GENERATIVE MODELING OF DISCRETE DATA USING GEOMETRIC LATENT SUBSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a geometric latent-subspace framework for generative modeling of discrete data. Specifically, we introduce latent subspaces in the exponential parameter space of product manifolds of categorical distributions as a novel approach to learning low-dimensional representations of high-dimensional discrete data. The resulting low-dimensional latent space captures statistical dependencies and removes redundant degrees of freedom among the categorical variables. We equip the parameter domain with a Riemannian geometry such that the latent subspace and induced data manifold are related isometrically, enabling consistent flow matching. Exploiting this structure, we propose a geometry-aware dimensionality reduction objective, called geometric PCA (GPCA), which we formulate as a regularized cross-entropy minimization that encourages small Riemannian distances between the data and their reconstructions. In particular, under the induced geometry, geodesics correspond to straight lines in the latent parameter space, allowing flow matching to be performed directly in reduced coordinates. Empirical results show that low-dimensional latent representations suffice to accurately model high-dimensional discrete data and enable substantially more computationally efficient flow matching.

1 INTRODUCTION

Generative modeling of discrete data constitutes a major area of research in machine learning. One strand of this research develops Flow Matching methods on categorical statistical manifolds to learn high dimensional joint distributions of discrete random variables, see Stark et al. (2024); Boll et al. (2024; 2025); Davis et al. (2024); Cheng et al. (2024; 2025); Williams et al. (2025), among others. Key questions in this context concern the *representation* of discrete data and the *geometry* of the spaces on which the corresponding probability distributions are defined.

Our **main contribution** in this paper combines *continuous and geometric* modeling to provide **(1)** the accurate *encoding* and *representation* of high dimensional *discrete* data using Geometric PCA (GPCA), **(2)** a *Riemannian geometry* that isometrically relates the latent GPCA subspace to the induced data manifold, and **(3)** an efficient *generative model* based on Flow Matching in low dimensional manifold structure.

To this end, we extend the *generalized PCA* approach Collins et al. (2001) to categorical statistical manifolds Amari & Nagaoka (2000); Amari (2016). We equip the statistical manifold with the e-metric rather than the Fisher-Rao metric, which makes the latent subspace isometric to the induced data manifold. We then learn this subspace using a regularized cross-entropy objective that controls the Riemannian reconstruction error. We call the representation *Geometric PCA (GPCA)*. Under this geometry, *geodesics* are *straight lines* in the low-dimensional GPCA coordinates, which makes Flow Matching computationally efficient. Figure 1 summarizes our approach, and Appendix C gives a geometric picture through synthetic examples. We demonstrate empirically on realistic high-dimensional datasets that GPCA achieves significant dimensionality reduction. This highlights the basic manifold assumption of data science and machine learning Fefferman et al. (2016); Goldt et al. (2020); Meilă & Zhang (2024), here specifically for *discrete* data and generative models.

Our paper is organized as follows. Section 2 introduces the basic information geometry of categorical statistical manifolds and generalized PCA. Section 3 introduces the exponential geometry compatible with the GPCA parametrization. Section 4 describes the training of the generative model

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

using Flow Matching in the GPCA representation. A more detailed discussion of related work and our contribution is provided in Section 5. Section 6 underpin quantitatively our claims.

Frequently used symbols and basic notation are listed in Appendix A. Mathematical proofs and additional illustrative figures are provided in Appendices B, G, H, and I.

