ReSFlow: Rectifying Categorical Flows on Statistical Manifolds for One-Step Generation

Xiaohuan Jia[1]
xiaohuanjia@mail.imu.edu.cn

Renzhe Xu[2]
xurenzhe@sufe.edu.cn

Xiao Wang[2]
wangxiao@sufe.edu.cn

Jiayun Wu[3]
jiayunw@cmu.edu

Shaohua Fan[1]*
shfan@imu.edu.cn

[1]Inner Mongolia University; [2]Shanghai University of Finance and Economics;
[3]Carnegie Mellon University

Abstract

Generative modeling for discrete data has been significantly advanced by learning continuous vector fields on the statistical manifolds to transport simple priors to target discrete distributions. However, because these continuous-time flows are not constrained to follow geodesics on the statistical manifold, thereby requiring multi-step numerical integration for simulation and leading to substantial computational costs during inference. In this work, drawing inspiration from the "straight and fast" properties of Rectified Flow in continuous Euclidean space, we propose the Rectified Statistical Flow (ReSFlow) model, which enables one-step categorical generation on statistical manifolds by explicitly learning geodesic trajectories that adhere to the intrinsic Riemannian geometry. By rectifying the flow on the statistical manifold, we theoretically guarantee that trajectories between prior and target distributions approximate geodesics. This rectification significantly reduces the transport cost by aligning the learned trajectories directly along geodesics on the statistical manifold, allowing the model to map pure noise to target discrete data in a single step. Through extensive experiments on diverse real-world generative tasks, ranging from image and text to biological domains, we demonstrate ReSFlow's superior one-step generation quality, which truly embodies geodesically optimal and rapid categorical generation. Our code is available at `https://github.com/xiaohuanjia/ReSFlow`.

1 Introduction

Generative modeling has advanced rapidly, with diffusion and flow-based methods achieving state-of-the-art performance across continuous domains such as image synthesis [1, 2], molecule and protein design [3, 4, 5], and sequence generation [6, 7]. A key idea is to learn continuous-time dynamics, especially through flow matching, which formulates ODEs with continuous vector fields and provides a simulation-free, theoretically grounded training framework [8]. However, extending these methods to discrete data—common in NLP, drug discovery, and protein engineering—remains difficult, since discrete variables do not have the differentiable structure required by continuous flows. Most existing flow-based generative models for discrete data act directly on the native discrete space by optimizing the categorical probabilities of tokens [9, 10, 11, 12]. Rather than operating strictly within this discrete realm, a promising direction maps discrete variables to statistical manifolds, treating each categorical distribution as a point on the probability simplex with Riemannian geometry [13]. This allows smooth

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

vector fields and continuous-time flows to transport priors to target discrete distributions, bridging continuous generative modeling and discrete data synthesis.

However, a critical limitation persists: The learned ODE trajectories deviate from geodesics because the conditional vector fields arising from independent couplings of prior and target samples yield a mixture of intersecting geodesics that, once marginalized, produce highly curved aggregate flows not aligned with any single geodesic path [14]. Simulating these tortuous trajectories demands multi-step numerical integration with fine temporal discretization, incurring substantial computational overhead during inference. This stands in stark contrast to the desiderata of modern generative systems, where rapid, ideally single-step, sampling is essential for scalable deployment. The fundamental tension between the geometric complexity of learned flows and the practical need for efficient generation thus constitutes a major bottleneck for categorical generative models.

In the continuous Euclidean setting, an elegant solution to this problem has emerged through the concept of Rectified Flow [15]. Rectified Flow enforces straight-line trajectories between paired noise and data samples, yielding flows that are both geometrically simple and computationally efficient to simulate. The resulting "straight and fast" property enables high-quality generation with very few or even a single integration step, dramatically accelerating inference. Naturally, one might ask: *can we bring the same benefits of trajectory rectification to categorical generation on statistical manifolds?*

The answer, however, is far from straightforward. The probability simplex, when equipped with the Fisher-Rao metric [16], forms a Riemannian manifold whose geometry is fundamentally non-Euclidean. Straight paths in ambient Euclidean space do not respect the intrinsic curvature of this manifold—they may leave the simplex, violate probability constraints, or deviate substantially from the true shortest paths. In this curved geometric setting, the natural analogue of a "straight path" is a geodesic: the length-minimizing curve that faithfully adheres to the manifold's intrinsic Riemannian structure. Consequently, achieving efficient one-step categorical generation requires not Euclidean straightening, but rather a fundamentally geometric approach that aligns learned trajectories with the geodesics of the underlying statistical manifold.

In this work, we propose Rectified Statistical Flow (ReSFlow), a framework that enables one-step discrete data generation by rectifying flows on statistical manifolds. The core idea is to map data from the probability simplex onto the positive orthant of the hypersphere via their natural diffeomorphism, and then perform flow rectification in this spherical space where geodesics admit closed-form great-circle expressions. By iteratively recoupling the learned ODE marginals and retraining the vector field on the hypersphere, ReSFlow progressively straightens transport paths toward geodesics, reducing both transport cost and path curvature. Extensive evaluations across vision, NLP, and bioinformatics benchmarks demonstrate that ReSFlow, requiring only a single sampling step, matches or outperforms state-of-the-art methods that rely on hundreds of steps.

2 Preliminaries

Notations. Let $\mathcal{X} = \{1, \dots, K\}$ be a finite categorical state space and $\mathcal{P}(\mathcal{X})$ the probability simplex over $\mathcal{X}$, i.e. the set of all categorical distributions. We use $\mu, \nu \in \mathcal{P}(\mathcal{X})$ to denote individual categorical distributions, and $p, q \in \mathcal{P}(\mathcal{P}(\mathcal{X}))$ to denote distributions over categorical distributions. We let $p_0, q \in \mathcal{P}(\mathcal{P}(\mathcal{X}))$ denote the source (prior) and target (data) distributions, respectively. We write Δ^{K-1} for the $(K-1)$-simplex and $\mathbb{S}_+^{K-1}$ for the positive orthant of the unit hypersphere in $\mathbb{R}^K$. The square-root map $\pi\colon \Delta^{K-1} \to \mathbb{S}_+^{K-1}$, defined by $\pi(\mu)_i = \sqrt{\mu_i}$, is a diffeomorphism that renders the Fisher–Rao metric on Δ^{K-1} isometric to the round metric on $\mathbb{S}_+^{K-1}$; we overload notation and write $\pi(p)$ for the pushforward of a distribution $p \in \mathcal{P}(\mathcal{P}(\mathcal{X}))$ onto $\mathbb{S}_+^{K-1}$.

2.1 Statistical Flow Matching

Statistical Flow Matching (SFM) [13] performs flow matching for discrete categorical data directly on the statistical manifold induced by the Fisher information metric. Since the Fisher inner product is ill-defined on the boundary of Δ^{K-1}, SFM operates on $\mathbb{S}_+^{K-1}$ via the square-root map π (see Appendix A for detailed motivations). Specifically, we independently draw $\mu_0 \sim p_0$ and $\mu_1 \sim q$ and map them to $X_0 = \pi(\mu_0)$, $X_1 = \pi(\mu_1)$ on $\mathbb{S}_+^{K-1}$. SFM constructs a conditional path via geodesic

interpolation:

$$X_t := \psi_t(X_0, X_1) = \exp_{X_0}\big(t \log_{X_0} X_1\big), \quad t \in [0,1], \tag{1}$$

where exp and log are the Riemannian exponential and logarithmic maps on $\mathbb{S}^{K-1}$. Letting $\theta = \arccos(X_0^\top X_1)$ denote the geodesic distance, the conditional vector field $u_t = \frac{\mathrm{d}}{\mathrm{d}t} X_t$ is

$$u_t^{\mathbb{S}}(X_t \mid X_0, X_1) = \frac{\theta}{\sin\theta}\big[-\cos\big((1-t)\theta\big)\, X_0 + \cos(t\theta)\, X_1\big]. \tag{2}$$

A parameterised velocity field $v_\theta(X_t, t)$ is trained by minimizing:

$$\mathcal{L}_{\mathrm{SFM}} = \mathbb{E}_{t,\, X_0,\, X_1}\big[\|u_t^{\mathbb{S}}(X_t \mid X_0, X_1) - v_\theta(X_t, t)\|_2^2\big], \tag{3}$$

where $t \sim U[0,1]$, $X_0 \sim \pi(p_0)$, $X_1 \sim \pi(q)$. By exploiting the intrinsic geometry of $\mathbb{S}_+^{K-1}$, SFM ensures the flow remains on the simplex and provides a principled optimal transport path, avoiding quantisation issues prevalent in Euclidean treatments of categorical data.

2.2 Rectified Flow

Rectified Flow [15] learns straight ODE trajectories that transport a source distribution ρ_0 to a target distribution ρ_1, with an iterative *reflow* procedure that progressively straightens the paths to enable few-step or even one-step generation. Originally formulated in $\mathbb{R}^d$, we present it here in our unified notation; in Section 3 we extend it to $\mathbb{S}_+^{K-1}$. Given a coupling (μ_0, μ_1) with $\mu_0 \sim p_0$ and $\mu_1 \sim q$, Rectified Flow defines the linear interpolation path $X_t = t\,\mu_1 + (1-t)\,\mu_0$ for $t \in [0,1]$ and learns a velocity model v_θ by minimizing

$$\hat{\theta} = \arg\min_\theta \ \mathbb{E}\big[\|\mu_1 - \mu_0 - v_\theta(X_t,\, t)\|_2^2\big], \quad t \sim \mathrm{Uniform}([0,1]). \tag{4}$$

Once trained, new samples are generated by solving the ODE $\mathrm{d}Z_t = v_{\hat{\theta}}(Z_t, t)\,\mathrm{d}t$ forward from $Z_0 \sim p_0$ to obtain Z_1, yielding a full trajectory $\mathbf{Z} = \{Z_t : t \in [0,1]\}$. A distinguishing feature of Rectified Flow is the *reflow* operation. Starting from an initial independent coupling $(Z_0^0, Z_1^0) = (\mu_0, \mu_1)$, the k-th reflow step applies the above procedure to the induced coupling (Z_0^k, Z_1^k), whose endpoints are obtained by simulating the previously learned flow: $(Z_0^{k+1}, Z_1^{k+1}) = \texttt{RectFlow}\big((Z_0^k, Z_1^k)\big)$.

Each reflow iteration reduces transport cost and trajectory curvature, bringing the induced coupling closer to a deterministic map. In the limit, $v_\theta(Z_t, t)$ is nearly constant along each path and the ODE can be solved in a single Euler step. Although reflow successfully straightens trajectories in Euclidean space, it remains unexplored how to extend it to non-Euclidean manifolds such as $\mathbb{S}_+^{K-1}$, where straight lines must be replaced by geodesics. More discussion on related work is in Appendix B.

3 Method

As discussed above, the extension of rectified flow to statistical manifolds poses additional challenges, as the model needs to be fully aware of the intrinsic geometry induced by the Fisher information metric to properly define and straighten the ODE trajectories. In particular, the velocity field estimation and the evolution of marginal distributions—both central to the rectified flow framework—must be reformulated to respect the curvature and geometric structure of the statistical manifold. To address these challenges, we propose the Rectified Categorical Flow model, which reflows and straightens the ODE trajectories along geodesics on a mapped hyperspherical manifold.

3.1 Why Euclidean Reflow Fails on the Statistical Manifold

Before presenting our approach, we highlight why a naïve application of Euclidean rectified flow to categorical distributions is fundamentally inadequate. Recall that standard rectified flow defines straight-line interpolations $X_t = t\,\mu_1 + (1-t)\,\mu_0$ in $\mathbb{R}^d$ and trains the velocity field to match the constant direction $\mu_1 - \mu_0$. Reflow guarantees that iterative re-coupling yields non-crossing straight paths, enabling single-step generation. However, when the data live on the probability simplex Δ^{K-1} equipped with the Fisher–Rao metric, this guarantee breaks down:

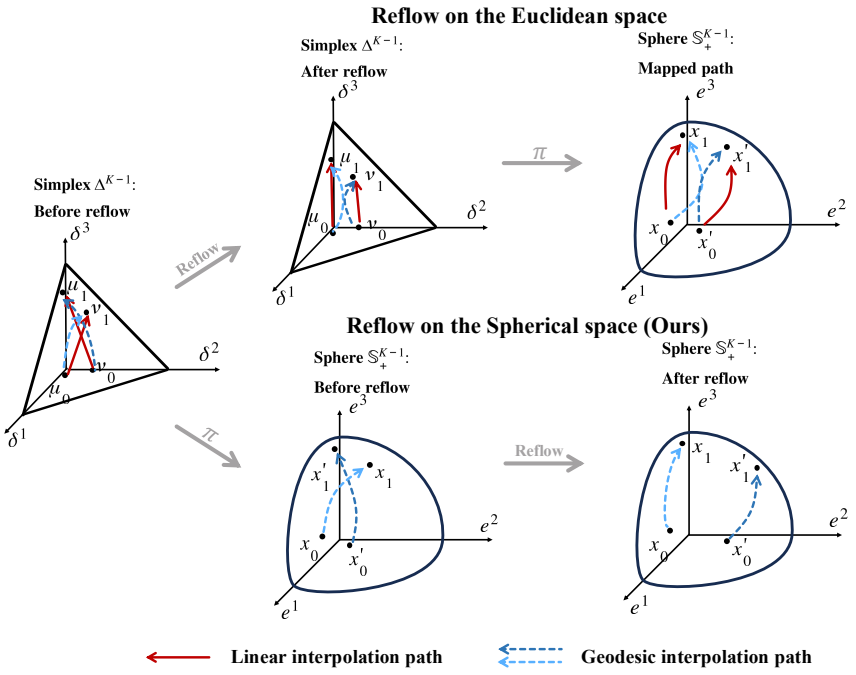


Figure 1: **Overview of Spherical Rectified Flow.** *Top:* Standard reflow straightens linear paths on the simplex, but their corresponding spherical geodesics still intersect, preventing one-step sampling. *Bottom:* We perform reflow directly on the sphere $\mathbb{S}_+^{K-1}$. This straightens the geodesics themselves to be non-intersecting, ensuring faithful one-step generation.

