a stochastic process $\mathbb{P}^{\text{SB}}$ such that:

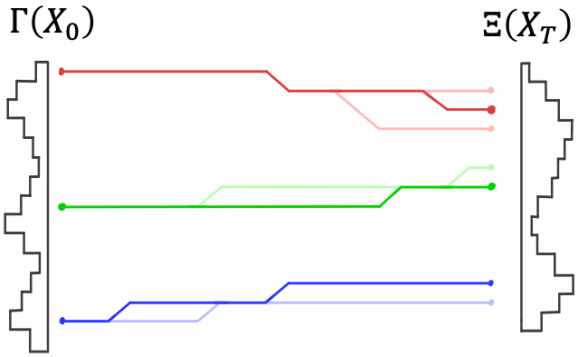
$$\mathbb{P}^{\text{SB}} = \text{argmin}_{\mathbb{P}}\{KL(\mathbb{P}\|\mathbb{Q}), \quad \mathbb{P}_{t=0} = p_0, \mathbb{P}_{t=1} = p_1\},$$

where $\mathbb{P}$ and $\mathbb{Q}$ are jump processes or CTMC (in learnable case - discrete diffusion). The main example of "noising" process $\mathbb{Q}$ is uniform categorical noising process (Eq.7 [7]). As in continuous case the solution is a process that starts and ends at p_0 and p_1 correspondingly and minimizes the $KL(\mathbb{P}\|\mathbb{Q})$, which is equivalent to minimizing the energy of translation. See the visualization at Figure 2.

The SB method in discrete state space at the moment utilizes only the IMF procedure [14] in continuous in time setting [5] and discrete in time setting [7]. Such methods are applied for unpaired domain translation on graphs and images.



(a) Discrete Diffusion



(b) Discrete Schrödinger Bridge

Figure 2. Both processes start and end at the same distribution but Schrödinger Bridge spends less energy.

2.3 Discrete diffusion models.

In recent time discrete state space generative modeling have taken a look into diffusion approach for generation. Effective methods for text generation [2, 3], image generation [4] and graph generation [5] have emerged. These approaches compared to autoregressive models have a potential to work faster since several tokens can be generated at once. Further scaling include code generation Inception Labs code generation model and Gemini Diffusion.

2.4 Discrete flow matching.

Discrete Flow Matching (DFM) [16] extends continuous-time flow ideas to high-dimensional categorical data by replacing real-valued velocity fields with token-wise jump rates in a continuous-time Markov chain. Starting from an easy-to-sample source sequence (often all [MASK] tokens), DFM learns a probability path that smoothly transports mass toward the data distribution while satisfying a discrete continuity equation. A single Transformer denoiser, trained with a simple cross-entropy loss to predict source or target token distributions at random intermediate times, yields a time-independent network whose outputs can be reused across steps. Generation proceeds by simulating the learned chain in parallel over all tokens, so sampling cost scales with the number of function evaluations (NFEs) rather than sequence length, offering substantial speed-ups over left-to-right language models. Early re-

sults are promising: a 1.7 B-parameter DFM reaches sub-10 generative perplexity on Llama-2 text and competitive HumanEval code accuracy while requiring only a few hundred NFEs; discrete-image variants achieve FID ≈ 3.6 on CIFAR-10, outperforming MaskGIT and earlier discrete flows. Current evidence positions DFM as a compelling non-autoregressive alternative for discrete modalities.

3 Methods.

Our categorical Iterative Proportional Fitting (C-IPF) approach builds upon the regular Iterative Proportional Fitting (IPF) and the Discrete Flow Matching (DFM) framework for generating discrete data. While traditional IPF methods rely on continuous diffusion processes using Wiener processes W^ϵ, our approach replaces this with a discrete categorical noising process $\underline{\mathbb{R}^\alpha}$ within the DFM framework. This key difference enables more efficient modeling of discrete data distributions without requiring continuous embeddings or score function approximations.

The method alternates between learning forward and backward discrete flows that bridge the source distribution p_0 and target distribution p_1. It is worth noting that continuous state-space IPF can use reverse diffusion and stochastic analogies to flow matching (bridge matching). In that light, instead of using discrete state-space reverse diffusion, we use discrete flow matching [17].