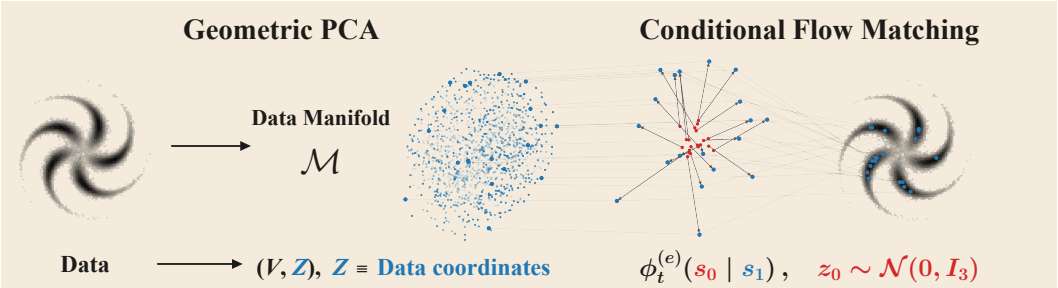


Figure 1: **Approach (sketch):** Geometric PCA represents the data on a nonlinear low-dimensional manifold $\mathcal{M}$ through an exponential parametrization specified by (V, Z). The conditional diffeomorphism $\phi_t^{(e)}$, defines a geodesic interpolation between base samples $s_0 = \varphi^{-1}(V \odot z_0)$, marked in red, and data reconstructions $s_1 = \varphi^{-1}(V \odot z)$, represented in blue. In exponential coordinates, these geodesic interpolants are represented by straight lines, making flow matching computationally efficient while remaining compatible with the underlying manifold geometry.

2 Discrete distributions and generalized PCA

2.1 Geometry of categorical distributions

We recall some basic concepts from information geometry of discrete distributions (Amari & Nagaoka, 2000; Amari, 2016). Every discrete probability measure p on a finite set of categories $\mathcal{C} = [c] := \{1, \dots, c\}$ is uniquely determined by its singleton probabilities $p_j = p(\{j\})$, $j \in \mathcal{C}$, and can therefore be identified with a point in the probability simplex. Throughout this work, we restrict attention to the relative interior of the simplex, corresponding to categorical distributions with full support, which we denote by $\mathcal{S}_c := \left\{ p \in \mathbb{R}^c_{>0} : \sum_{j\in[c]} p_j = 1 \right\}$. The set $\mathcal{S}_c$ is a smooth $(c-1)$-dimensional statistical manifold, and the additive log-ratio transformation

$$\varphi : \mathcal{S}_c \to \mathbb{R}^{c-1}, \qquad \varphi(p) := \left(\log \frac{p_j}{p_c} \right)_{j\in[c-1]} \tag{1}$$

is a diffeomorphism and defines a global coordinate chart on $\mathcal{S}_c$. Its inverse $p_\theta := \varphi^{-1}(\theta)$ yields a minimal exponential-family parametrization of categorical distributions (Barndorff-Nielsen, 1978; Aitchison, 1982; Brown, 1986). More precisely, defining the sufficient statistic $T(j) := (\delta_{jk})$, $k \in [c-1]$, $j \in [c]$, where δ_{jk} denotes the Kronecker delta, we have

$$p_\theta(j) = \exp\big(\langle \theta, T(j) \rangle - \psi(\theta)\big), \quad j \in [c], \qquad \psi(\theta) := \log\left(1 + \sum_{k=1}^{c-1} e^{\theta_k}\right), \quad \theta \in \mathbb{R}^{c-1}. \tag{2}$$

Here, in particular, $T(c) = 0 \in \mathbb{R}^{c-1}$ and therefore $p_\theta(c) = \big(1 + \sum_{k=1}^{c-1} e^{\theta_k}\big)^{-1}$.

The log-partition function $\psi : \mathbb{R}^{c-1} \to \mathbb{R}$ is smooth, strictly convex, and its Hessian gives the coordinate representation of the Fisher–Rao metric in natural coordinates (Amari, 2016, Theorem 2.1). The gradient $\nabla\psi(\theta)$ coincides with the first $c-1$ components of $\varphi^{-1}(\theta)$, with the last component recovered by normalization. The convex conjugate ψ^* is the negative entropy (Amari, 2016, Section 1.4), and its Bregman divergence Zhang (2004); Basseville (2013) coincides with the Kullback–Leibler divergence

$$D_{\psi^*}(\eta, \widetilde{\eta}) = D_{\mathrm{KL}}(p, \widetilde{p}). \tag{3}$$

Here, η and $\widetilde{\eta}$ contain the first $c-1$ probabilities of $p, \widetilde{p} \in \mathcal{S}_c$, respectively.

Throughout this work, we consider product distributions with n categorical factors on $\mathcal{S}_c^n := \mathcal{S}_c \times \cdots \times \mathcal{S}_c$. We regard $\mathcal{S}_c^n$ as a product statistical manifold and, depending on the context, equip it with a Riemannian metric, such as the Fisher-Rao metric or with the exponential geometry introduced in Section 3. We emphasize that, although our geometric construction is formulated on the product space $\mathcal{S}_c^n$, the distribution learned by our model is not required to factorize and may represent a general discrete joint distribution.