Proposition 3.1 (Non-preservation of non-crossing under Fisher–Rao geodesics)**.** *Let* $\Delta^{K-1} = \{\mu \in \mathbb{R}^K : \mu_i > 0,\ \sum_i \mu_i = 1\}$ *be the open probability simplex with Fisher–Rao metric* $g_{ij}(\mu) = \delta_{ij}/\mu_i$. *For two pairs* (μ_0, μ_1) *and* (ν_0, ν_1) *in* Δ^{K-1}, *let*

$$\ell_\mu(t) = (1-t)\,\mu_0 + t\,\mu_1, \quad \ell_\nu(t) = (1-t)\,\nu_0 + t\,\nu_1, \quad t \in [0,1] \tag{5}$$

be the linear interpolations, and $\gamma_\mu(t)$, $\gamma_\nu(t)$ *the Fisher–Rao geodesics. Then for* $K \geq 3$,

$$\ell_\mu([0,1]) \cap \ell_\nu([0,1]) = \emptyset \quad \not\Rightarrow \quad \gamma_\mu([0,1]) \cap \gamma_\nu([0,1]) = \emptyset. \tag{6}$$

Proof sketch. Under the square-root embedding $\pi\colon \mu \mapsto (\sqrt{\mu_1}, \ldots, \sqrt{\mu_K}) \in \mathbb{S}_+^{K-1}$, Fisher–Rao geodesics become great-circle arcs: $\pi(\gamma_\mu(t)) = \frac{\sin((1-t)\beta_\mu)}{\sin\beta_\mu}\,\pi(\mu_0) + \frac{\sin(t\,\beta_\mu)}{\sin\beta_\mu}\,\pi(\mu_1)$, where $\beta_\mu = \arccos(\sum_i \sqrt{\mu_{0,i}\,\mu_{1,i}})$. The nonlinear map π does not preserve the non-crossing property: geodesics curve toward the interior of Δ^{K-1} relative to linear paths, creating intersections even when the straight paths are separated. □

Proposition 3.1 reveals that Euclidean reflow straightens paths in $\mathbb{R}^K$ so that linear interpolation path no longer cross, yet the induced geodesics on the statistical manifold can still intersect (Fig. 1, top). This motivates our approach: we perform reflow directly on $\mathbb{S}_+^{K-1}$, straightening the geodesic paths themselves to obtain non-crossing trajectories that enable single-step sampling (Fig. 1, bottom).

3.2 Geodesic Interpolation on the Hypersphere

Leveraging the square-root diffeomorphism $\pi : \Delta^{K-1} \to \mathbb{S}_+^{K-1}$, we lift all operations to the positive orthant of the unit hypersphere, where the Fisher-Rao geometry becomes isometric to the standard spherical geometry. On $\mathbb{S}_+^{K-1}$, geodesics are great-circle arcs, and the geodesic interpolation between a coupling (X_0, X_1) on sphere $\mathbb{S}_+^{K-1}$ is given by the spherical linear interpolation:

$$X_t = \mathrm{slerp}(X_0, X_1; t) = \frac{\sin((1-t)\theta)}{\sin(\theta)} X_0 + \frac{\sin(t\theta)}{\sin(\theta)} X_1, \quad t \in [0,1], \tag{7}$$

where $\theta = \arccos(X_0^\top X_1)$ is the geodesic distance between X_0 and X_1. This interpolation has several desirable properties: (i) X_t remains on $\mathbb{S}_+^{K-1}$ for all $t \in [0,1]$ by construction, automatically satisfying the simplex constraints after the inverse map π^{-1}; (ii) the curve $t \mapsto X_t$ is a constant-speed geodesic with $\|\dot{X}_t\| = \theta$ for all t; and (iii) the tangent vector along this path has the closed-form expression given in Equation (2).

Our key observation is that, just as Euclidean rectified flow uses linear interpolation $tX_1 + (1-t)X_0$ as its "straight" path, the natural equivalent on the hypersphere is the geodesic *slerp*. Consequently, a flow with geodesic integral curves maintains a constant velocity magnitude, enabling exact simulation in a single Euler step.

3.3 Rectified Categorical Flow via Spherical Reflow

We now present the core of our method: the iterative reflow procedure on $\mathbb{S}_+^{K-1}$ that progressively straightens the learned ODE trajectories toward geodesics. The procedure consists of two alternating stages, followed by an efficient inference phase.

Stage 1: Initial Flow Training ($k = 0$). We begin with an independent coupling $(X_0, X_1) \sim \pi(p_0) \times \pi(q)$. We train a velocity field v_θ by minimizing the spherical flow matching objective:

$$\mathcal{L}_{\text{ReSFlow}}^{(0)} = \mathbb{E}_{t,X_0,X_1}\left[\left\|v_\theta(X_t,t) - u_t^S(X_t \mid X_0, X_1)\right\|_2^2\right], \tag{8}$$

where $X_t = \text{slerp}(X_0, X_1; t)$ is the geodesic interpolant. Crucially, $v_\theta(X_t, t)$ is constrained to the tangent space $\mathcal{T}_{X_t}\mathbb{S}^{K-1}$ to ensure the flow remains on the manifold (details in Appendix C.1).

At this stage, the marginal velocity field v_θ approximates the conditional expectation $\mathbb{E}[u_t^S \mid X_t]$. Because the independent coupling produces a mixture of intersecting geodesics, averaging their tangent vectors at intersection points yields a direction that deviates from any single geodesic. Consequently, the resulting ODE trajectories are curved, requiring many steps to simulate accurately.

Stage 2: Trajectory-Induced Recoupling (Reflow). To straighten the trajectories, we exploit the uniqueness of ODE solutions. Given the trained model $v_{\hat{\theta}}^{(k)}$, we simulate the ODE forward on $\mathbb{S}_+^{K-1}$:

$$\frac{dX_t}{dt} = v_{\hat{\theta}}^{(k)}(X_t, t), \quad X_0 \sim \pi(p_0), \quad t \in [0,1]. \tag{9}$$

This simulation produces new endpoint pairs (Z_0^{k+1}, Z_1^{k+1}). We then train the $(k+1)$-th rectified flow using the geodesic interpolation between these recoupled pairs:

$$\mathcal{L}_{\text{ReSFlow}}^{(k+1)} = \mathbb{E}_{t,Z_0^{k+1},Z_1^{k+1}}\left[\left\|v_\theta^{(k+1)}(Z_t^{k+1},t) - u_t^S(Z_t^{k+1} \mid Z_0^{k+1}, Z_1^{k+1})\right\|_2^2\right]. \tag{10}$$

The deterministic ODE defines an injective transport map. Because this map is continuous, nearby points in the source distribution are mapped to nearby points in the target. Thus, the geodesics connecting the recoupled pairs tend to run parallel rather than intersect. With strictly fewer crossing geodesics, the conditional expectation at each point is dominated by a single geodesic direction, making the updated velocity field closer to a pure geodesic field. The complete reflow iteration on $\mathbb{S}_+^{K-1}$ can be summarized as: $(Z_0^{k+1}, Z_1^{k+1}) = \texttt{RectFlow}_{\mathbb{S}}\big((Z_0^k, Z_1^k)\big)$.

Inference: Efficient One-Step Generation. Once the iterative reflow procedure converges, the learned vector field v_θ effectively models non-intersecting, constant-speed geodesics. Given a prior sample $X_0 \sim \pi(p_0)$, we can project the initial velocity onto the manifold to generate the target data sample X_1 in a single step via the spherical exponential map:

$$\hat{X}_1 = \text{Exp}_{X_0}(v_\theta(X_0, 0)) = X_0 \cos(\|v_0\|_2) + \frac{v_0}{\|v_0\|_2}\sin(\|v_0\|_2), \tag{11}$$

where $v_0 = v_\theta(X_0, 0)$. This formulation bridges continuous-time ODEs with discrete geometric operations, unlocking highly accurate one-step generation for categorical data. The final discrete sample is obtained by applying the inverse square-root map: $\hat{p} = \pi^{-1}(\hat{X}_1) = (\hat{X}_{1,1}^2, \ldots, \hat{X}_{1,K}^2) \in \Delta^{K-1}$, followed by taking the argmax (or sampling from) this categorical distribution. We summarize the complete Rectified Categorical Flow pipeline in Algorithm 1.

3.4 Theoretical Guarantees of Spherical Reflow

To ground our geometric intuition, we establish formal theoretical guarantees for the spherical reflow process based on the framework of non-Euclidean probability transport. Detailed proofs for all statements are provided in Appendix D.

First, a valid generative model must ensure that modifying the trajectories does not alter the target data distribution. We confirm that spherical reflow strictly preserves the marginal distributions.

Theorem 3.1 (Spherical Marginal Preservation; Informal; See Theorem D.1)**.** *Let X_t be the spherical geodesic interpolation process. If the flow is rectifiable, its induced spherical rectified flow Z_t preserves the marginals:* $\mathrm{Law}(Z_t) = \mathrm{Law}(X_t)$ *for all* $t \in [0, 1]$.

To formalize trajectory straightness, we define the Spherical Transport Cost for a coupling (X_0, X_1) on $\mathbb{S}_+^{K-1}$ as $\mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big] = \mathbb{E}[\arccos(X_0^\top X_1)^2]$, which measures the expected squared geodesic distance. Reflow theoretically guarantees the optimization of this transport cost.

Theorem 3.2 (Contraction of Spherical Transport Cost; Informal; See Theorem D.2)**.** *Let (Z_0^k, Z_1^k) be the coupling induced by the k-th reflow iteration. The spherical transport cost is non-increasing:*

$$\mathbb{E}\big[d_{\mathbb{S}}(Z_0, Z_1)^2\big] \le \mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big]. \tag{12}$$

Proof. Proof sketch in Appendix D.3: The ODE trajectories have finite length, and the geodesic connecting the recoupled endpoints represents the absolute shortest path on the manifold, enforcing a monotonic decrease in transport cost. □

Finally, we provide the following theorem shows that iterative spherical rectification progressively straightens trajectories at a guaranteed rate:

Theorem 3.3 (Spherical Reflow Straightening Rate; Informal; See Theorem D.4)**.** *Let (X_0, X_1) be a coupling on $\mathbb{S}_+^{K-1}$ with $\mathbb{E}[d_{\mathbb{S}}(X_0, X_1)^2] < +\infty$. Set $(Z_0^0, Z_1^0) = (X_0, X_1)$ and $Z^{k+1} = \texttt{RectFlow}_{\mathbb{S}}((Z_0^k, Z_1^k))$ for $k = 0, \dots, N_{\mathrm{reflow}} - 1$. Then*

$$\sum_{k=0}^{N_{\mathrm{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] \le \mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big], \tag{13}$$

where $S_{\mathbb{S}} \ge 0$ measures the excess kinetic energy over the geodesic cost and $V_{\mathbb{S}} \ge 0$ measures the velocity variance. Consequently,

$$\min_{0 \le k \le N_{\mathrm{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] = \mathcal{O}(1/N_{\mathrm{reflow}}). \tag{14}$$

See Appendix D.4 for the detailed proof.

This theoretical framework confirms that combining hyperspherical geometry with iterative reflow explicitly forces the learned vector fields to approximate constant-speed geodesics, providing the rigorous foundation for our highly accurate one-step generation via the exponential map. Theoretical results are empirically validated in Appendix F.

4 Experiments

4.1 Experimental setup

Our proposed ReSFlow models arbitrary distributions over categorical probability measures and enables one-step sampling via geodesic rectification on the statistical manifold. We first validate this property on a synthetic Swiss-roll distribution on the simplex, and then evaluate ReSFlow on real-world discrete generation tasks, including binarized MNIST image generation, character-level Text8 modeling, and conditional promoter DNA sequence design. Our primary comparison is with SFM [13], which achieves state-of-the-art performance on these benchmarks. The additional baselines reported in that work rely on multi-step sampling and do not outperform SFM; we therefore do not repeat them here. To disentangle the effects of geometry and rectification, we compare four controlled variants: *Linear Flow Matching* (LinearFM) [13], which uses Euclidean linear interpolation on

Table 1: Swiss-roll results on the 2-simplex. We report ODE-based NLL and ODE sampling time for generating 1,000 samples. Bold indicates the best result, and '$*$' indicates the second-best result.

Metric	LinearFM	SFM [13]	R-LFM	ReSFlow
NLL ↓	-1.88 ± 0.06	**-2.38 ± 0.08**	-1.87 ± 0.06	-2.25 ± 0.08*
Time (s) ↓	0.1059	0.0679	0.0512*	**0.0361**

the simplex; *Statistical Flow Matching* (SFM), which maps the simplex to the positive orthant of the hypersphere and follows spherical geodesics; *Rectified Linear Flow Matching* (R-LFM), which applies rectification in the Euclidean setting; and *Rectified Statistical Flow* (ReSFlow), our proposed spherical reflow model. This 2×2 design separates the choice of geometry, Euclidean versus spherical, from the effect of rectification. Moreover, we also compare with SOTA flow-based one-step generation ReDi [11], which rectifies the flow trajectories of continuous token embeddings in Euclidean space. The fundamental difference between their setting and ours lies in the data preprocessing requirement. While their method relies on an external pre-trained model to tokenize the input data, our ReSFlow operates as a fully end-to-end framework directly on the raw data space. By bypassing the discrete tokenization step, our approach is highly advantageous for domains and modalities where well-established pre-trained tokenizers are unavailable. Due to the absence of well-established tokenizers for binarized MNIST and promoter DNA sequences, we restrict our comparison with ReDi solely to the Text8, where characters are tokenized using the GPT-2 tokenizer.

We evaluate all methods under different numbers of function evaluations (NFE). For sample-quality metrics such as FID [17] and SP-MSE, we report both one-step generation with NFE$= 1$ and accurate ODE simulation using the adaptive Dormand–Prince 5 (Dopri5) solver [18]. For the Swiss-roll benchmark, we report only Dopri5-based results, including ODE-based NLL and Dopri5 sampling time. Likelihood-based metrics are evaluated only with ODE integration: NLL estimation uses the continuous-time change-of-variables formula and therefore requires accurate integration of the learned dynamics, while a single Euler or exponential-map step does not yield reliable likelihoods. For Text8, BPC is computed from the variational integral formulation rather than generated samples, and is therefore not tied to one-step sampling. Thus, one-step sampling measures fast generation quality, whereas ODE-based integration evaluates NLL and BPC. We also report wall-clock sampling time for binarized MNIST, promoter sequence design, and Swiss-roll to quantify practical runtime , but omit timing for Text8, since the reported BPC metric is based on likelihood evaluation rather than sample generation. Detailed architectures and hyperparameters are provided in Appendix E.

4.2 Toy Example: Swiss Roll on the Simplex

We first evaluate ReSFlow on a Swiss-roll distribution on the 2-simplex as a low-dimensional geometric sanity check. Following Cheng et al. [13], we normalize a two-dimensional Swiss roll and embed it into the simplex by setting $\mu_3 = 1 - \mu_1 - \mu_2$. All methods use the same time-dependent MLP and training budget. LinearFM and R-LFM use Euclidean interpolation on the simplex; SFM and ReSFlow use the square-root map with spherical geodesic interpolation on $\mathbb{S}^2_+$. This controlled comparison separates the effect of geometry from the effect of rectification. As shown in Table 1, ReSFlow achieves an NLL of -2.25, improving over LinearFM (-1.88) and R-LFM (-1.87), and remains competitive with SFM (-2.38). It also attains the fastest ODE sampling time among all methods, reducing the runtime from 0.0679s for SFM to 0.0361s. This indicates that ReSFlow preserves the geometric advantage of Fisher–Rao geodesics and enabling efficient ODE generation.