3.1 Discrete Flow Matching Framework

We adapt the DFM approach from [17], which defines a probability path p_t interpolating between p_0 and p_1 through:

$$p_t(x) = \sum_{x_0, x_1 \in \mathcal{D}} p_t(x|x_0, x_1)\pi(x_0, x_1) \tag{1}$$

where $\pi(x_0, x_1)$ is a coupling between source and target distributions. The categorical noising process $\underline{\mathbb{R}^\alpha}$ serves as our discrete analog to the continuous Wiener process, with transition probabilities:

$$r(x_t = j|x_{t-1} = i) = \begin{cases} \alpha_t + \dfrac{1-\alpha_t}{K}, & \text{if } j = i, \\ \dfrac{1-\alpha_t}{K}, & \text{if } j \neq i. \end{cases} \tag{2}$$

Here K is the number of categories and α_t is a noising schedule controlling the rate of uniformization.

3.2 Forward and Backward Flow Learning

The IPF procedure alternates between learning two key components:

Backward DFM: Learns to denoise samples from p_1 through the $\mathbb{R}^\alpha$ process. This corresponds to estimating the probability velocity:

$$u_t^i(x^i, z) = \frac{\dot{\kappa}_t}{1-\kappa_t}\left[p_{1|t}(x^i|z) - \delta_z(x^i)\right] \tag{3}$$

where $p_{1|t}(x^i|z)$ is the probability denoiser.

Forward DFM: Learns to generate samples from p_1 starting from p_0 using:

$$u_t^i(x^i, z) = \frac{\dot{\kappa}_t}{\kappa_t}\left[\delta_z(x^i) - p_{0|t}(x^i|z)\right] \tag{4}$$

where $p_{0|t}(x^i|z)$ is the noise predictor.

3.3 Algorithm Implementation

The complete categorical IPF procedure is implemented as shown in Algorithm 2, which alternates between these two learning phases until convergence. Key features include:

- Minibatch processing for scalability
- Exact preservation of marginal constraints through the DFM framework
- Flexible noising schedule α_t control
- Parallel token updates enabled by discrete flow matching

The algorithm outputs both forward ($\mathbb{P}^{\text{SB, fw}}$) and backward ($\mathbb{P}^{\text{SB, bw}}$) Schrödinger bridge solutions, providing bidirectional mapping between distributions.

Algorithm 2: Categorical Iterative Proportional Fitting

Input: Samples from p_0: $\{p_0^{(i)}\}_{i=1}^n$, Samples from p_1: $\{p_1^{(i)}\}_{i=1}^n$
Output: SBP solution $\mathbb{P}^{\text{SB}}$

$\{x_0^{(i)}\}_{i=1}^m \leftarrow$ Draw minibatch from p_1;
$\underline{\mathbb{R}^\alpha}(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start $\underline{\mathbb{R}^\alpha}$ noising process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
$\mathbb{Q}^{\text{backward}} \leftarrow$ *Learn a backward DFM process on pairs* $[\{x_0^{(i)}\}_{i=1}^m, \mathbb{R}^\alpha(\{x_0^{(i)}\}_{i=1}^m)]$;
while *not converged* **do**
  $\{x_0^{(i)}\}_{i=1}^m \leftarrow$ Draw minibatch from p_0;
  $\mathbb{Q}^{\text{backward}}(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start $\mathbb{Q}^{\text{backward}}$ process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
  $\mathbb{Q}^{\text{forward}} \leftarrow$ *Learn a forward DFM process on pairs* $[\{x_0^{(i)}\}_{i=1}^m, \mathbb{Q}^{\text{backward}}(\{x_0^{(i)}\}_{i=1}^m), x_0^{(i)}]$;
  $\leftarrow$ Draw minibatch from p_1;
  $\mathbb{Q}^{\text{forward}}(\{x_0^{(i)}\}_{i=1}^m) \leftarrow$ Start $\mathbb{Q}^{\text{forward}}$ process from minibatch $\{x_0^{(i)}\}_{i=1}^m$;
  $\mathbb{Q}^{\text{backward}} \leftarrow$ *Learn a backward DFM process on pairs* $[\{x_0^{(i)}\}_{i=1}^m, \mathbb{Q}^{\text{forward}}(\{x_0^{(i)}\}_{i=1}^m),]$;
$\mathbb{P}^{\text{SB, fw}} \leftarrow \mathbb{Q}^{\text{forward}}$
$\mathbb{P}^{\text{SB, bw}} \leftarrow \mathbb{Q}^{\text{backward}}$

4 Results

4.1 2D distributions

We start by implementing IPF algorithm for generation of simple 2D distributions.