Data representation. We work with data $\mathcal{X}_N^n = \{x_i\}_{i\in[N]}$ consisting of N data points $x_i = (x_{i1}, \ldots, x_{in})^\top \in \mathcal{X}^n$, where each component takes one of c categorical values in $\mathcal{C}$. Under the one-hot encoding, each data point is identified with a corner of the product simplex. We perturb these points slightly toward the barycenter and work on the full-supported manifold $\mathcal{S}_c^n$.

2.2 Generalized PCA and induced data manifolds

Following the *generalized PCA* framework of Collins et al. (2001), we parametrize categorical distributions through a linear subspace of the natural-parameter space $\Theta := \mathbb{R}^{n\times(c-1)}$,

$$\mathcal{U} := \operatorname{span}\{V^1, \ldots, V^d\} \subset \mathbb{R}^{n\times(c-1)}, \qquad \textbf{(GPCA subspace)} \tag{4}$$

of dimension $d \ll n(c-1)$, where $V^\ell \in \mathbb{R}^{n\times(c-1)}$ denotes the ℓ-th basis element. Given latent coordinates $z \in \mathbb{R}^d$, we write the corresponding natural parameter as

$$\widehat{\theta} := \sum_{\ell=1}^{d} z_\ell V^\ell =: V \odot z \in \mathcal{U}. \tag{5}$$

The basis vectors $V = (V^1, \ldots, V^d)$ along with latent coefficient vectors $Z = (Z_1, \ldots, Z_N)^\top$ associated with the training data $\mathcal{X}_N^n$ are obtained by minimizing the empirical objective

$$\mathcal{L}(V, Z) := \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} D_{\mathrm{KL}}\left(x_{ij}, \varphi^{-1}\big((V \odot Z_i)_j\big)\right). \tag{6}$$

Here and below, φ^{-1} is applied factorwise, with the last probability recovered by normalization as described above. We note that Collins et al. (2001) originally formulated generalized PCA as a Bregman divergence minimization. We write it here in terms of KL divergence by equation 3. Since the entropy of the data is fixed, minimizing this objective is equivalent to minimizing the corresponding cross-entropy loss. The optimization can be carried out by alternating over convex minimization subproblems or by gradient-based methods using automatic differentiation. We refer to Appendix D for further details on the objective and its optimization.

The GPCA subspace $\mathcal{U}$ induces the nonlinear **data manifold** $\mathcal{M} := \varphi^{-1}(\mathcal{U})$. The inverse coordinate map φ^{-1} acts as a *decoder*. Conversely, φ maps categorical distributions back to natural parameters. Once the GPCA parameters $(\widehat{V}, \widehat{Z})$ are obtained by minimizing the objective equation 6, new data can be projected onto $\mathcal{M}$ via

$$\Pi_{\mathcal{M}}(x) = \varphi^{-1}(\widehat{V} \odot \widehat{z}),\ \widehat{z} = \operatorname*{arg\,min}_{z\in\mathbb{R}^d} \frac{1}{n} \sum_{j=1}^{n} D_{\mathrm{KL}}\big(x_j, \varphi^{-1}((\widehat{V} \odot z)_j)\big). \tag{7}$$

By equation 3, this corresponds to a Bregman projection. Figure 1 sketches the GPCA procedure and our approach for learning generative models of discrete data.

3 Exponential geometry and geometric PCA

3.1 Riemannian structure induced on $\mathcal{M}$

The generalized PCA construction defines a linear subspace $\mathcal{U} \subset \mathbb{R}^{n\times(c-1)}$ and the reconstructed data manifold $\mathcal{M} = \varphi^{-1}(\mathcal{U})$. To formulate flow matching on $\mathcal{M}$ in the next section, the geometry on $\mathcal{S}_c^n$ should preserve the relation between the linear latent coordinates in $\mathcal{U}$ and their reconstructed distributions in $\mathcal{M}$. The Fisher–Rao metric does not generally have this property, since its Riemannian geodesics are not straight lines in natural coordinates and geodesics between two points of $\mathcal{M}$

need not remain in $\mathcal{M}$. We therefore pull back the Euclidean metric on $\mathbb{R}^{c-1}$ through the natural-coordinate chart φ defined in equation 1. The resulting geometry makes φ a Riemannian isometry, so that straight lines in $\mathcal{U}$ are mapped to geodesics that remain in $\mathcal{M}$. This leads to the following definition.