The samples in Fig. 2 further illustrate the role of spherical reflow under one-step sampling. Non-rectified LinearFM and SFM fail to recover the Swiss-roll structure in the one-step setting, because their marginal vector fields average over many crossing conditional paths, producing curved trajectories that are poorly approximated by a single update. Euclidean reflow improves one-step behavior relative to LinearFM, but it still rectifies paths under the wrong geometry, leading to samples that only partially capture the target structure. ReSFlow instead performs reflow directly on the hypersphere, aligning the learned trajectories with Fisher–Rao geodesics and producing the most faithful one-step samples. This supports the central claim that geometry and rectification are complementary: the hyperspherical formulation provides the correct transport geometry, while reflow reduces trajectory curvature for accurate one-step sampling.

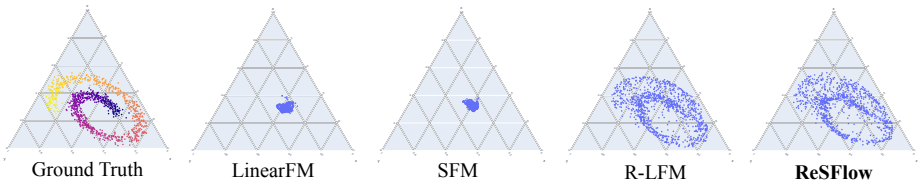


Figure 2: Swiss-roll samples on the simplex under one-step Euler sampling.

Table 2: Binarized MNIST results. Mean $\pm$ standard deviation over three random seeds; time is for generating 1,000 samples. Bold indicates the best mean, and $*$ indicates the second-best mean.

Method	ODE (NFE = adaptive)		Euler one-step (NFE = 1)	
	FID ↓	Time / 1k samples (s) ↓	FID ↓	Time / 1k samples (s) ↓
LinearFM [13]	$5.187 \pm 0.129^{*}$	87.13 ± 0.44	25.317 ± 0.517	$\mathbf{0.406 \pm 0.178}$
SFM [13]	$\mathbf{4.748 \pm 0.092}$	112.81 ± 8.20	35.640 ± 0.487	0.719 ± 0.175
R-LFM	5.681 ± 0.252	$61.19 \pm 0.28^{*}$	$6.261 \pm 0.379^{*}$	0.583 ± 0.244
ReSFlow	5.541 ± 0.252	$\mathbf{36.07 \pm 0.60}$	$\mathbf{5.973 \pm 0.442}$	$0.452 \pm 0.243^{*}$

4.3 Binarized MNIST

We next evaluate on binarized MNIST to test whether the geometric and rectification effects observed in the Swiss-roll experiment extend to high-dimensional discrete generation. Each image is modeled as a collection of Bernoulli categorical variables, so good samples require both local binary consistency and global digit structure. This benchmark therefore allows us to examine two questions in a realistic setting: whether Fisher–Rao geometry improves categorical transport, and whether reflow makes the learned dynamics accurate under NFE = 1 sampling. Following the evaluation protocol above, we compare adaptive ODE sampling with Euler one-step generation.

Table 2 reports mean FID and wall-clock sampling time over three random seeds. Under adaptive ODE sampling, SFM achieves the best FID, confirming that spherical geodesic interpolation provides a stronger reference geometry than Euclidean interpolation for binary categorical data. ReSFlow is slightly worse than SFM in this full-ODE regime, but remains competitive and is the fastest ODE-based method. Under one-step sampling, non-rectified LinearFM and SFM degrade substantially, indicating that their learned trajectories are too curved to be approximated by a single update. Rectification alleviates this issue, and ReSFlow achieves the best one-step FID by performing reflow under the hyperspherical Fisher–Rao geometry. Compared with SFM under accurate ODE sampling, one-step ReSFlow achieves an orders-of-magnitude improvement in sampling speed (0.452 vs. 112.81 seconds per 1,000 samples), while incurring only a mild and acceptable degradation in FID (5.973 vs. 4.748). These results show that ReSFlow improves trajectory geometry and enables efficient one-step sampling.

Figure 3 further supports the benefit of reflow under the Fisher–Rao geometry. ReSFlow achieves a mean FID of 5.973 ± 0.442 with a single function evaluation, while SFM requires 34 evaluations to obtain a comparable result (5.922 ± 0.262). This reduction in sampling steps shows that rectification makes the learned dynamics more suitable for one-step generation.

4.4 Text8

We further evaluate ReSFlow on character-level language modeling with Text8 [19], where each token is a categorical variable and the sequence distribution is substantially more structured than the toy and image settings. This experiment is designed to test whether the proposed manifold-aware rectification remains effective for high-entropy discrete sequences with a larger alphabet and stronger contextual dependencies. Since text generation is typically evaluated by predictive coding efficiency, we report bits per character (BPC), which naturally reflects whether the learned categorical transport is efficient and sufficiently well aligned with the data distribution: a lower BPC indicates that the model assigns higher probability to the observed character sequences and induces less distorted

transport over categorical states. Since BPC is evaluated from the continuous-time likelihood, it requires ODE integration; we therefore report only ODE-based BPC in Table 3.

Table 3 shows that the choice of geometry has a pronounced effect on categorical sequence modeling. SFM substantially improves over LinearFM, reducing BPC from 2.154 to 1.830, which supports the importance of modeling categorical data under the Fisher–Rao geometry rather than using Euclidean interpolation on the simplex. Moreover, R-LFM performs worse than LinearFM, increasing BPC to 3.267. This agrees with the failure mode discussed in Section 3.1: rectifying paths in the ambient Euclidean space optimizes the wrong notion of straightness and transport cost. The effect is especially harmful for character-level sequences, where local geometric errors can accumulate over many categorical variables and time steps, ultimately degrading the likelihood. Furthermore, although ReDi leverages pre-trained token embeddings to map discrete data into a continuous space, its Euclidean rectification only yields a BPC of 2.306, exposing a substantial performance gap compared to our ReSFlow (1.261). This sharp contrast underscores that relying on pre-trained continuous representations cannot compensate for geometric mismatches, proving the necessity of optimizing discrete data strictly within its suitable intrinsic space. By performing rectification under the correct manifold geometry, ReSFlow achieves the best overall BPC, confirming the effectiveness of our approach. Compared with binarized MNIST, Text8 has a larger categorical alphabet and richer sequential dependencies, so accumulated geometric errors are more likely to affect the final likelihood. By combining hyperspherical statistical-manifold modeling with reflow, ReSFlow preserves the intrinsic geometric advantage of SFM while further aligning the learned trajectories with geodesic transport on the manifold. These results support our design choice: the manifold formulation captures the proper geometry of discrete data, and rectification should be performed within that geometry to obtain efficient and reliable categorical generation.

Table 3: Text8 character-level modeling results. ODE-based BPC, reported as mean $\pm$ standard deviation over three random seeds. Bold indicates the best mean, and $*$ indicates the second-best mean.

Method	BPC $\downarrow$ (adaptive ODE)
LinearFM [13]	2.15448 ± 0.00024
SFM [13]	$1.82998 \pm 0.00028^{*}$
R-LFM	3.26687 ± 0.00012
ReDi [11]	2.30562 ± 0.01010
ReSFlow	$\mathbf{1.26146 \pm 0.00036}$

4.5 Promoter Sequence Design

We finally evaluate on conditional promoter DNA sequence design [20], a long-range biological sequence generation task over the nucleotide alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$. This experiment is designed to test whether the proposed manifold-aware rectification remains effective when the data are both discrete and conditionally structured: unlike binarized images, promoter sequences require preserving position-dependent motifs and global consistency with the transcription initiation signal. Thus, this benchmark stresses the same issue discussed in the previous sections, namely whether a learned categorical flow can be compressed to one step without losing the geometry needed for faithful conditional generation. We use signal proxy mean squared error (SP-MSE) as the evaluation metric, where lower is better. Following the standard protocol, generated promoters are passed to a pre-trained Sei predictor, and SP-MSE measures the discrepancy between the predicted activity profiles of generated and real human promoter sequences.

We report results under both accurate Dopri5 ODE simulation and one-step sampling in Table 4. Under ODE sampling, ReSFlow is nearly twice as fast as LinearFM (67.94 vs. 119.04 seconds per 1,000 sequences), while yielding a slightly higher mean SP-MSE (0.028743 vs. 0.027734). More importantly, ReSFlow preserves its performance in the one-step setting: its mean SP-MSE is 0.028718, compared with 0.028743 under ODE sampling, while the sampling time decreases from 67.94 seconds to 0.409 seconds. By comparison, SFM performs competitively with full ODE integration but degrades to 0.035564 under one-step sampling, showing that manifold-aware interpolation alone is insufficient for reliable single-step generation. R-LFM gives substantially worse SP-MSE values, 0.052146 with ODE and 0.060504 with Euler one-step sampling, suggesting that

rectifying trajectories in the ambient Euclidean geometry can be detrimental for structured categorical sequences. These results support the central design of ReSFlow: reflow should be performed under the intrinsic Fisher–Rao geometry so that the resulting trajectories remain suitable for efficient one-step conditional generation.

Table 4: Conditional promoter design results. Mean $\pm$ standard deviation over three random seeds; time is for generating 1,000 sequences of length 1024. Bold indicates the best mean, and $*$ indicates the second-best mean.

Method	ODE (NFE=adaptive)		Euler one-step (NFE=1)	
	SP-MSE ↓	Time ↓	SP-MSE ↓	Time ↓
LinearFM [13]	**0.027734 ± 0.000897**	119.04 ± 16.57	0.029449 ± 0.001673*	0.410 ± 0.004*
SFM [13]	0.028015 ± 0.000257*	82.18 ± 4.26	0.035564 ± 0.007768	0.410 ± 0.005
R-LFM	0.052146 ± 0.000781	**67.88 ± 4.17**	0.060504 ± 0.000351	**0.409 ± 0.001**
ReSFlow	0.028743 ± 0.000755	67.94 ± 0.35*	**0.028718 ± 0.000366**	**0.409 ± 0.003**

5 Conclusion

We proposed Rectified Statistical Flow (ReSFlow), a principled generative framework for discrete data that performs flow matching directly on the probability simplex under its natural spherical geometry. By proposing a reflow procedure in the spherical space via the square-root parameterization, ReSFlow respects the intrinsic Riemannian structure of categorical distributions while admitting closed-form conditional velocity fields. We further established a theoretical guarantee on the straightening rate of iterative spherical rectification, and demonstrated the effectiveness of ReSFlow over existing baselines across diverse discrete generation tasks. As future work, the straightening property of ReSFlow naturally lends itself to single-step generation on other spherical data, such as unit-norm embeddings, directional statistics, and rotation-valued representations.

Acknowledgments and Disclosure of Funding

We thank the anonymous reviewers for their time, thoughtful comments, and constructive suggestions, which helped improve the clarity and presentation of this work.

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62402263 and 62672259), the Major Science and Technology Special Project Program of the Inner Mongolia Autonomous Region (Grant No. 2026ZD0089), and the Open Competition Project under the Science and Technology Breakthrough Initiative of the Inner Mongolia Autonomous Region (Grant No. 2026KJTW0003). This work was also supported by the Inner Mongolia University High-level Talent Project (Grant No. 10000-A260015/322). Renzhe Xu's research was supported by the Shanghai Sailing Program (No. 24YF2711600).

References

[1] Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space. *arXiv preprint arXiv:2307.08698*, 2023.

[2] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *ICLR*, 2023.

[3] Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *ICML*, pages 5453–5512, 2024.

[4] Olga Zaghen, Floor Eijkelboom, Alison Pouplin, Cong Liu, Max Welling, Jan-Willem van de Meent, and Erik J Bekkers. Riemannian variational flow matching for material and protein design. *arXiv preprint arXiv:2502.12981*, 2025.

[5] Guillaume Huguet, James Vuckovic, Kilian Fatras, Eric Thibodeau-Laufer, Pablo Lemos, Riashat Islam, Cheng-Hao Liu, Jarrid Rector-Brooks, Tara Akhound-Sadegh, Michael Bronstein,

et al. Sequence-augmented se (3)-flow matching for conditional protein generation. *NeurIPS*, 37:33007–33036, 2024.

[6] Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data, 2024.

[7] Neta Shaul, Itai Gat, Marton Havasi, Daniel Severo, Anuroop Sriram, Peter Holderrieth, Brian Karrer, Yaron Lipman, and Ricky TQ Chen. Flow matching with general discrete paths: A kinetic-optimal perspective. *ICLR*, 2024.

[8] Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *ICLR*, 2023.

[9] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *NeurIPS*, volume 37, pages 133345–133385, 2024.

[10] Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. In *ICLR*, 2025.

[11] Jaehoon Yoo, Wonjung Kim, and Seunghoon Hong. Redi: Rectified discrete flow. *NeurIPS*, 2025.

[12] Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. *ICML*, 2024.

[13] Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. *NeurIPS*, 37:54787–54819, 2024.

[14] Chaoran Cheng, Yusong Wang, Yuxin Chen, Xiangxin Zhou, Nanning Zheng, and Ge Liu. Riemannian consistency model. *NeurIPS*, 2025.

[15] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2023.

[16] C. Radhakrishna Rao. *Information and the Accuracy Attainable in the Estimation of Statistical Parameters*, pages 235–247. Springer New York, New York, NY, 1992.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017.

[18] John R Dormand and Peter J Prince. A family of embedded runge-kutta formulae. *Journal of computational and applied mathematics*, 6(1):19–26, 1980.

[19] Matt Mahoney. Large text compression benchmark, 2011.

[20] Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *ICML*, pages 1276–1301. PMLR, 2023.

[21] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023.

[22] Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *NeurIPS*, 2022.

[23] Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İ Ceylan, Michael Bronstein, and Avishek J Bose. Fisher flow matching for generative modeling over discrete data. *NeurIPS*, 37:139054–139084, 2024.

[24] Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

[25] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, pages 6613–6623, 2024.

[26] Caglar Gulcehre. Beyond autoregression: Fast llms via self-distillation through time. In *ICLR*. ICLR, 2025.

[27] Satoshi Hayakawa, Yuhta Takida, Masaaki Imaizumi, Hiromi Wakaki, and Yuki Mitsufuji. Distillation of discrete diffusion through dimensional correlations. *arXiv preprint arXiv:2410.08709*, 2024.