Firstly, we train a simple MLP to transfer points from one straight line to another. Figure **??** shows that trajectories generated by IPF-trained model have less intersections, than trajectories of regular Flow Matching. It means that IPF reduces transport cost.

Secondly, we train the backward flow matching to generate a 2D Swiss Roll from a mixture of 6 gaussians. The iterations of IPF algorithm are shown in the Figure 5. The resulting trajectories of the generated samples are shown in the Figure 4. The trajectories from the IPF-trained model appear to be straighter compared to the original Flow Matching model.

4.2 Color MNIST

We next evaluate the performance of our method on more complex data. We begin by training discrete Flow Matching on Color MNIST dataset in order to verify that chosen architecture will work in further experiments for C-IPF method. We observe that the model successfully maps samples from p_0 to samples from p_1 (see Figure 6). Note that proposed samples have different colors because Flow Matching does not obtain optimal transport mapping. Finally, we apply IPF algorithm based on Flow Mathing to Color MNIST. Figure 7 shows that, in contrast to regular Flow Mathing, the IPF-trained model preserves the color of numbers. This suggests that IPF-trained model minimizes transport cost and has straighter generation trajectories, thus requiring less generation steps.

5 Conclusion

This project explores optimal transport methods in discrete spaces. We have implemented IPF algorithm based on flow matching and validated its performance on small datasets. Future areas of work may include implementation of IPF based on categorical Shrödinger Bridge. A promising line of work is the application of the algorithm to larger and more complex datasets, especially to the textual modality.

References

[1] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

[2] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[3] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*.

[4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[5] Jun Hyeong Kim, Seonghwan Kim, Seokhyun Moon, Hyeongwoo Kim, Jeheon Woo, and Woo Youn Kim. Discrete diffusion schr\" odinger bridge matching for graph transformation. *arXiv preprint arXiv:2410.01500*, 2024.

[6] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces, 2023.

[7] Grigoriy Ksenofontov and Alexander Korotin. Categorical schr\" odinger bridge matching. *arXiv preprint arXiv:2502.01416*, 2025.

[8] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709, 2021.

[9] Francisco Vargas, Pierre Thodoroff, Austen Lamacraft, and Neil Lawrence. Solving schrödinger bridges via maximum likelihood. *Entropy*, 23(9):1134, 2021.

[10] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[11] Nikita Gushchin, Alexander Kolesov, Petr Mokrov, Polina Karpikova, Andrei Spiridonov, Evgeny Burnaev, and Alexander Korotin. Building the bridge of schrödinger: A continuous entropic optimal transport benchmark. *Advances in Neural Information Processing Systems*, 36:18932–18963, 2023.

[12] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

[13] Nikita Gushchin, Alexander Kolesov, Alexander Korotin, Dmitry P Vetrov, and Evgeny Burnaev. Entropic neural optimal transport via diffusion processes. *Advances in Neural Information Processing Systems*, 36:75517–75544, 2023.

[14] Yuyang Shi, Valentin De Bortoli, Andrew Campbell, and Arnaud Doucet. Diffusion schrödinger bridge matching. *Advances in Neural Information Processing Systems*, 36:62183–62223, 2023.

[15] Sergei Kholkin, Grigoriy Ksenofontov, David Li, Nikita Kornilov, Nikita Gushchin, Alexandra Suvorikova, Alexey Kroshnin, Evgeny Burnaev, and Alexander Korotin. Diffusion & adversarial schr\" odinger bridges via iterative proportional markovian fitting. *arXiv preprint arXiv:2410.02601*, 2024.

[16] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching, 2024.

[17] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 133345–133385. Curran Associates, Inc., 2024.