Definition 3.1 (e-metric)**.** *We define the e-metric on $\mathcal{S}_c$ as the Riemannian metric obtained by pulling back the Euclidean metric on $\mathbb{R}^{c-1}$ through the natural-coordinate chart φ defined in equation 1. For $p \in \mathcal{S}_c$ and $u, v \in T_p\mathcal{S}_c$, it is given by*

$$g_p^{(e)}(u, v) := \langle \mathrm{d}\varphi_p[u], \mathrm{d}\varphi_p[v] \rangle \,. \tag{8}$$

For $v \in T_p\mathcal{S}_c$, we denote the induced tangent norm by $\|v\|_{e,p}^2 := g_p^{(e)}(v, v)$.

The coordinate matrix of the e-metric in natural coordinates is the identity matrix I_{c-1}. Hence, the global coordinate chart φ is a Riemannian isometry from $(\mathcal{S}_c, g^{(e)})$ to $(\mathbb{R}^{c-1}, \langle \cdot, \cdot \rangle)$. The length of a curve in $\mathcal{S}_c$ therefore agrees with the Euclidean length of its image under φ, and the induced Riemannian distance is

$$d_e(p, q) = \|\varphi(p) - \varphi(q)\|_2 \,, \qquad p, q \in \mathcal{S}_c \quad \textbf{(Riemannian distance)}. \tag{9}$$

We extend the e-metric and its distance factorwise to $\mathcal{S}_c^n$, so that $d_e^2(s, \widetilde{s}) = \sum_{j=1}^n d_e^2(s_j, \widetilde{s}_j)$. Furthermore, Proposition 3.2 shows that the associated Levi–Civita connection coincides with the exponential connection $\nabla^{(e)}$, studied in information geometry Amari & Nagaoka (2000); Amari (2016). For $s_0, s_1 \in \mathcal{S}_c^n$, we write $\exp_{s_0}^{(e)}(\cdot)$ and $\log_{s_0}^{(e)}(\cdot)$ for the exponential and logarithmic maps associated with $\nabla^{(e)}$ at s_0, respectively. The e-geodesic interpolant between s_0 and s_1 is then given by

$$\phi_t^{(e)}(s_0 \mid s_1) = \exp_{s_0}^{(e)} \left(t \log_{s_0}^{(e)}(s_1) \right), \qquad t \in [0, 1] \quad \textbf{(geodesic interpolant)}. \tag{10}$$

In the following propositions we show that the Riemannian structure on $\mathcal{S}_c^n$ defined above captures the e-geometry. In particular, geodesic interpolants between reconstructed points in $\mathcal{M}$ (cf. Figure 1) remain close, in terms of the Riemannian distance equation 9, to the corresponding geodesic interpolants in $\mathcal{S}_c^n$ whenever the corresponding initial and terminal points are close.

Proposition 3.2. *The Riemannian connection associated with the e-metric g_e on $\mathcal{S}_c^n$ coincides with the e-connection $\nabla^{(e)}$.*

Proposition 3.3 (isometry relations)**.** *The mapping $\varphi^{-1} : \mathcal{U} \to \mathcal{M} \subset \mathcal{S}_c^n$ defines an isometric embedding of $\mathcal{U}$ equipped with the Euclidean metric, into $\mathcal{S}_c^n$ is equipped with the metric g_e. In particular, φ^{-1} maps geodesics in $\mathcal{U}$, i.e. lines of the form $\theta_t = (1-t)\theta_0 + t\theta_1$, to e-geodesics in $\mathcal{S}_c^n$.*

Proposition 3.4 (geodesic interpolants on $\mathcal{M}$ and $\mathcal{S}_c^n$)**.** *Let $p_0, p_1 \in \mathcal{S}_c^n$ and let $\widehat{p}_0, \widehat{p}_1 \in \mathcal{M}$ denote their latent subspace approximations. If the initial and terminal points are close to their approximations in Riemannian distance, then the corresponding geodesic interpolants remain uniformly close*

$$d_e(p_i, \widehat{p}_i) \le \varepsilon, \quad i = 0, 1 \quad \Longrightarrow \quad d_e\big(\phi_t^{(e)}(p_0 \mid p_1), \phi_t^{(e)}(\widehat{p}_0 \mid \widehat{p}_1)\big) \le \varepsilon, \quad t \in [0, 1]. \tag{11}$$

We refer to Appendix B.1 for the proof.