[28] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. In *ICLR*, 2025.

[29] John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, 2nd edition, 2013.

A Statistical Manifold of Categorical Distributions and the Spherical Projection

To formulate continuous-time generative dynamics for discrete data, a natural approach is to consider the n-class categorical distributions as points lying on the $(n-1)$-dimensional probability simplex Δ^{n-1}. Geometrically, Δ^{n-1} can be treated as a Riemannian manifold, known as the statistical manifold, where the intrinsic geometry is governed by the Fisher Information Metric.

For a distribution $\mu \in \Delta^{n-1}$ (where $\sum_{i=1}^{n} \mu_i = 1$ and $\mu_i \geq 0$) and tangent vectors u, v at μ, the Fisher inner product is defined as:

$$\langle u, v \rangle_\mu = \sum_{i=1}^{n} \frac{u_i v_i}{\mu_i}. \tag{15}$$

The Boundary Illness. While the Fisher metric elegantly captures the information geometry of probability distributions, Equation (15) reveals a critical flaw for generative modeling: the inner product becomes ill-defined on the boundary of the simplex. As any probability mass $\mu_i \to 0$, the denominator causes the metric to diverge. In discrete generation tasks, the target data points are typically deterministic, represented as one-hot vectors. These one-hot vectors lie exactly on the sharp corners (boundaries) of the simplex. Consequently, performing flow matching or integrating vector fields directly on Δ^{n-1} using the Fisher metric leads to severe numerical instability and singularities near the targets.

Necessity of the Spherical Projection. To circumvent this boundary illness while preserving the intrinsic geometry, we introduce the square-root diffeomorphism:

$$\pi : \Delta^{n-1} \to \mathbb{S}_+^{n-1}, \quad \mu_i \mapsto x_i = \sqrt{\mu_i}. \tag{16}$$

Since $\sum_{i=1}^{n} \mu_i = 1$, it naturally follows that $\sum_{i=1}^{n} x_i^2 = 1$. This transformation smoothly maps the original probability simplex to the positive orthant of the unit sphere $\mathbb{S}_+^{n-1}$. Under this mapping, the intrinsic geometry of the statistical manifold is beautifully preserved up to a constant factor. Specifically, the geodesic distance d_{cat} on the simplex is strictly proportional to the standard great-circle distance d_S on the sphere:

$$d_{\text{cat}}(\mu, \nu) = 2 d_{\mathbb{S}}(\pi(\mu), \pi(\nu)), \tag{17}$$

where $d_{\mathbb{S}}(x, y) = \arccos(\sum_{i=1}^{n} x_i y_i)$ for $x, y \in \mathbb{S}_+^{n-1}$. This indicates that we can seamlessly perform geometric operations on the unit sphere instead of the simplex.

Beyond preserving the distance metric, this transformation fundamentally resolves the boundary illness. For a point $x \in \mathbb{S}_+^{n-1}$, its tangent space is given by $T_x \mathbb{S}_+^{n-1} = \{\tilde{u} \in \mathbb{R}^n \mid \sum_{i=1}^{n} x_i \tilde{u}_i = 0\}$. For any two tangent vectors $\tilde{u}, \tilde{v} \in T_x \mathbb{S}_+^{n-1}$, the inner product on the sphere simplifies to the standard Euclidean dot product:

$$\langle \tilde{u}, \tilde{v} \rangle_x = \sum_{i=1}^{n} \tilde{u}_i \tilde{v}_i. \tag{18}$$

Contrasting Equation (18) with Equation (15) makes the advantage immediately clear: the spherical inner product does not involve the coordinates (x_i or μ_i) in the denominator. Consequently, the metric, vector fields, and geodesics remain perfectly well-defined and numerically stable everywhere. By formulating our continuous dynamics on $\mathbb{S}_+^{n-1}$, our framework elegantly resolves the singularity at the boundary ($x_i = 0$), enabling stable and exact flow matching to deterministic one-hot targets.

B Related Work

Flow matching was originally proposed for continuous domains as an alternative to diffusion, learning transport along a prescribed probability path [21, 8, 15]. Extending this idea to discrete spaces is more challenging, since the dynamics evolve over categorical distributions rather than Euclidean variables. Related work has modeled discrete denoising in continuous time using continuous-time Markov chains (CTMCs), bridging diffusion-style methods and discrete data [22]. Building on this perspective, Discrete Flow Matching [9] extends flow matching to discrete generation, while

subsequent methods further refine the geometry and path design, including Fisher Flow Matching [23] and Dirichlet Flow Matching [12]. Compared with discrete diffusion, which typically learns reverse transitions, these approaches emphasize direct probability transport. Among them, Statistical Flow Matching (SFM) [13] is particularly appealing, as it formulates categorical generation on the simplex as a statistical manifold, better respecting the geometry of categorical distributions while avoiding explicit CTMC jump-rate parameterization. This provides a cleaner and often more stable framework for discrete generation, and thus serves as the foundation of our model.

Flow-based models can generate high-quality discrete samples, but inference is often costly, typically requiring tens to hundreds of function evaluations. Moreover, most one- or few-step generative methods focus on continuous data [15, 24, 14, 25]; in discrete spaces, fast sampling has been explored mainly within diffusion-based frameworks through distillation or specialized acceleration schemes [26, 27, 10, 28]. Flow matching is often considered more suitable for low-step generation because it learns transport dynamics rather than local denoising and allows the design of straighter probability paths, e.g., rectified or OT-inspired flows. Its deterministic ODE formulation may therefore be easier to compress into a few updates, although standard flow matching is not inherently one-step. Recently, ReDi [11] extended rectified flow to discrete spaces via iterative reflow. However, this approach—like many continuous relaxation methods for discrete spaces—heavily requires pre-trained token embeddings to function, restricting its applicability when training purely from scratch. Moreover, the extension is performed entirely in Euclidean space, ignoring the intrinsic Riemannian geometry of the probability simplex. As a result, linear interpolation paths are not true geodesics on the statistical manifold, leading to suboptimal transport costs and trajectories that are inherently harder to straighten through reflow. Consequently, ReDi's practical advantages over standard discrete flow matching remain less clear-cut than those of rectified flow in continuous settings.

C Implementation Details

C.1 Tangent Space Projection

The learned $v_\theta(X_t, t)$ must lie in the tangent space $T_{X_t}\mathbb{S}^{K-1}$ at each point to ensure the flow remains on the manifold. We enforce this via tangent space projection:

$$v_\theta^{\tan}(X_t, t) = v_\theta(X_t, t) - \langle v_\theta(X_t, t), X_t \rangle \, X_t, \tag{19}$$

which removes the normal component of the predicted velocity at each point on the sphere.

C.2 Algorithm Procedure

Algorithm 1 summarizes the complete ReSFlow procedure. For each reflow iteration k, we train a velocity field v_θ on the current coupling (Z_0^k, Z_1^k). At each training step, a mini-batch of B triplets $\{(t_j, z_{0,j}, z_{1,j})\}_{j=1}^B$ is sampled, where $t_j \sim \mathcal{U}[0,1]$ and $(z_{0,j}, z_{1,j})$ are drawn from the current coupling. Geodesic interpolants z_{t_j} and their corresponding tangent velocities u_{t_j} are computed via spherical linear interpolation (slerp) on $\mathbb{S}_+^{K-1}$. The network parameters θ are then updated by minimizing the mini-batch Monte Carlo estimate of the projected velocity matching loss: $\frac{1}{B}\sum_{j=1}^B \|v_\theta(z_{t_j}, t_j) - \langle v_\theta(z_{t_j}, t_j), z_{t_j}\rangle z_{t_j} - u_{t_j}\|^2$. After training converges, the coupling is rectified by simulating the learned ODE forward to produce new pairs (Z_0^{k+1}, Z_1^{k+1}), progressively straightening the flow trajectories. At inference time, a single Euler step via the exponential map yields the final estimate $\hat{p} \in \Delta^{K-1}$.

D Spherical Rectified-Flow Proofs

For a random variable X, $\mathrm{Law}(X)$ denotes its probability distribution. This appendix follows the structure of the rectified-flow analysis [15], but replaces the Euclidean linear interpolation by spherical geodesic interpolation and makes explicit the geometric steps needed on $M = \mathbb{S}^{K-1}$. The first subsection collects the sphere geometry and conditional-expectation identities used below. The second subsection proves the spherical marginal-preservation property. The third subsection shows that spherical rectification reduces convex nondecreasing functions of the geodesic transport distance.

Algorithm 1 Rectified Statistical Flow (ReSFlow)

Require: Prior p_0, target q, diffeomorphism $\pi : \Delta^{K-1} \to \mathbb{S}_+^{K-1}$, reflow iterations N_{reflow}

Initialize: Sample $(X_0, X_1) = (\pi(\mu), \pi(\nu))$ independently, $\mu \sim p_0$, $\nu \sim q$
  Set $(Z_0^0, Z_1^0) = (X_0, X_1)$
for $k = 0, 1, \ldots, N_{\text{reflow}}$ **do**
  // Train: learn velocity field on current coupling
  repeat
    Sample mini-batch $\{(t_j,\, z_{0,j},\, z_{1,j})\}_{j=1}^{B}$, $t_j \sim \mathcal{U}[0,1]$, $(z_{0,j},\, z_{1,j})$ from coupling (Z_0^k, Z_1^k)
    $z_{t_j} \leftarrow \mathrm{slerp}(z_{0,j}, z_{1,j};\, t_j)$, $\theta_j = \arccos(z_{0,j}^\top z_{1,j})$ ▷ Geodesic interpolation on $\mathbb{S}_+^{K-1}$
    $u_{t_j} \leftarrow \frac{\theta_j}{\sin\theta_j}[-\cos((1-t_j)\theta_j)\, z_{0,j} + \cos(t_j\theta_j)\, z_{1,j}]$
    Update θ by minimizing $\frac{1}{B}\sum_{j=1}^{B} \|v_\theta(z_{t_j}, t_j) - \langle v_\theta(z_{t_j}, t_j), z_{t_j}\rangle z_{t_j} - u_{t_j}\|^2$
  until converged
  if $k < N_{\text{reflow}}$ **then**
    // Reflow: recouple via ODE simulation
    $\forall\, Z_0^{k+1} \sim \pi(p_0)$: $Z_1^{k+1} \leftarrow \texttt{ODESolve}(v_{\hat\theta^{(k)}},\, Z_0^{k+1},\, 0 \to 1)$
  end if
end for

One-Step Inference:
Sample $X_0 \sim \pi(p_0)$; $w \leftarrow v_{\hat\theta}(X_0, 0)$
$\hat X_1 \leftarrow \cos(\|w\|)\, X_0 + \sin(\|w\|)\, \frac{w}{\|w\|}$ ▷ Exponential map on $\mathbb{S}^{K-1}$
return $\hat p = \pi^{-1}(\hat X_1) \in \Delta^{K-1}$

The final subsection states the straightening effect and the reflow $O(1/N_{\text{reflow}})$ rate in the same spirit as the original rectified-flow paper, with the corresponding spherical notions of straightness and energy.

D.1 Geometric and Probabilistic Preliminaries

Sphere Geometry. Let $M = \mathbb{S}^{K-1} = \{x \in \mathbb{R}^K : \|x\|_2 = 1\}$. We use the Riemannian metric inherited from $\mathbb{R}^K$. Thus the tangent space and inner product are

$$T_x M = \{u \in \mathbb{R}^K : u^\top x = 0\}, \qquad \langle u, v\rangle_x = u^\top v, \qquad \|u\|_x = \sqrt{u^\top u}.$$

The geodesic distance is

$$d_{\mathbb{S}}(x, y) = \arccos(x^\top y) \in [0, \pi].$$

The cut locus of x is the antipodal point $-x$. When $y \neq -x$, the logarithmic map is single-valued. Writing $\theta = d_{\mathbb{S}}(x, y)$, for $0 < \theta < \pi$,

$$\log_x(y) = \frac{\theta}{\sin\theta}(y - \cos\theta\, x) \in T_x M, \qquad \|\log_x(y)\|_x = \theta. \tag{20}$$

For $y = x$, we set $\log_x(x) = 0$. The exponential map is

$$\exp_x(u) = \cos(\|u\|_x)x + \sin(\|u\|_x)\frac{u}{\|u\|_x}, \qquad u \in T_x M,\ u \neq 0,$$

with $\exp_x(0) = x$.

Lemma D.1 (Constant-speed spherical geodesic)**.** *Let $x, y \in M$ with $y \neq -x$. The curve*

$$\gamma_t = \exp_x(t \log_x(y)), \qquad t \in [0, 1],$$

is the minimizing geodesic from x to y. It has the slerp form

$$\gamma_t = \frac{\sin((1-t)\theta)}{\sin\theta}x + \frac{\sin(t\theta)}{\sin\theta}y, \qquad \theta = d_{\mathbb{S}}(x, y), \tag{21}$$

and it has constant speed

$$\|\dot\gamma_t\|_{\gamma_t} = \theta = d_{\mathbb{S}}(x, y).$$

Proof. For $0 < \theta < \pi$, define

$$e = \frac{y - \cos\theta\, x}{\sin\theta}.$$

Then $e^\top x = 0$, $\|e\| = 1$, and $y = \cos\theta\, x + \sin\theta\, e$. By Equation (20), $\log_x(y) = \theta e$, so

$$\gamma_t = \exp_x(t\theta e) = \cos(t\theta)x + \sin(t\theta)e.$$

Substituting the definition of e gives Equation (21). Differentiating gives $\dot\gamma_t = \theta[-\sin(t\theta)x + \cos(t\theta)e]$, whose norm is θ. The curve stays in the two-dimensional plane spanned by x and y and traces the shorter great-circle arc, hence it is the minimizing geodesic. The case $x = y$ is the constant curve. □

Lemma D.2 (Path length and energy bounds)**.** *If z_t is any absolutely continuous curve on M, then*

$$d_{\mathbb{S}}(z_0, z_1) \le \int_0^1 \|\dot z_t\|_{z_t}\, dt \tag{22}$$

and

$$d_{\mathbb{S}}(z_0, z_1)^2 \le \int_0^1 \|\dot z_t\|_{z_t}^2\, dt. \tag{23}$$

Proof. The geodesic distance is the minimum length over all curves connecting the two endpoints, so it is no larger than the length of z_t, proving Equation (22). Squaring Equation (22) and using Cauchy–Schwarz,

$$\left(\int_0^1 \|\dot z_t\|_{z_t}\, dt\right)^2 \le \left(\int_0^1 1^2 dt\right)\left(\int_0^1 \|\dot z_t\|_{z_t}^2 dt\right),$$

which gives Equation (23). □

Probabilistic Identities. All conditional expectations below are taken after embedding tangent vectors in $\mathbb{R}^K$. This creates no ambiguity because each tangent space is a linear subspace of $\mathbb{R}^K$.