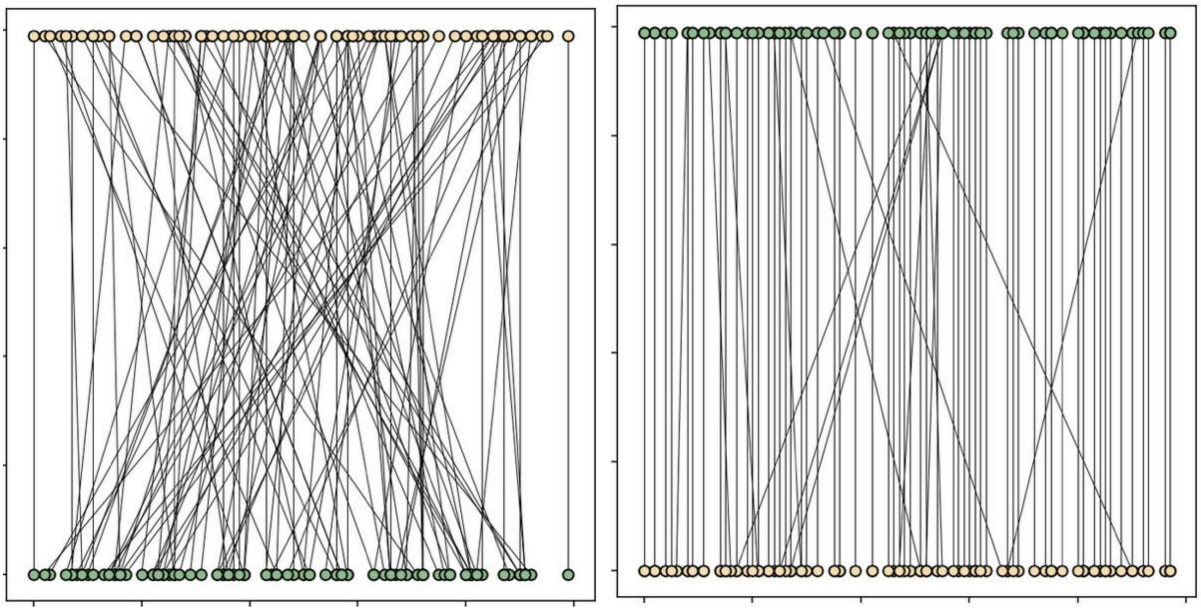


Figure 3. Trajectories of samples from Flow Matching (left) and IPF (right).

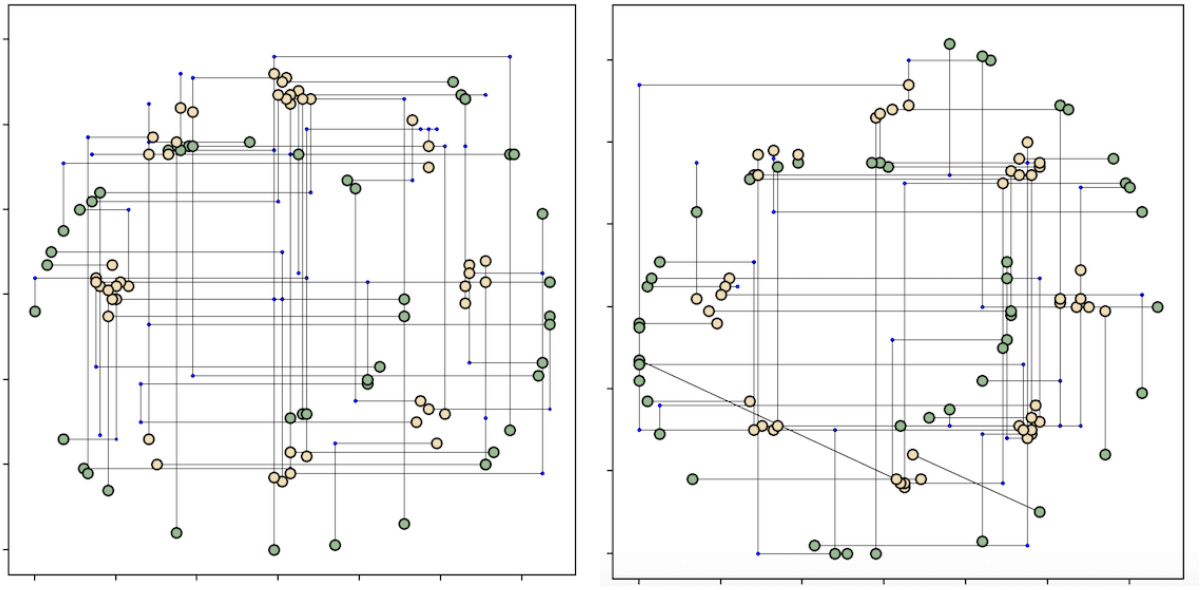


Figure 4. From 6 gaussians to Swiss Roll. Trajectories have become straighter after IPF.