3.2 Geometric PCA

Proposition 3.4 shows that geodesic interpolants on the induced submanifold $\mathcal{M} = \varphi^{-1}(\mathcal{U})$ remain close to the corresponding geodesics interpolants in the data space, provided that the reconstructions are close to the data in Riemannian distance. In this regime, Corollary 4.3 further provides a corresponding approximation bound for the flow matching objective on the data space.

In the next proposition, we show that the Riemannian reconstruction error decreases together with the GPCA loss.

Proposition 3.5. *Let $K \subset \mathcal{S}_c$ be a compact subset such that $\varphi(K)$ is convex. Then there exist constants $C_1, C_2 > 0$ such that, for all $p, p' \in K$,*

$$C_1 d_e^2(p, p') \;\le\; D_{\mathrm{KL}}(p, p') \;\le\; C_2 d_e^2(p, p'). \tag{12}$$

In particular, on K, the KL divergence and the squared e-distance are equivalent up to multiplicative constants.

Remark. In practice, minimizing the GPCA loss in equation 6 does not necessarily ensure that the reconstructions remain within a fixed compact subset of $\mathcal{S}_c^n$. Consequently the cross-entropy loss may decrease substantially while the Riemannian reconstruction error does not. Consequently, we address this issue by regularizing the GPCA objective to discourage reconstructed points from approaching the boundary.

Geometric regularization. We introduce an explicit e-distance penalty and minimize the **GPCA Loss**

$$\mathcal{L}_\lambda(V,Z)=\frac{1}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n} D_{\mathrm{KL}}(x_{ij},\varphi^{-1}((V\odot Z_i)_j))+\lambda\, d_e^2(x_{ij},\varphi^{-1}((V\odot Z_i)_j)). \tag{13}$$

The parameter $\lambda \geq 0$ controls the trade-off between KL reconstruction error and Riemannian reconstruction error. This regularization provides improved control of the e-geodesic reconstruction on $\mathcal{M}$ relative to that on $\mathcal{S}_c^n$. Empirically, we find that minimizing the objective in equation 13 reduces the Riemannian reconstruction error while retaining low cross-entropy and mean Hamming reconstruction errors. The regularization discourages reconstructions from approaching the boundary of $\mathcal{S}_c^n$. Section 6 and Appendix G demonstrate these effects and compare the previously introduced objectives. Due to this geometric set-up we call our method *Geometric PCA (GPCA)*.

4 Flow matching on $\mathcal{M}$

The goal of *Flow Matching (FM)* Liu et al. (2022); Albergo & Vanden-Eijnden (2022); Lipman et al. (2023), is to learn a time-dependent vector field which generates a probability path P_t interpolating between a base distribution P_0 and the data distribution P. When the data distribution is discrete, several works Stark et al. (2024); Boll et al. (2024; 2025); Davis et al. (2024); Cheng et al. (2024; 2025); Williams et al. (2025) proposed to work within the simplex geometry, following the Riemannian FM framework Chen & Lipman (2024).

The goal is to learn a time-dependent vector field

$$u_t : \mathcal{S}_c^n \;\to\; T_0^n, \tag{14}$$

which induces a flow that generates the probability path $P_t = (\phi_t)_\sharp P_0$ satifying $P_{t=0} = P_0$ and $P_{t=1} = P$. The induced flow ϕ_t is the unique solution of the ordinary differential equation (ODE)

$$\frac{d}{dt}\phi_t(x) = u_t(\phi_t(x)), \quad \phi_0(x) = x. \tag{15}$$

The choice of the base distribution P_0 is arbitrary for these models. Common choices are uniform distribution on the simplex or a transformed Gaussian distribution on the simplex. The time horizon of the flows can also be extended to $\mathbb{R}_+$.