Lemma D.3 (Conditional tangent vectors)**.** *Let X be an M-valued random variable and let U be a random vector with $U \in T_X M$ almost surely and $\mathbb{E}[\|U\|^2] < \infty$. Then*

$$m(X) := \mathbb{E}[U \mid X] \in T_X M \qquad \textit{a.s.}$$

Moreover,

$$\mathbb{E}[\|U\|^2] = \mathbb{E}[\|m(X)\|^2] + \mathbb{E}[\|U - m(X)\|^2].$$

Proof. Since $U \in T_X M$, we have $X^\top U = 0$ a.s. Conditioning on X,

$$X^\top m(X) = \mathbb{E}[X^\top U \mid X] = 0,$$

so $m(X) \in T_X M$. For the decomposition, expand the square:

$$\|U\|^2 = \|m(X)\|^2 + \|U - m(X)\|^2 + 2\langle m(X), U - m(X)\rangle.$$

The expectation of the cross term is zero because $\mathbb{E}[U - m(X) \mid X] = 0$. □

Lemma D.4 (Convexity of radial costs)**.** *If $\phi : [0, \infty) \to \mathbb{R}$ is convex and nondecreasing, then*

$$F(w) = \phi(\|w\|)$$

is convex on every Euclidean vector space.

Proof. For a, b in a Euclidean vector space and $\lambda \in [0, 1]$, the triangle inequality gives

$$\|\lambda a + (1 - \lambda)b\| \le \lambda\|a\| + (1 - \lambda)\|b\|.$$

Because ϕ is nondecreasing and convex,

$$\phi(\|\lambda a + (1 - \lambda)b\|) \le \phi(\lambda\|a\| + (1 - \lambda)\|b\|) \le \lambda\phi(\|a\|) + (1 - \lambda)\phi(\|b\|).$$

□

D.2 Marginal Preservation

Definition D.1 (Expected spherical velocity)**.** *For a path-wise continuously differentiable random process $X = \{X_t : t \in [0,1]\}$ on M, its expected spherical velocity v^X is defined as*

$$v^X(x,t) = \mathbb{E}[\dot{X}_t \mid X_t = x] \in T_x M, \qquad x \in \operatorname{supp}(X_t).$$

For $x \notin \operatorname{supp}(X_t)$, we set $v^X(x,t)$ arbitrarily as a measurable tangent vector. The velocity is tangent by Lemma D.3.

Definition D.2 (Spherical rectifiability)**.** *We call $X = \{X_t : t \in [0,1]\}$* rectifiable *if the velocity field $v^X : M \times [0,1] \to TM$ is continuous in (x,t) and locally Lipschitz in x, uniformly on compact time intervals. Under this assumption, the characteristic ODE*

$$\dot{Z}_t = v^X(Z_t, t), \qquad Z_0 = X_0, \tag{24}$$

admits, by the Cauchy–Lipschitz theorem on the compact manifold M [29, Thm. 9.12], a unique global flow $\Phi : [0,1] \times M \to M,\ (t,x) \mapsto \Phi_t(x)$, which is a C^1 diffeomorphism for each t. We call $Z_t := \Phi_t(X_0)$ the spherical rectified flow *induced by X.*

Theorem D.1 (Spherical marginal preservation)**.** *Assume $X = \{X_t : t \in [0,1]\}$ is rectifiable and Z is its spherical rectified flow. Then*

$$\operatorname{Law}(Z_t) = \operatorname{Law}(X_t), \qquad \forall t \in [0,1].$$

Proof. Let $\mu_t = \operatorname{Law}(X_t)$ and $\nu_t = \operatorname{Law}(Z_t)$. We show $\mu_t = \nu_t$ by verifying that both satisfy the same continuity equation and applying a duality argument against arbitrary test functions $h \in C^\infty(M)$.

Step 1 (ν_t is the pushforward). Since $Z_t = \Phi_t(X_0)$ and $X_0 \sim \mu_0$, the definition of the pushforward measure gives

$$\nu_t = (\Phi_t)_{\#}\mu_0, \qquad t \in [0,1].$$

Step 2 (Weak formulation for μ_t). For any $h \in C^\infty(M)$, path-wise differentiability of X and the tower property of conditional expectation yield

$$\frac{d}{dt}\int_M h\, d\mu_t = \frac{d}{dt}\mathbb{E}[h(X_t)] = \mathbb{E}\Big[\langle \nabla_{\mathbb{S}} h(X_t), \dot{X}_t \rangle_{X_t}\Big] = \int_M \langle \nabla_{\mathbb{S}} h(x), v^X(x,t)\rangle_x \, d\mu_t(x). \tag{25}$$

Step 3 (Weak formulation for $\nu_t = (\Phi_t)_{\#}\mu_0$). Apply the chain rule to $h(\Phi_t(x))$ and integrate against μ_0:

$$\begin{aligned}\frac{d}{dt}\int_M h\, d\nu_t &= \frac{d}{dt}\int_M h(\Phi_t(x))\, d\mu_0(x) = \int_M \langle \nabla_{\mathbb{S}} h(\Phi_t(x)), v^X(\Phi_t(x),t)\rangle_{\Phi_t(x)}\, d\mu_0(x) \\ &= \int_M \langle \nabla_{\mathbb{S}} h(x), v^X(x,t)\rangle_x \, d\nu_t(x).\end{aligned} \tag{26}$$

The interchange of d/dt and the integral is justified by dominated convergence, as Φ is C^1 and M is compact.

Step 4 (Equality via the transition flow). Fix $t \in (0,1]$. To transport the test function backward from time t to $s \in [0,t]$, we define

$$h_s := h \circ \Phi_t \circ \Phi_s^{-1}.$$

Note that $h_s \in C^1(M)$, $h_t = h$, and $h_0 = h \circ \Phi_t$. By definition, $h_s(\Phi_s(x)) = h(\Phi_t(x))$, which is identically constant in s. Differentiating this relation with respect to s using the chain rule gives

$$\partial_s h_s(\Phi_s(x)) + \langle \nabla_{\mathbb{S}} h_s(\Phi_s(x)), \partial_s \Phi_s(x)\rangle_{\Phi_s(x)} = 0.$$

Since $\partial_s \Phi_s(x) = v^X(\Phi_s(x), s)$, evaluating at $y = \Phi_s(x)$ yields the backward transport equation:

$$\partial_s h_s(y) = -\langle \nabla_{\mathbb{S}} h_s(y), v^X(y,s)\rangle_y. \tag{27}$$

Now, consider the difference $F_h(s) := \int_M h_s \, d\mu_s - \int_M h_s \, d\nu_s$. Applying the weak formulations (25) and (26) extended to time-dependent test functions, we obtain:

$$\begin{aligned}\tfrac{d}{ds}\int_M h_s \, d\mu_s &= \int_M \partial_s h_s(x) \, d\mu_s(x) + \int_M \langle \nabla_{\mathbb{S}} h_s(x), v^X(x,s)\rangle_x \, d\mu_s(x) \\ &= \int_M \Big(- \langle \nabla_{\mathbb{S}} h_s(x), v^X(x,s)\rangle_x + \langle \nabla_{\mathbb{S}} h_s(x), v^X(x,s)\rangle_x \Big) \, d\mu_s(x) = 0,\end{aligned}$$

and identically $\frac{d}{ds}\int_M h_s \, d\nu_s = 0$. Thus, $F_h(s)$ is constant for $s \in [0,t]$.

Evaluating at $s = 0$, since $\mu_0 = \nu_0$, we have

$$F_h(0) = \int_M (h \circ \Phi_t) \, d\mu_0 - \int_M (h \circ \Phi_t) \, d\nu_0 = 0.$$

Evaluating at $s = t$, we get

$$F_h(t) = \int_M h \, d\mu_t - \int_M h \, d\nu_t = 0.$$

Since this holds for every $h \in C^\infty(M)$, we conclude $\mu_t = \nu_t$. □

D.3 Spherical Transport-Cost Contraction

Definition D.3 (Spherically rectifiable coupling)**.** *A coupling (X_0, X_1) on M is called spherically rectifiable if $X_0 \neq -X_1$ almost surely and its spherical geodesic interpolation process*

$$X = \big\{ \exp_{X_0}\big(t \log_{X_0}(X_1)\big) : t \in [0,1] \big\}$$

is rectifiable. In this case, the process $Z = \{Z_t : t \in [0,1]\}$ in Equation (24) *is called the spherical rectified flow of the coupling (X_0, X_1), denoted as $Z = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1))$, and (Z_0, Z_1) is called the spherical rectified coupling of (X_0, X_1), denoted as $(Z_0, Z_1) = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1))$.*

The Euclidean rectified-flow theorem controls arbitrary convex costs of the displacement vector $X_1 - X_0$. On a curved manifold there is no globally consistent displacement vector, so the coordinate-free replacement is a convex nondecreasing function of the geodesic distance.

Theorem D.2 (Contraction of convex distance costs)**.** *Let (X_0, X_1) be spherically rectifiable and let $(Z_0, Z_1) = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1))$. For every convex nondecreasing $\phi : [0,\infty) \to \mathbb{R}$,*

$$\mathbb{E}[\phi(d_{\mathbb{S}}(Z_0, Z_1))] \leq \mathbb{E}[\phi(d_{\mathbb{S}}(X_0, X_1))] \, . \tag{28}$$

In particular,

$$\mathbb{E}\big[d_{\mathbb{S}}(Z_0, Z_1)^2\big] \leq \mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big] \, .$$

Proof. For each ODE trajectory, Lemma D.2 gives

$$d_{\mathbb{S}}(Z_0, Z_1) \leq \int_0^1 \|\dot{Z}_t\|_{Z_t} \, dt = \int_0^1 \|v^X(Z_t, t)\|_{Z_t} \, dt.$$

For each fixed t, Lemma D.4 shows that $w \mapsto \phi(\|w\|)$ is convex on the tangent space $T_{X_t} M$. Using $v^X(X_t, t) = \mathbb{E}[\dot{X}_t \mid X_t]$ and Jensen's inequality in the tangent space,

$$\begin{aligned}\phi\big(\|v^X(X_t,t)\|_{X_t}\big) &= \phi\Big(\|\mathbb{E}[\dot{X}_t \mid X_t]\|_{X_t}\Big) \\ &\leq \mathbb{E}\Big[\phi(\|\dot{X}_t\|_{X_t}) \mid X_t\Big] \, .\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}[\phi(d_{\mathbb{S}}(Z_0, Z_1))] &\le \mathbb{E}\Big[\phi\Big(\int_0^1 \|v^X(Z_t,t)\|_{Z_t}\, dt\Big)\Big] && \text{(by Lemma D.2 and monotonicity of } \phi) \\
&\le \int_0^1 \mathbb{E}\big[\phi\big(\|v^X(Z_t,t)\|_{Z_t}\big)\big]\, dt && \text{(by Jensen's inequality in time)} \\
&= \int_0^1 \mathbb{E}\big[\phi\big(\|v^X(X_t,t)\|_{X_t}\big)\big]\, dt && \text{(by Theorem D.1)} \\
&\le \int_0^1 \mathbb{E}\Big[\mathbb{E}\Big[\phi(\|\dot{X}_t\|_{X_t}) \mid X_t\Big]\Big]\, dt && \text{(by conditional Jensen's inequality)} \\
&= \int_0^1 \mathbb{E}\Big[\phi(\|\dot{X}_t\|_{X_t})\Big]\, dt && \text{(by the law of total expectation)} \\
&= \int_0^1 \mathbb{E}[\phi(d_{\mathbb{S}}(X_0, X_1))]\, dt && \text{(by Lemma D.1)} \\
&= \mathbb{E}[\phi(d_{\mathbb{S}}(X_0, X_1))] && \text{(since the interval has length one)}
\end{aligned}$$

which is Equation (28). □

D.4 The Straightening Effect

A spherically rectifiable coupling (X_0, X_1) is called straight if it is a fixed point of spherical rectification, that is, if $(Z_0, Z_1) = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1)) = (X_0, X_1)$.

For a spherically rectifiable coupling (X_0, X_1), let

$$X_t = \exp_{X_0}\big(t \log_{X_0}(X_1)\big), \qquad t \in [0,1],$$

be its spherical geodesic interpolation. Define

$$V_{\mathbb{S}}((X_0, X_1)) = \int_0^1 \mathbb{E}\Big[\|\dot{X}_t - v^X(X_t,t)\|^2_{X_t}\Big]\, dt.$$

For $Z = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1))$, define

$$S_{\mathbb{S}}(Z) = \int_0^1 \mathbb{E}\Big[\|\dot{Z}_t\|^2_{Z_t}\Big]\, dt - \mathbb{E}\big[d_{\mathbb{S}}(Z_0, Z_1)^2\big].$$

Theorem D.3 (Spherical straight couplings)**.** *Assume (X_0, X_1) is spherically rectifiable, let X_t be its spherical geodesic interpolation, and let $Z = \texttt{RectFlow}_{\mathbb{S}}((X_0, X_1))$. Then (X_0, X_1) is straight iff the following equivalent statements hold:*

1. *The squared spherical transport cost is unchanged:*
$$\mathbb{E}\big[d_{\mathbb{S}}(Z_0, Z_1)^2\big] = \mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big].$$
2. *The coupling is a fixed point of spherical rectification:*
$$(Z_0, Z_1) = (X_0, X_1).$$
3. *The rectified flow coincides with the spherical geodesic interpolation: $Z_t = X_t$ almost surely for all $t \in [0,1]$.*
4. *$V_{\mathbb{S}}((X_0, X_1)) = 0$.*

Proof. The implications $3 \Rightarrow 2 \Rightarrow 1$ are immediate. To show $1 \Rightarrow 4$, take $\phi(r) = r^2$ in the proof of Theorem D.2. The same calculation gives

$$\begin{aligned}
\mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big] &= \int_0^1 \mathbb{E}\Big[\|\dot{X}_t\|^2_{X_t}\Big]\, dt \\
&= V_{\mathbb{S}}((X_0, X_1)) + \int_0^1 \mathbb{E}\big[\|v^X(X_t,t)\|^2_{X_t}\big]\, dt \\
&= V_{\mathbb{S}}((X_0, X_1)) + \int_0^1 \mathbb{E}\Big[\|\dot{Z}_t\|^2_{Z_t}\Big]\, dt,
\end{aligned}$$

Here the first equality uses the constant speed of the spherical geodesic in Lemma D.1, the second equality uses the conditional variance decomposition in Lemma D.3, and the last equality uses marginal preservation. Therefore

$$\mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big] - \mathbb{E}\big[d_{\mathbb{S}}(Z_0, Z_1)^2\big] = S_{\mathbb{S}}(Z) + V_{\mathbb{S}}((X_0, X_1)). \tag{29}$$

Both terms on the right are nonnegative: $V_{\mathbb{S}} \geq 0$ by definition, and $S_{\mathbb{S}}(Z) \geq 0$ by Lemma D.2. Therefore equality of the squared costs implies $V_{\mathbb{S}}((X_0, X_1)) = 0$.