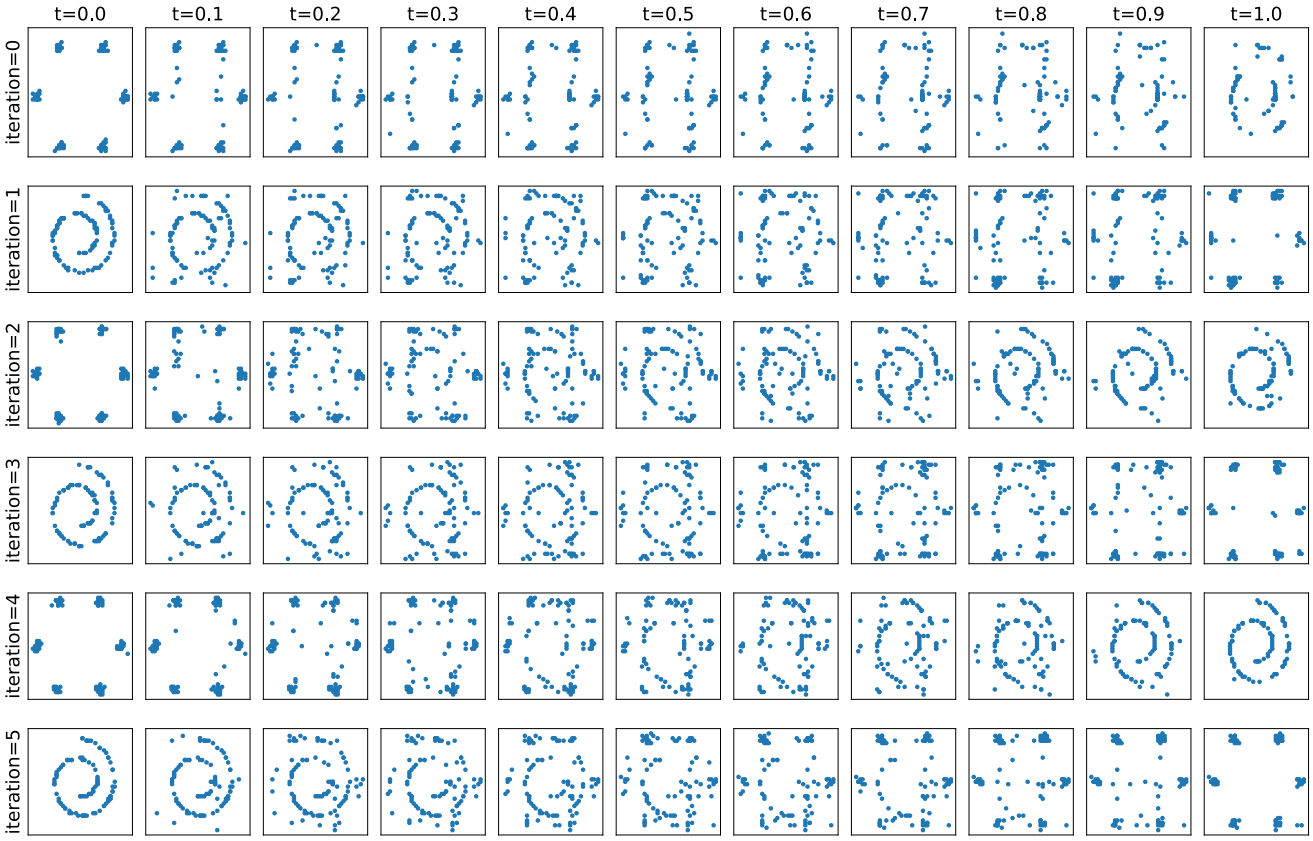


Figure 5. Evolution of distributions throughout IPF training. First six iterations are shown.