Since neither P_t nor u_t are available in closed form, the FM framework instroduces geodesic interpolants $\phi_t(\cdot|x_1) : \mathcal{S}_c^n \to \mathcal{S}_c^n$ *conditioned* on the data, and to minimize a conditional FM objective. The choice of conditional interpolants and underlying geometry used on the FM objective may vary. We propose to work within the e-geometry, and minimize the conditional FM objective

$$\mathcal{L}_{\mathrm{CFM}}^{(e)}(w) = \mathbb{E}_{t,x_0,x_1}\Big[\|v_t(x_t;w) - \partial_t\phi_t^{(e)}(x_0|x_1)\|_{e,x_t}^2\Big], \tag{16}$$

with $t \sim \mathrm{unif}(0,1)$, $x_0 \sim P_0$, $x_1 \sim P_1$, $\|\cdot\|_{e,x_t}$ being the norm in Definition 3.1, and $x_t = \phi_t^{(e)}(x_0 \mid x_1)$ being the geodesic interpolants in equation 10. Our choice of conditional interpolants and the Riemannian structure allows the following formulation in *natural* parameters.

Proposition 4.1 (CFM in natural parameters)**.** *The CFM objective equation 16 is equivalent to*

$$\mathcal{L}_{\mathrm{CFM}}^{(e)}(w) = \mathbb{E}_{t,\theta_0,\theta_1}\Big[\|v_t^\psi(\theta_t;w) - (\theta_1-\theta_0)\|_2^2\Big], \tag{17}$$

where $\theta_0 \sim \varphi_\sharp P_0$ and $\theta_1 \sim \varphi_\sharp P_1$ are the natural parameters corresponding to x_0 and x_1, $\theta_t = t\theta_1 + (1-t)\theta_0$, and $v_t^\psi(\theta_t;w) \coloneqq d\varphi_{\varphi^{-1}(\theta_t)}\big[v_t(\varphi^{-1}(\theta_t);w)\big]$.

The objective function equation 17 shows that flow matching with respect to the e-geometry of the simplex is performed via *straight line interpolation* in the space of natural parameters.

Remark 4.2 (computational savings)**.** *The mapping*

$$\mathcal{Z} \xrightarrow{V} \mathcal{U} \xrightarrow{\varphi^{-1}} \mathcal{M} \subset \mathcal{S}_c^n \tag{18}$$

allows us to pull back the metric g_e from $\mathcal{S}_c^n$ to $\mathcal{Z}$. The resulting scalar product on $\mathcal{Z}$ is given by $\langle Vz, Vz' \rangle$, for all z, z' in the tangent bundle $T\mathcal{Z}$. We denote the corresponding norm function by $\|\cdot\|_V$. The CFM objective in $\mathcal{Z}$ then reads

$$\mathcal{L}_{\mathrm{CFM}}^{\mathcal{Z}}(w_z) = \mathbb{E}_{t,z_0,z_1}\Big[\|v_t(z_t; w) - (z_1 - z_0)\|_V^2\Big]. \tag{19}$$

Choosing $p_z = \mathcal{N}(0, I_d)$ yields the normal distribution $V_\sharp p_z$ in $\mathcal{U}$, which we can use in the objective the objective in equation 17 to identify both CFM objectives. This allows for significant computational savings through lower-dimensional representations and smaller network architectures, since the network architecture operates in the lower-dimensional space $\mathcal{Z}$.

Corollary 4.3 (approximate flow matching)**.** *Let (V, Z) be the GPCA parameters obtained from minimizing the GPCA objective equation 13, and assume that $v_t^\psi(\cdot; w)$, defined in Proposition 4.1, is L-Lipschitz uniformly in t. If for some $\varepsilon > 0$, the reconstruction error satisfies*

$$d_e(x_i, \widehat{x}_i) \le \sqrt{\varepsilon}, \quad x_i \sim P_i,\ \widehat{x}_i = \Pi_{\mathcal{M}}(x_i),\ i = 0, 1, \tag{20}$$

then it also holds

$$\begin{aligned}\mathbb{E}_{t,x_0,x_1}&\Big[\|v_t(x_t; w) - \partial_t \phi_t^{(e)}(x_0|x_1)\|_{e,x_t}^2\Big] \\ &\le (1 + \sqrt{\varepsilon})\mathbb{E}_{t,\widehat{x}_0,\widehat{x}_1}\Big[\|v_t(\widehat{x}_t; w) - \partial_t \phi_t^{(e)}(\widehat{x}_0|\widehat{x}_1)\|_{e,\widehat{x}_t}^2\Big] + (L+2)^2(\varepsilon + \sqrt{\varepsilon}),\end{aligned} \tag{21}$$

where $t \sim \mathrm{unif}(0, 1), x_0 \sim P_0, x_1 \sim P_1$ and $\widehat{x}_0 \sim (\Pi_{\mathcal{M}})_\sharp P_0, \widehat{x}_1 \sim (\Pi_{\mathcal{M}})_\sharp P_1$.