It remains to show $4 \Rightarrow 3$. If $V_{\mathbb{S}}((X_0, X_1)) = 0$, then

$$\dot{X}_t = v^X(X_t, t) \qquad \text{a.s. for a.e. } t.$$

By Fubini's theorem, after removing a single probability-zero set of sample paths, this equality holds for Lebesgue-a.e. t along each remaining path. Since the paths are path-wise continuously differentiable, the fundamental theorem of calculus in the ambient Euclidean space gives, for every $t \in [0, 1]$,

$$X_t = X_0 + \int_0^t v^X(X_s, s)\, ds.$$

Thus X satisfies the same characteristic equation as Z with the same initial condition. By the uniqueness assumed in spherical rectifiability, $X = Z$. □

Theorem D.4 (Spherical reflow straightening rate)**.** *Let (X_0, X_1) be a coupling on M, let $N_{\textit{reflow}} \geq 1$ be an integer, and set $(Z_0^0, Z_1^0) = (X_0, X_1)$. Given (Z_0^k, Z_1^k), let*

$$Z^{k+1} = \texttt{RectFlow}_{\mathbb{S}}((Z_0^k, Z_1^k)),$$

and write (Z_0^{k+1}, Z_1^{k+1}) for its endpoint coupling. Assume each (Z_0^k, Z_1^k) is spherically rectifiable for $k = 0, \ldots, N_{\textit{reflow}} - 1$. Then

$$\sum_{k=0}^{N_{\textit{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] \leq \mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big]. \tag{30}$$

Consequently, if $\mathbb{E}[d_{\mathbb{S}}(X_0, X_1)^2] < +\infty$, then

$$\min_{0 \leq k \leq N_{\textit{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] = O(1/N_{\textit{reflow}}).$$

Proof. Apply Equation (29) to each rectification step:

$$\mathbb{E}\big[d_{\mathbb{S}}(Z_0^k, Z_1^k)^2\big] - \mathbb{E}\big[d_{\mathbb{S}}(Z_0^{k+1}, Z_1^{k+1})^2\big] = S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k)).$$

Summing over $k = 0, \ldots, N_{\text{reflow}} - 1$ telescopes and yields

$$\sum_{k=0}^{N_{\text{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] = \mathbb{E}\big[d_{\mathbb{S}}(Z_0^0, Z_1^0)^2\big] - \mathbb{E}\Big[d_{\mathbb{S}}(Z_0^{N_{\text{reflow}}}, Z_1^{N_{\text{reflow}}})^2\Big].$$

Since the second term on the right is nonnegative and $(Z_0^0, Z_1^0) = (X_0, X_1)$, Equation (30) follows. The $O(1/N_{\text{reflow}})$ statement follows from the explicit bound

$$\min_{0 \leq k \leq N_{\text{reflow}}-1} \big[S_{\mathbb{S}}(Z^{k+1}) + V_{\mathbb{S}}((Z_0^k, Z_1^k))\big] \leq \frac{\mathbb{E}\big[d_{\mathbb{S}}(X_0, X_1)^2\big]}{N_{\text{reflow}}}.$$

□

E Experimental Setup

In this appendix, we provide dataset-specific implementation details for all benchmarks. Since the network architectures and training protocols differ across datasets, we describe the data processing, model architecture, and optimization setup separately for each benchmark. Unless otherwise specified, all rectified variants use a single reflow iteration, and all reflow models are initialized from the previous checkpoint and fine-tuned rather than trained from scratch. We summarize sampling procedures and evaluation metrics at the end of this section.

E.1 Swiss Roll on the Simplex

Dataset and preprocessing. We follow the Swiss-roll-on-simplex construction used in prior categorical flow-matching work. The data lie on the 2-simplex with $K = 3$ categories, i.e., each sample is a probability vector $\mu \in \Delta^2$. We generate 1000 samples and use the resulting fixed empirical distribution as the target distribution q for this toy experiment. Since this benchmark is used as a geometric sanity check, we do not use a separate validation or test split. Reported likelihoods are evaluated on the same fixed empirical distribution and should be interpreted as relative comparisons among methods rather than as generalization metrics.

The construction maps a two-dimensional Swiss roll to the simplex as follows. Let x and y denote the two Swiss-roll coordinates. We first shift each coordinate by subtracting its batchwise minimum and adding a padding constant $\mathrm{pad} = 1.0$. We then define

$$M = \max_i (x_i + y_i) + \mathrm{pad}, \qquad z_i = M - (x_i + y_i),$$

and obtain the simplex point by normalizing

$$\mu_i = \frac{(x_i, y_i, z_i)}{M}.$$

This construction ensures that all coordinates are nonnegative and sum to one. There is no additional dataset-level ϵ-smoothing parameter; numerical stabilization inside the flow model uses $\epsilon = 10^{-4}$ where needed.

The prior distribution p_0 is sampled on the simplex by drawing independent exponential random variables, normalizing them to sum to one, applying a small clamp for numerical stability, and renormalizing. The target distribution q is the empirical distribution over the 1000 generated Swiss-roll samples. Unless otherwise stated, we use random seed 42 for data and model initialization, and seed 0 for generating reflow pairs.

Model architecture. For this benchmark, all methods use the same MLP vector-field backbone. The input has per-position dimension 3 and tensor shape $(B, 1, 3)$, and the network outputs a vector field with the same per-position dimension. The hidden dimension is set to 128.

The encoder consists of an input branch, a time branch, and a fusion head. The input branch maps the current state through two linear layers with ReLU activation. The time branch applies two linear layers to a sinusoidal time-step embedding. The time embedding dimension is 128. The resulting state and time features are concatenated and passed through a fusion MLP with three linear layers and ReLU activations to produce the final vector-field prediction. We do not use dropout or normalization layers in this toy architecture.

All compared methods use this same MLP backbone in the Swiss-roll experiment. The methods differ only in their geometry and flow construction. LinearFM and R-LFM use Euclidean interpolation and projection on the simplex, whereas SFM and ReSFlow use the square-root map and spherical geometry on $\mathbb{S}_+^{K-1}$. Tangent-space projection is applied as described in Appendix C.1.

Training protocol. For the initial flow-matching model, we train with Adam using learning rate 10^{-3} and no weight decay. Since the dataset contains only 1000 points, we use full-batch training with batch size 1000. Each model is trained for 2000 epochs, corresponding to one parameter update per epoch. We do not use learning-rate scheduling, warmup, gradient clipping, or exponential moving average for this toy experiment.

For the reflow variants, we use one reflow iteration. In both R-LFM and ReSFlow, the reflow model is initialized from the previous checkpoint and then fine-tuned, rather than trained from scratch. At the beginning of the reflow stage, we generate $10{,}000$ recoupled endpoint pairs in batches of 256 using ODE sampling from $t = 0$ to $t = 1$ with solver tolerances $\mathrm{atol} = 10^{-5}$ and $\mathrm{rtol} = 10^{-5}$. The resulting model is trained for 2000 epochs using mini-batches of 256 recoupled pairs. The two methods differ only in the geometry used for reflow: R-LFM performs Euclidean/straight-line reflow in the simplex geometry, whereas ReSFlow uses spherical tangent reflow under the square-root map.

E.2 Binarized MNIST

Dataset and preprocessing. We use the static binarized MNIST dataset with the standard $50{,}000/10{,}000/10{,}000$ train/validation/test split. Each image has spatial resolution 28×28. The original data are represented as 784 binary variables, and we treat each pixel as a categorical variable with $K = 2$ categories. Concretely, each binary pixel value $x \in \{0, 1\}$ is represented by the two-dimensional simplex coordinate $(x, 1 - x)$. Thus, the model input is arranged as a tensor of shape $(28, 28, 2)$.

The prior distribution p_0 is sampled independently at each pixel location by drawing exponential random variables for the two categories, normalizing them onto the simplex, applying a small clamp for numerical stability, and renormalizing. The data are not smoothed or dequantized. Unless otherwise specified, numerical stabilization inside the flow model uses $\epsilon = 10^{-4}$.

No additional image preprocessing is applied. In the convolutional vector-field network, the simplex-valued input is internally rescaled from $[0, 1]$ to $[-1, 1]$ before the first convolutional layer.

Model architecture. For binarized MNIST, we use a convolutional vector-field predictor. The input and output tensors both have shape $(B, 28, 28, 2)$, where the last dimension corresponds to the two categorical states of each pixel. Internally, the categorical dimension is treated as the channel dimension for convolutional processing.

All methods compared on binarized MNIST use the same convolutional backbone unless otherwise stated. The differences between methods are in the geometric parameterization and the reflow procedure. LinearFM and R-LFM use Euclidean/simplex geometry, whereas SFM and ReSFlow use the square-root map and spherical geometry on $\mathbb{S}_+^{K-1}$. The shared backbone uses four residual stages, four top-down refinement blocks, ELU activations, InstanceNorm++ normalization, and additive sinusoidal time embeddings with embedding dimension 128.

Training protocol. For the base model, we train with Adam using learning rate 3×10^{-4}, weight decay 0, and batch size 256. Models are trained for $100{,}000$ optimization steps. We use a reduce-on-plateau learning-rate schedule with decay factor 0.5, patience 10, and minimum learning rate 10^{-4}. No warmup is used. We clip the global gradient norm to 100. No exponential moving average is used. Unless otherwise specified, the random seed is set to 42.

For rectified models, we generate recoupled endpoint pairs from the previously trained model by ODE sampling from $t = 0$ to $t = 1$ with solver tolerances $\mathrm{atol} = 10^{-5}$ and $\mathrm{rtol} = 10^{-5}$. The rectified model is initialized from the previous checkpoint and fine-tuned rather than trained from scratch. For each method, we generate $50{,}000$ recoupled pairs in batches of 256 and train the rectified model for $100{,}000$ optimization steps with batch size 256 and learning rate 10^{-4}. ReSFlow uses spherical tangent reflow, whereas R-LFM uses Euclidean/simplex reflow. Experiments were run on an NVIDIA GeForce RTX 5090 GPU with CUDA 13.0 and PyTorch 2.7.0.

E.3 Text8

Dataset and preprocessing. We use the Text8 character-level language modeling benchmark. We follow the standard character-level Text8 split: the first 90M characters are used for training, the next 5M for validation, and the final 5M for testing. The vocabulary consists of the 26 lowercase English letters and the space character, resulting in 27 symbols. Each character is mapped to an integer token.

We use a sequence length of 256. Each training example is a contiguous window of 256 characters. Tokens are represented as one-hot vectors, so each sample has shape $(256, 27)$, and a mini-batch has shape $(B, 256, 27)$. No additional text normalization, smoothing, or dequantization is applied beyond the character-level encoding.

The prior distribution p_0 is sampled independently at each sequence position by drawing exponential random variables over the 27 categories, normalizing them onto the simplex, applying a small clamp for numerical stability, and renormalizing. Unless otherwise specified, numerical stabilization inside the flow model uses $\epsilon = 10^{-4}$.

Model architecture. For Text8, all methods using the DiT backbone share the same architecture unless otherwise specified. The vector-field predictor operates on sequences of length 256 with

vocabulary size 27, using 12 transformer blocks with hidden dimension 768, 12 attention heads, MLP expansion ratio 4, GELU feed-forward activations, SiLU time-conditioning activations, dropout 0.1, layer normalization, and rotary positional embeddings. Time is encoded by a sinusoidal timestep embedding of size 256 followed by a two-layer MLP producing a 128-dimensional conditioning vector, which is injected into each block through adaptive layer normalization via shift, scale, and gate parameters. The final layer outputs a 27-dimensional vector-field prediction at each sequence position.

Training protocol. In the training stage, the base DiT model is trained with AdamW using learning rate 3×10^{-4}, weight decay 0.1, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. We train for 750,000 optimization steps with global batch size 512, using linear warmup for the first 1000 steps followed by a reduce-on-plateau learning-rate schedule with minimum learning rate 10^{-5}. The global gradient norm is clipped to 1.0.

In the reflow stage, recoupled endpoint pairs are generated from the previous model by ODE sampling from $t = 0$ to $t = 1$ with tolerances $\mathrm{atol} = 10^{-5}$ and $\mathrm{rtol} = 10^{-5}$. The reflow model is initialized from the previous checkpoint and fine-tuned rather than trained from scratch. We generate 100,000 recoupled pairs in batches of 256 and train the reflow model for 40,000 optimization steps with batch size 256 and learning rate 4×10^{-5}. ReSFlow uses the spherical parameterization induced by the square-root map, while R-LFM uses the Euclidean reflow parameterization. Training was performed on 4 NVIDIA A800 GPUs and took approximately 6 days.

E.4 Promoter Sequence Design

Dataset and preprocessing. We use the promoter sequence design dataset from the official Zenodo data package[2], which includes the GRCh38 reference genome, FANTOM transcription start site (TSS) annotations, CAGE signal tracks, blacklist regions, and the pre-trained Sei model for downstream evaluation. Each example is a 1024 bp DNA sequence over the alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$, represented as a one-hot tensor in $\mathbb{R}^{1024\times 4}$; ambiguous bases are encoded as the uniform vector $(1/4, 1/4, 1/4, 1/4)$. The data split is chromosome-based: after taking the first 100,000 FANTOM TSS entries, chromosomes chr8 and chr9 are used for testing, chr10 for validation, and all remaining chromosomes for training.

The conditioning signal is the FANTOM CAGE signal aligned to the same 1024 bp window as the DNA sequence. We use the plus-strand CAGE channel as a per-position scalar condition, giving a conditioning tensor in $\mathbb{R}^{1024\times 1}$. During training, the TSS position is randomly shifted by an offset sampled from $\{-10, \ldots, 10\}$ before extracting the sequence and signal window; no offset is used for validation or testing. The flow prior p_0 is the categorical simplex prior, sampled independently at each sequence position and normalized to match the 1024×4 sequence representation, while the target distribution is the empirical distribution of one-hot promoter sequences paired with their CAGE conditions.