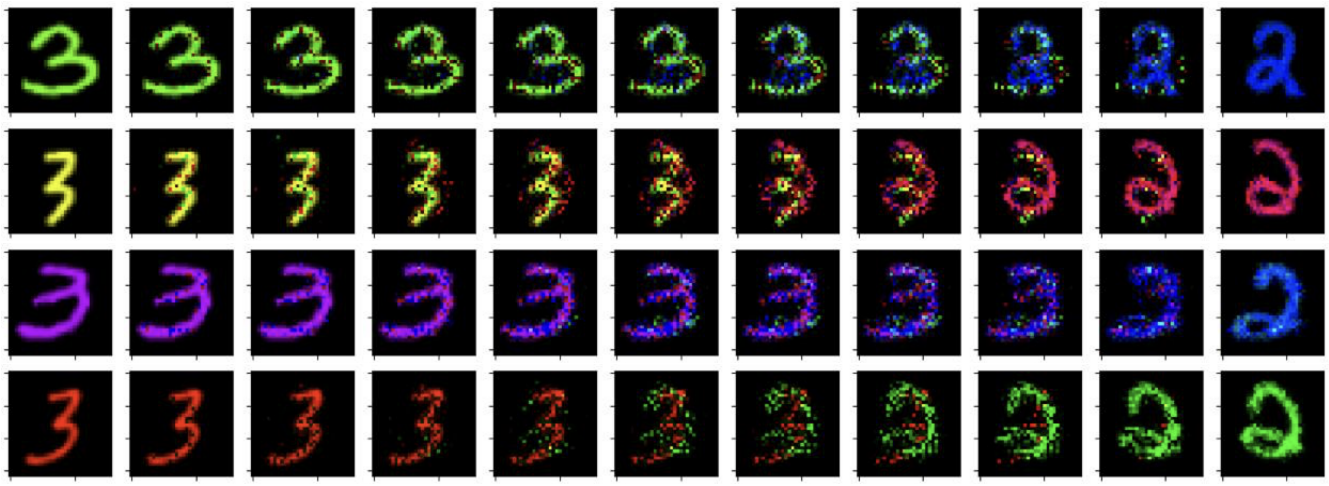


Figure 6. Sampling trajectory from learned Flow Matching model.

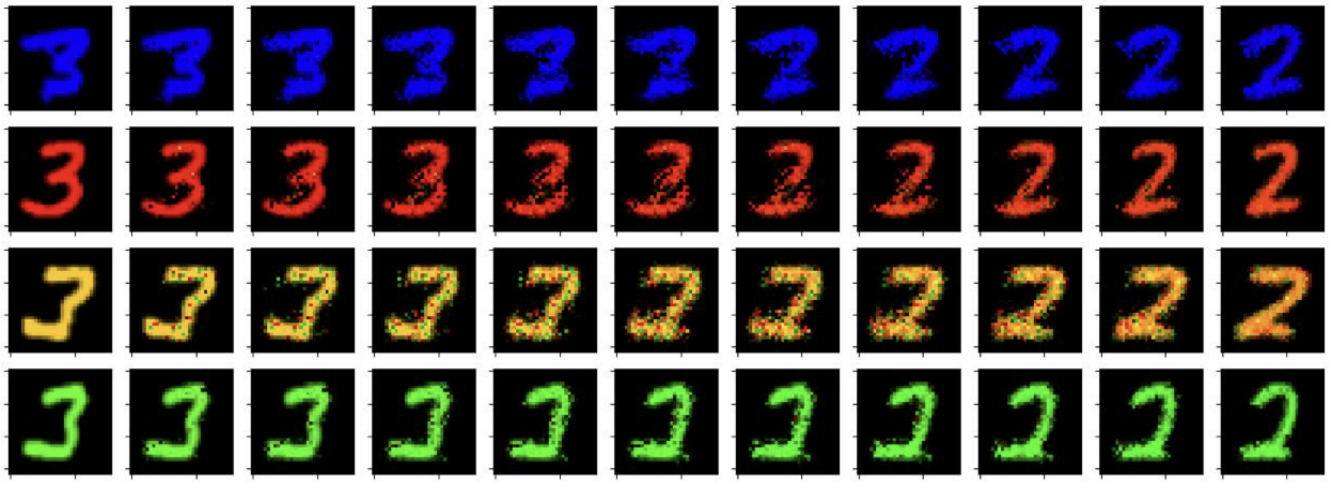


Figure 7. Sampling trajectory from the IPF-trained model.