Corollary 4.3 and Proposition 4.1 motivate flow matching on $\mathcal{M}$ and $\mathcal{U}$, respectively. We choose the target distribution $\widehat{P}_1 = (\Pi_{\mathcal{M}})_\sharp P_1$, whose reconstructions have natural parameters in $\mathcal{U}$. The assumption in equation 20, for $i = 1$, is addressed with the objective equation 13, which controls the reconstruction error. See also Section 6 and Appendix G for data evaluations. Choosing P_0 supported on $\mathcal{M}$ ensures zero reconstruction error and the assumption in equation 20 for $i = 0$. In practice, v_t is parametrized by a neural network, for which we assume the uniform Lipschitz bound required by Corollary 4.3. We refer to Appendix B.2 for the proofs.

5 Related work and discussion

We distinguish between *prior* and *recent and current* related work, briefly reporting both in order to position our contribution and to point out limitations and perspectives.

Prior related work. A major component of our approach, viz. employing *geometric* PCA in the canonical (exponential) parameter space of discrete distributions, has been motivated by a precursor, namely *generalized* PCA, introduced by Collins et al. (2001). The basic idea is to regard the squared Euclidean norm, which is used as the objective function for Euclidean data compression in classical principal component analysis (PCA) Jolliffe (2002), as a special divergence function Basseville (2013) and to replace it with a general Bregman divergence induced by the log-partition function of a regular, minimally represented exponential family of distributions Brown (1986), Barndorff-Nielsen (1978). From a *geometric* viewpoint, Bregman divergences play a prominent role in the general theory of contrast functions Matumoto (1993); Zhang (2004), in particular in connection with statistical manifolds and information geometry Lauritzen (1987); Amari & Nagaoka (2000); Ay et al. (2017).

Historically, the set-up of GPCA relates through the notion of a 'link function' associated with the generalized linear models of classical statistics for regression McCullagh & Nelder (1989); Critchley et al. (1994); Kivinen & Warmuth (2001) and applications to online learning Kivinen & Warmuth (1997); Mahony & Williamson (2001). The gradients of the aforementioned log-partition functions of exponential families of distributions provide a set of canonical link functions.

Our approach elaborates this idea for joint distributions of *discrete* high-dimensional data and combines it with a *geometric* latent space suited for generative modeling (see Sections 2 and 3).

Recent and current related work. Several recent works formulate generative modeling of discrete data through flow matching on statistical manifolds. The information-geometric approaches of Boll et al. (2024; 2025) consider statistical product manifolds of categorical distributions with full support, equipped with the canonical Fisher–Rao metric and the e-connection. This set-up provides a specific novel instance of the general geometric approach Chen & Lipman (2024) to flow matching Liu et al. (2022); Albergo & Vanden-Eijnden (2022); Lipman et al. (2023). The recent works Cheng et al. (2024; 2025) also fall into this category. In addition, the latter paper studies various α-connections besides the e-connection corresponding to $\alpha = 1$. A common limitation of these approaches is that they do not explicitly employ latent subspaces. In particular, (Cheng et al., 2024, pp. 10) point out the fixed input size and the independence assumption between classes as limitations. **Our GPCA subspace approach** addresses these issues in three respects: it enables efficient encoding and compression of discrete data, represents *non*-factorizing discrete joint distributions and hence statistical dependencies, and carries a natural Riemannian geometry that isometrically relates latent and data spaces. See Sections 6 and G for further analysis and empirical results. Regarding *latent space models*, we point out that our encoding map (cf. Figure 1) is *directly* given by information geometry and the geometric representation of discrete joint distributions as statistical manifolds. This stands in contrast to encoder maps that are parametrized and *learned* by a network in more general *non*-discrete scenarios like, e.g., in auto-encoder architectures Dao et al. (2023); Samaddar et al. (2025). However, despite the *absence* of a *pre-trained* encoding map and hence using *purely geometric* and *unsupervised* latent space embeddings in our approach (in particular, no encoder training is used), the resulting data representations appear to be natural and semantically meaningful, cf. Figures 13 and 14 Research on *generative models* for *discrete* data has become very active recently, e.g. Stark et al. (2024); Boll et al. (2024; 2025); Davis et al. (2024); Cheng et al. (2024; 2025). We cannot provide a comprehensive review here. An interesting *alternative geometric* approach has been proposed recently in the paper Williams et al. (2025). The authors employ Aitchison's geometry known from the subarea of statistical research concerned with compositional data Aitchison (1982); Filzmoser et al. (2018) and point out as major ingredient of their approach a bijection between the probability simplex and a Euclidean domain, which enables to model and work in the latter space. In this context, we note that information geometry equips statistical manifolds with a dual geometric structure Amari & Nagaoka (2000). For the manifold of categorical distributions considered here, this structure is dually flat. A detailed comparison with compositional data analysis is given by Erb & Ay (2021), who emphasize the invariance properties of the Fisher information metric and the role of Bregman divergences and the Pythagorean theorem of information geometry.