Model architecture. The vector-field predictor for promoter design is a one-dimensional dilated convolutional residual network. It takes the current categorical state $x_t \in \mathbb{R}^{1024\times 4}$ and the CAGE condition $c \in \mathbb{R}^{1024\times 1}$, concatenates them into a 5-channel input, and maps them to 256 hidden channels with an initial kernel-size-9 convolution. The backbone contains 20 residual convolutional blocks with 256 channels, kernel size 9, GroupNorm with one group, Swish activations, and additive time conditioning. The dilation pattern is $(1, 1, 4, 16, 64)$ repeated four times. Time is embedded with Gaussian Fourier features followed by a linear layer and SiLU activation, producing a 256-dimensional embedding that is added to each block after projection. The CAGE signal is not separately encoded, but is injected only by input concatenation. A final 1×1 convolutional head with GELU activation maps $256 \to 256 \to 4$, yielding a four-channel vector-field prediction at each sequence position. No dropout is used. All promoter experiments use this same backbone. The differences between the base and reflow models come from the flow geometry and reflow procedure rather than from the sequence encoder.

Training protocol. For the base promoter model, we train with Adam using learning rate 5×10^{-4}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay. The batch size is 256, and training runs for 200,000

[2] `https://doi.org/10.5281/zenodo.7943307`

optimization steps with gradient clipping at global norm 100 and random seed 42. We do not use learning-rate warmup or exponential moving average. Validation is performed every $5{,}000$ steps using the corresponding flow-matching objective on the validation split, and checkpoints are saved periodically.

For ReSFlow, we perform a single reflow stage on top of the base model. The reflow model is initialized from the base checkpoint and fine-tuned rather than trained from scratch. We generate $100{,}000$ recoupled endpoint pairs by ODE sampling from prior samples, integrating the learned vector field from $t = 0$ to $t = 1$ with batch size 128 and tolerances $\mathrm{atol} = \mathrm{rtol} = 10^{-5}$. During reflow pair generation, the CAGE condition c is kept fixed: for each conditioned prior sample (x_0, c), we integrate the conditional vector field to obtain the paired endpoint x_1. The resulting recoupled dataset therefore consists of triples (x_0, x_1, c). The reflow stage is trained for $100{,}000$ optimization steps with batch size 128 and learning rate 10^{-4}.

E.5 Evaluation Metrics, NFE, and Sampling Protocols

We describe the evaluation metrics and sampling protocols used in the main experiments. Since our model represents categorical variables as points on the probability simplex and performs flow matching after the square-root embedding, we distinguish likelihood-based evaluation, sample-quality evaluation, and task-specific downstream evaluation.

Negative log-likelihood. For the Swiss-roll experiment, we report negative log-likelihood (NLL) using the continuous-time change-of-variables formula associated with the learned ODE. This evaluation treats the data as samples from a continuous density on the simplex, or equivalently on the positive orthant of the sphere after the square-root map. Therefore, unlike discrete cross-entropy, the resulting NLL is a differential likelihood quantity and is not constrained to be non-negative. Accurate NLL estimation requires numerical integration of the learned dynamics, and we therefore evaluate NLL only with adaptive ODE integration rather than with one-step sampling.

Fréchet Inception Distance. For binarized MNIST, we report Fréchet Inception Distance (FID) as a sample-quality metric. Generated binary images are compared with the reference data distribution using the standard FID protocol; lower values indicate that the generated samples better match the real data distribution. We report FID both under adaptive ODE sampling and under one-step sampling. This allows us to separate the quality of the learned continuous-time flow from the quality of the compressed one-step sampler.

Bits per character. For Text8, we report bits per character (BPC). Since ReSFlow and the compared flow-matching baselines are not autoregressive language models, we do not evaluate BPC by multiplying token-wise autoregressive conditionals. Instead, we use a variational integral objective that measures how well the learned vector field predicts the terminal categorical state from an intermediate point along the probability path.

Let $\mu_1 \sim q$ be a data sequence and $\mu_0 \sim p_0$ be a prior sample. We map them to the sphere as $X_0 = \pi(\mu_0)$ and $X_1 = \pi(\mu_1)$, and define the intermediate point by spherical geodesic interpolation

$$X_t = \mathrm{slerp}(X_0, X_1; t), \qquad \mu_t = \pi^{-1}(X_t).$$

Given X_t, the model predicts the endpoint by extrapolating the learned velocity field from time t to time 1 on the sphere:

$$\widehat{X}_1(X_t, t) = \mathrm{Exp}_{X_t}\left((1-t)\, v_\theta(X_t, t)\right), \qquad \widehat{\mu}_1(\mu_t, t) = \pi^{-1}\Big(\widehat{X}_1(X_t, t)\Big).$$

For a sequence of length L, the variational negative log-likelihood is computed as

$$\mathcal{L}_{\mathrm{var}} = -\mathbb{E}_{\mu_0 \sim p_0,\, \mu_1 \sim q}\left[\int_0^1 \frac{1}{1-t} \sum_{\ell=1}^{L} \log \langle \widehat{\mu}_{1,\ell}(\mu_t, t), \mu_{1,\ell} \rangle \, dt\right],$$

where $\langle \cdot, \cdot \rangle$ denotes the standard Euclidean inner product over the categorical alphabet at each sequence position. Since $\mu_{1,\ell}$ is one-hot for Text8, the inner product selects the probability assigned to the ground-truth character at position ℓ.

The factor $1/(1-t)$ makes direct numerical integration near $t = 1$ unstable. Following the standard variational evaluation used for continuous-time categorical flows, we apply the change of variables $s = -\log(1-t)$, or equivalently $t = 1 - e^{-s}$. This gives

$$\mathcal{L}_{\text{var}} = -\mathbb{E}_{\mu_0 \sim p_0,\, \mu_1 \sim q}\left[\int_0^{\infty} \sum_{\ell=1}^{L} \log \langle \widehat{\mu}_{1,\ell}(\mu_t, t), \mu_{1,\ell} \rangle \, ds\right], \qquad t = 1 - e^{-s}.$$

In practice, the integrand becomes negligible for large s because t is then very close to 1. We truncate the integral at a finite upper limit and evaluate it numerically with an adaptive solver. The reported BPC is obtained by normalizing the variational negative log-likelihood by the sequence length and converting from nats to bits:

$$\text{BPC} = \frac{\mathcal{L}_{\text{var}}}{L \log 2}.$$

This metric is likelihood-based and therefore is reported only for ODE-based evaluation, not for one-step sample generation.

Signal proxy mean squared error. For conditional promoter sequence design, we use signal proxy mean squared error (SP-MSE). Each generated promoter sequence is evaluated by a pre-trained sequence-to-function predictor, and the predicted promoter activity is compared with that of real human promoter sequences on the test chromosomes. Denoting by r_i the predicted activity profile of the real promoter and by $\widehat{r}_i$ the predicted activity profile of the generated promoter under the same conditioning signal, SP-MSE is computed as

$$\text{SP-MSE} = \frac{1}{N} \sum_{i=1}^{N} \|\widehat{r}_i - r_i\|_2^2 .$$

Lower SP-MSE indicates that the generated DNA sequences better preserve the desired promoter activity implied by the conditioning signal. Unlike FID or NLL, this is a downstream functional metric: it evaluates whether generated sequences are consistent with the biological signal rather than only measuring distributional similarity in sequence space.

Number of function evaluations. We use NFE to denote the number of evaluations of the learned vector field during sampling. Adaptive Dormand–Prince sampling uses a variable number of function evaluations determined by the ODE solver tolerance. In contrast, one-step generation uses NFE$= 1$. For ReSFlow, one-step sampling evaluates the velocity field once at $t = 0$ and maps the result to the endpoint using the spherical exponential map:

$$\widehat{X}_1 = \text{Exp}_{X_0}(v_\theta(X_0, 0)), \qquad \widehat{\mu}_1 = \pi^{-1}(\widehat{X}_1).$$

The final discrete sample is obtained from $\widehat{\mu}_1$ by sampling from, or taking the maximum-probability category of, the predicted categorical distribution.

Wall-clock sampling time. We report wall-clock sampling time for binarized MNIST and promoter sequence design to quantify practical inference efficiency. Timing is measured for generating $1{,}000$ samples and includes the sampling procedure only; it does not include training, checkpoint loading, or metric computation. For promoter design, the reported time corresponds to generating $1{,}000$ sequences of length 1024. We do not report Text8 sampling time in the main table because the reported BPC is computed from the variational likelihood integral rather than from generated samples.

F Additional Results

F.1 FID versus NFE on Binarized MNIST

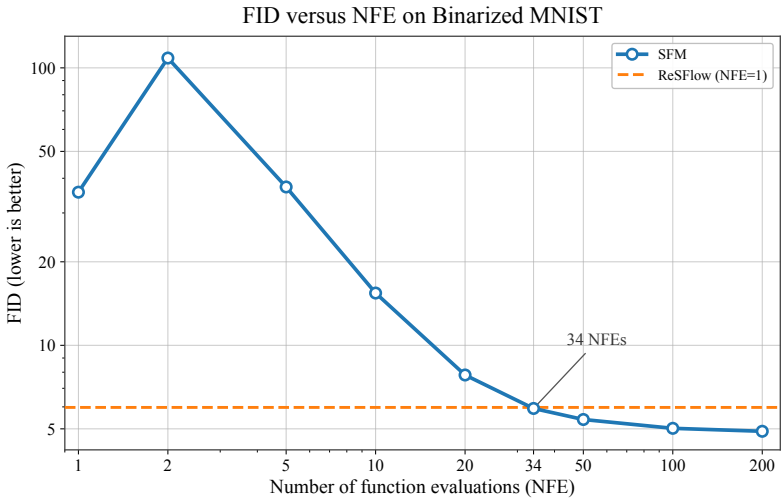


Figure 3: Binarized MNIST FID versus NFE using the matched spherical exponential-map Euler sampler, three seeds, and 1,000 generated samples per seed. The solid curve shows the SFM mean; the dashed line marks the one-step ReSFlow mean (5.973). SFM first has a lower mean at NFE$=34$.

F.2 Fisher–Rao Transport Cost Analysis of Reflow Couplings

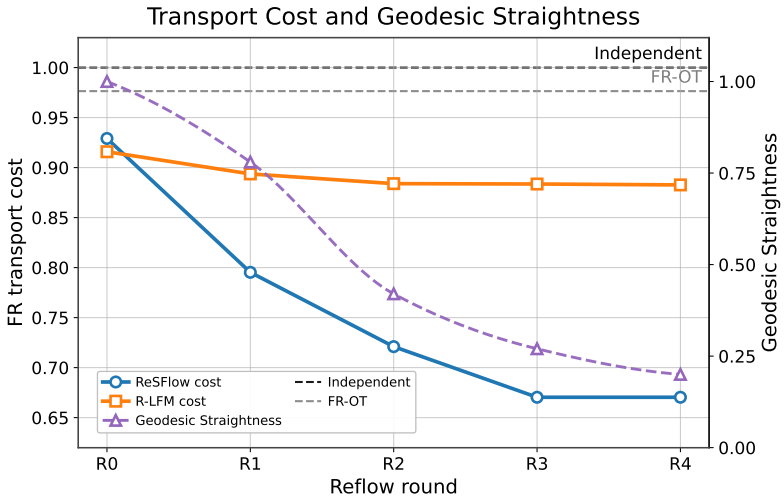


Figure 4: Fisher–Rao transport cost and Geodesic Straightness on BMNIST.

To further examine why spherical reflow improves one-step generation, we analyze the geometry of the couplings produced during reflow on BMNIST. Since each pixel is represented as a categorical distribution on the probability simplex, Euclidean distance is not the appropriate notion of discrepancy. We therefore measure whether reflow produces shorter and more geodesically consistent couplings under the intrinsic Fisher–Rao geometry. This analysis complements the previous sampling results by directly evaluating whether rectification simplifies the transport problem on the simplex.

Following the Fisher–Rao distance defined earlier, for two categorical distributions $\mu, \nu \in \Delta^{C-1}$, we denote this distance by

$$d_{\text{cat}}(\mu, \nu) = 2 \arccos\left(\sum_{c=1}^{C} \sqrt{\mu_c \nu_c}\right). \tag{31}$$

For a full image with D pixels, the corresponding Fisher–Rao transport cost between a pair (z_0, z_1) is defined as

$$\mathcal{C}_{\text{FR}}(z_0, z_1) = \sum_{j=1}^{D} d_{\text{cat}}(z_{0,j}, z_{1,j}), \tag{32}$$

where $z_{0,j}$ and $z_{1,j}$ denote the categorical distributions at pixel j. We compare the reflow couplings against two baselines shown in Fig. 4: an independent random coupling, where prior and data samples are paired independently, and a mini-batch Fisher–Rao optimal transport coupling (FR-OT), where pairings are optimized within each mini-batch using the same Fisher–Rao cost. We also report the costs of the couplings produced by ReSFlow and by the Euclidean reflow variant R-LFM.

As shown in Fig. 4, the independent coupling has the largest transport cost, indicating that arbitrary pairings between prior samples and data samples are geometrically inefficient. FR-OT reduces this cost by explicitly selecting lower-cost assignments within each mini-batch, but the improvement is limited, suggesting that mini-batch optimal transport only provides a locally optimized coupling and cannot substantially simplify the global transport structure. Notably, across reflow rounds, ReSFlow substantially reduces the Fisher–Rao transport cost, whereas R-LFM only achieves a mild reduction and quickly plateaus. This indicates that the improvement of ReSFlow comes from rectifying the coupling in the square-root hyperspherical representation induced by the Fisher–Rao metric, where categorical distributions are transported along intrinsic spherical directions. However, Euclidean reflow updates the coupling in a geometry that is not aligned with the Fisher–Rao structure, leading to much weaker reduction in intrinsic transport cost. These results support our central claim that hyperspherical flow modeling and reflow rectification are complementary for simplifying transport on the probability simplex, empirically corroborating the spherical transport-cost contraction in Theorem 3.2.

F.3 Riemannian Geodesic Straightness of Learned Trajectories

We further evaluate a complementary trajectory-level property: *whether the actual ODE trajectories induced by the learned vector field become closer to geodesics after reflow*. To this end, we design a Riemannian analogue of the straightness metric:

Metric design. In Euclidean Rectified Flow, trajectory straightness is measured by comparing the learned velocity along an ODE trajectory with the displacement between the trajectory's initial and terminal points. On a curved statistical manifold, however, the Euclidean displacement vector is not intrinsic. Since ReSFlow operates on the hypersphere $\mathbb{S}_+^{K-1}$ through the square-root map π, the natural analogue of a straight line is a spherical geodesic, and the natural analogue of a displacement vector is the Riemannian logarithmic map.

For the k-th reflow checkpoint, let $v_{\hat{\theta}}^{(k)}$, denote the learned velocity field. Starting from an initial point $Z_0^{(k)} \sim \pi(p_0)$, we generate the actual model trajectory by integrating

$$\frac{dZ_t^{(k)}}{dt} = v_{\hat{\theta}}^{(k)}(Z_t^{(k)}, t), \qquad t \in [0, 1]. \tag{33}$$

We denote the terminal point of this integrated trajectory by $Z_1^{(k)}$. Importantly, $Z_1^{(k)}$ is the endpoint produced by the learned ODE itself, rather than an externally specified data sample. Therefore, the metric evaluates the intrinsic straightness of the model trajectory.

At an intermediate point $Z_t^{(k)}$, the tangent vector of the constant-speed geodesic that reaches the same endpoint $Z_1^{(k)}$ in the remaining time $1 - t$ is

$$u_{\text{geo}}^{(k)}(Z_t^{(k)}, t) = \frac{\text{Log}_{Z_t^{(k)}}\left(Z_1^{(k)}\right)}{1 - t}. \tag{34}$$

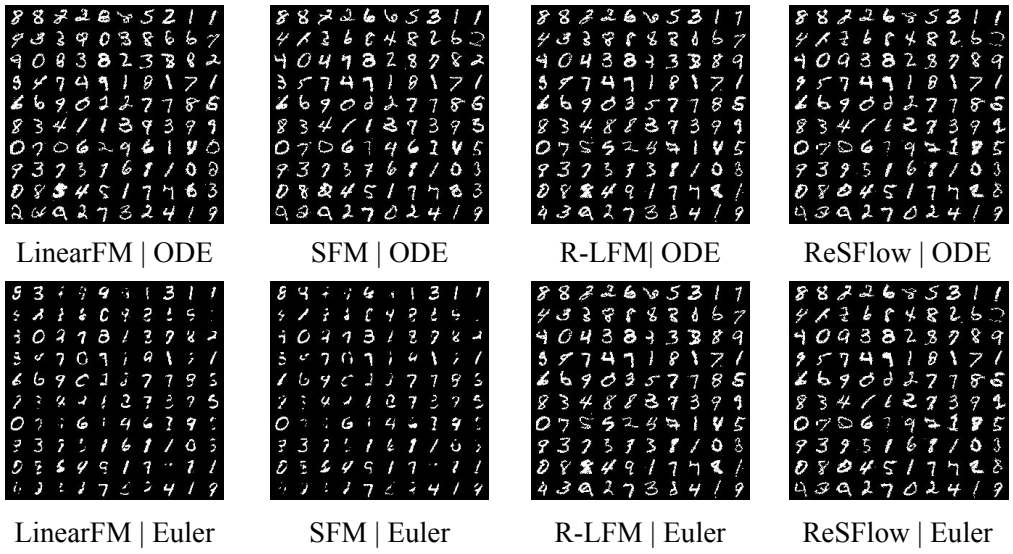


Figure 5: **Generated samples on binarized MNIST.** We compare LinearFM, SFM, R-LFM, and ReSFlow under ODE and one-step Euler sampling. Non-rectified methods degrade under one-step sampling, whereas rectified methods preserve digit structure more effectively. ReSFlow gives high-quality one-step samples by rectifying trajectories under the Fisher–Rao geometry.

We define the Riemannian geodesic straightness of the k-th checkpoint as

$$S_{\mathrm{Riem}}\left(Z^{(k)}\right) = \int_0^1 \mathbb{E}_{Z_0^{(k)}\sim\pi(p_0)}\left[\left\| v_{\hat{\theta}}^{(k)}(Z_t^{(k)},t) - \frac{\mathrm{Log}_{Z_t^{(k)}}(Z_1^{(k)})}{1-t}\right\|_g^2\right] dt. \tag{35}$$

Equivalently, this metric replaces the Euclidean displacement used in the original straightness measure by the instantaneous tangent vector required to move from the current point to the trajectory endpoint along a geodesic. If the learned trajectory itself is a constant-speed geodesic, then $S_{\mathrm{Riem}}(Z^{(k)}) = 0$. Hence, a smaller value indicates that the learned ODE path is more geodesic and more suitable for one-step generation via the exponential map.

Results on binarized MNIST. We evaluate the Riemannian geodesic straightness on binarized MNIST across successive reflow rounds, as reported in Figure 4. The straightness metric decreases monotonically as the number of reflow rounds increases. This indicates that reflow progressively straightens the learned trajectories on the statistical manifold. The decreasing trend directly supports the geometric interpretation of ReSFlow in the main text and empirically validates the straightening-rate guarantee in Theorem 3.3: reflow makes the learned ODE trajectories increasingly aligned with Fisher–Rao geodesics, thereby explaining why a single exponential-map update can achieve strong one-step generation performance.

F.4 Generated Samples

We provide additional qualitative samples for binarized MNIST, Text8, and conditional promoter sequence design in Fig. 5, Table 5, and Table 6, respectively. These examples complement the quantitative results by comparing ODE sampling with one-step Euler sampling.

For binarized MNIST, Fig. 5 shows samples generated by LinearFM, SFM, R-LFM, and ReSFlow. Non-rectified methods produce reasonable ODE samples but degrade noticeably under one-step sampling, while rectified methods better preserve digit structure. ReSFlow gives clear one-step samples, supporting the benefit of performing rectification under the Fisher–Rao geometry.

For Text8, Table 5 reports representative generated character sequences from each model and sampler. The one-step samples highlight whether the learned flow is sufficiently straight for fast

Table 5: **Generated Text8 samples.** We show representative character-level samples generated by each model under ODE and one-step Euler sampling. Underscores denote spaces in the decoded text.

Model	Sampler	Generated text
LinearFM	ODE	calls_vincenti_or_their_burnt_mochery_a_distinct_writing_artiast_is_also_known_as_the_yodioi_cascoliage_it_is_a_crutter_with_bianaple_it_is_known_as_a_while_green_hole_and_even_when_using_plhuhu_snow_and_wouldistay_is_one_of_several_ancient_nuts_of_early_w
LinearFM	Euler	_____e_____ _____e_ __e_____e_____ _____
SFM	ODE	tters_improvements_a_primary_work_to_most_word_genetic_cinematics_who_also_chose_a_word_for_genetic_generation_concurrence_with_a_pt_typical_sporting_the_world_is_often_the_most_for_compatisans_works_such_as_sorcery_and_the_dark_chasen_movements_the_player
SFM	Euler	__izg_i_ee_e_e__fe_____t_____g_t__e__a_ote_____e____e_____e___d_ _a__i_____t__q__t__tebe_s_____e____ee___s__d_i_oe_i__s___ee__ene___o_e__tei_ _eoe___e_n_____e_____a_ea__a__yer_ie__ei_teein__s_n_____e____o___iu___n_ _eo___f__ag_____ho__
R-LFM	ODE	_ners_vincento_wrata_orebry_n_mother_held_at_introducing_artiary_tale_two_beek_in_the_change_casckling_and_is_spyr_operating_fiama_of_rownspart_s_symbols_large_numbers_and_even_opecause_of_his_personalny_would_state_there_in_several_day_baptists_of_eyric_w
R-LFM	Euler	ieto__uaibe_and_the_early_ver_ion_an__a_aeementary_tounterlaei_eoond_the_forst_medonies_ere_ted_from_greatly_attended_from_the_naen_so_condonati_n_tiliter__head_tittre_ont_weec_sters_tore_tad_speecres_and_which_table_to_remume_expending_that_the_leies_anni
ReSFlow	ODE	another_role_ranged_in_from_one_eight_two_one_le_conseliation_of_louis_nweed_d_law_in_trlits_quits_teberry_in_children_muslid_born_herman_lempenn_one_eight_five_four_whischeuse_leaders_of_several_differences_of_privileges_sion_new_mans_to_the_war_knew_his_
ReSFlow	Euler	into_swy_no_and_they_are_also_itn_ha_is_also_ttaught_una__fee_be_onf_the_popular_oo_is_lireaand_for_exampard_altended_fr_a_aerenaen_so_land_external_links_the_d_rib_ol_blt_deer_opern_type_tad_seelanes_and_to_cannabse__k_vemuns_i__endite_fere_the_level_ard_

Table 6: **Generated promoter sequence samples.** Each generated promoter has length 1024. For readability, we show only the first 120 bases of one selected sample for each model and sampler. Spaces are inserted every 40 bases only for line breaking and are not part of the DNA sequence. Quantitative metrics are computed using the full-length sequences.

Model	Sampler	First 120 bases of generated promoter sequence
LinearFM	ODE	AAGAGCTCGGCCGCGGGCTCGGGTGCCCTCCCCCCGGCCC CAAAGCCCCGGGGATTTCCAAGGCCGGCAAGGCCGGGGGG CAGGGGGGAAGAGGGGCGGCACTGGGAGGGGGGATGCGAA
LinearFM	Euler	CTCGCCATCGGGCAGGGGGCTATTTTGCATTCTAACGCCA GGCAGTAGTCGTTTGGTCGCATGCAAACTCAGATGCGGGG GATTGGGTGGCATTCGGTCTCGGCGTTAGACCGAAACCAG
SFM	ODE	AAGAGCTAAGCCGCGGGCTCAGGTGCCCTCCCCCCAGCCC CAAAGCCCCGGGGAGCTCCAACGCCACCACCCCCAGGCGG CAGACCGGAACACGGGCGCCACTGCCAGGAAGGATGCGAA
SFM	Euler	AGGGGCCAAGCCCCGGCGGCGCGTGCGGGGCCGTCAGCTT CAGCCCCCCAGCGGATCCCGTAGACGCCAGCCCCAGGCGG CACTCCGGGACGCGGGCGCCGTCCCGGAGAAGGGCGCGGA
R-LFM	ODE	AAGAACTCCGCCGCGGGCTCGGGTGCCCTCCCCCCGGCCC CAAAGCCGCGGGGAACGCCACCGCCACCACAGCCGGCCGG CAGAAGGGAACAGGGGCGGCCCTAGCAGGCGGGAGGCGAA
R-LFM	Euler	AAGAACTCGGCCGCGGGCTCAGGTGCACTACCCCCGGCCC CAAAGCCGCGGGAAACGCCACCGCCAGCAAAGCCGGGGGG CAGAAGGGAACAGGGGCGGCCCTAGGAGGGGGGAGGCGAA
ReSFlow	ODE	AAGAGCTAAGCCGCGGGCTCAGGTGCCCTCCCCCCAGCCC CACAGCCCCGGGGAGCTCCAACGCCACCACCCCCAGGCGG CAGCCCGGAACACGGGCGCCCCTGCCAGGAAGGATGCGAA
ReSFlow	Euler	AAGACCTAAGCCGCGGGCTCAGGTGCCCTCCCCCCAGCCC CACAGCCCCGGGGAGCTCCAACGCCACCACCCCCAGGCGG CAGCCCGGAACACGGGCGCCCCTGCCAGGAAGGATGCGAA

generation. ReSFlow produces more structured character-level fragments than the non-rectified baselines, consistent with the BPC results in Table 3.

For promoter sequence design, Table 6 shows the first 120 bases of generated 1024-bp promoter sequences. Only prefixes are shown for readability, while all quantitative metrics are computed using the full-length sequences. ReSFlow maintains stable nucleotide patterns under one-step sampling, consistent with its SP-MSE performance in Table 4.

G Limitations and Broader Impact

While our manifold-based formulation successfully captures the intrinsic geometry of discrete data, it relies on model assumptions that may be violated in complex real-world scenarios. For example, the Fisher–Rao geometry inherently assumes semantic equidistance—treating all categorical states symmetrically without prior semantic bias. In reality, discrete tokens (e.g., words or subwords) possess rich, hierarchical semantic proximity (e.g., "cat" is conceptually closer to "dog" than to "apple"). While our method prevents the geometric distortion seen in continuous embedding methods (like ReDi), it does not currently incorporate pre-existing semantic priors. The assumption that pure statistical distances are sufficient may limit performance on tasks where capturing pre-trained semantic correlations is crucial.

ReSFlow is a foundational methodological contribution to generative modeling on the probability simplex, and is not tied to any particular deployment. On the positive side, it offers a geometry-aware and sampling-efficient framework that can benefit discrete generative tasks in language, biology (e.g., DNA/protein sequences), and other scientific domains where categorical data are central, while its straightening property reduces inference cost and the associated energy footprint. On the negative side, ReSFlow shares the generic risks of all improved generative models: higher-quality and faster discrete generation could in principle be misused for disinformation or other malicious content.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are aligned with the proposed method, theoretical motivation, and empirical results. We clearly state the scope of the work, including that the proposed rectification strategy is designed for categorical distributions on the probability simplex and is evaluated on the considered datasets.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the limitations of the proposed manifold-based formulation in the "Limitations and Broader Impact" section. In particular, it notes that the Fisher–Rao geometry treats categorical states symmetrically and does not incorporate pre-existing semantic priors, which may limit performance in tasks where hierarchical semantic relationships among tokens are important.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the assumptions required for the theoretical results and provides complete proofs in the appendix. The main text presents the key formulations and proof intuitions, while the formal derivations are given with the corresponding theorem and lemma statements properly numbered and cross-referenced.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the information needed to reproduce the main experimental results, including the datasets, evaluation metrics, model configurations, training procedures, and baseline methods. Additional implementation and experimental details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our implementation is publicly available at `https://github.com/xiaohuanjia/ReSFlow`. The repository includes environment setup instructions, data access and preprocessing instructions, and scripts for reproducing the main experimental results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification:The paper specifies the experimental settings, including datasets, data preprocessing, train/test protocols, model architectures, hyperparameters, optimizers, and evaluation metrics. These details are provided in the experimental section and appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: We report error bars for the main experiments where repeated runs are computationally feasible. For experiments without explicit error bars, especially sample-based evaluations, we reduce estimation variance by increasing the number of generated samples and apply the same evaluation protocol across all methods.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the compute resources used for the experiments, including the type of hardware, GPU memory, and approximate training or evaluation time. These details are provided to facilitate reproducibility of the reported results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The work is methodological in nature, uses standard benchmark datasets, and does not involve personally identifiable information, human-subject experiments, or unsafe data collection.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses both positive and negative broader impacts in the "Limitations and Broader Impact" section. It highlights potential benefits for efficient discrete generative modeling in language, biology, and scientific domains, while also noting possible misuse risks associated with improved generative models, such as disinformation or malicious content generation.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk pretrained generative models, scraped datasets, or deployment-ready systems that would require additional safeguards. The contribution is a foundational methodological framework evaluated on standard categorical datasets.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the existing datasets, codebases, and software libraries used in the experiments. The corresponding references, versions, and license information are provided where available.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper releases the implementation of the proposed method as a new code asset, together with documentation and instructions for reproducing the main experimental results.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human-subject research, and therefore IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this paper does not involve LLMs as an important, original, or non-standard component. Any use of LLMs, if any, was limited to writing, editing, or formatting assistance and did not affect the methodology, experiments, or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.