6 Experiments and limitations

6.1 GPCA reconstruction error

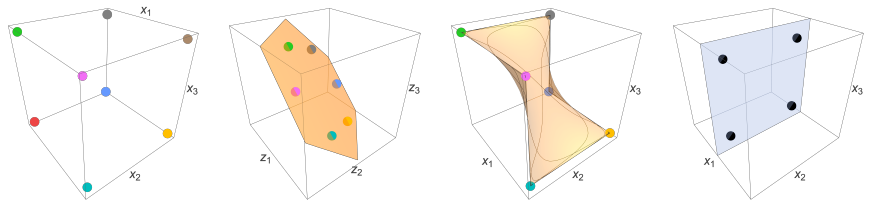
We evaluate reconstruction quality using the mean Hamming distance (MHD), which measures the fraction of mislabeled nodes, cf. equation 62, and additionally report the mean squared Riemannian distance (e-distance) in equation 68 to assess the geometric approximation error underlying the bounds in Proposition 3.4 and Corollary 4.3. We vary the latent dimension d and regularization parameter λ across synthetic categorical data, binarized MNIST LeCun et al. (2010), binarized Fashion-MNIST Xiao et al. (2017), Cityscapes Cordts et al. (2016), and promoter and enhancer DNA sequence datasets, following the experimental setups of Avdeyev et al. (2023) and Stark et al. (2024), respectively. For brevity, we report representative Cityscapes results in Table 1 and provide further dataset specifications, experimental configurations, and extended results in Appendix G.

Across multiple datasets, we observe that the proposed GPCA objective decreases consistently with the mean Hamming distance, yielding low reconstruction error at relatively small latent dimensions. However, the corresponding Riemannian distances do not necessarily decrease at the same rate. Table 1 illustrates this behavior and the effect of regularization. At fixed $\lambda = 10^{-3}$, increasing the latent dimension d reduces cross-entropy, MHD, and e-distance, even achieving zero label reconstruction error on the full dataset for the reported latent dimensions $d \geq 512$. In contrast, at fixed $d = 256$, decreasing λ reduces cross-entropy and MHD but increases the Riemannian distance. Thus, exact label reconstruction does not ensure small geometric approximation error, and the regularization parameter controls the trade-off between the two.

Table 1: Cityscapes reconstruction metrics for varying d and λ.

	Varying d with fixed $\lambda{=}10^{-3}$				Varying λ with fixed $d{=}256$			
	$d{=}256$	$d{=}512$	$d{=}1024$	$d{=}2048$	$\lambda{=}1$	$\lambda{=}0.01$	$\lambda{=}0.001$	$\lambda{=}0$
Cross-Entropy	0.119	0.087	0.074	0.0689	5.133	0.207	0.119	0.099
MHD	0.0004	0.00	0.00	0.00	0.144	0.012	0.0004	0
e-Distance	18.204	8.418	3.286	0.868	4.91	7.24	18.20	58.23

6.1.1 GPCA vs PCA



(a) (b) (c) (d)

Figure 2: **GPCA- vs. PCA-subspaces.** Toy example: nonlinear GPCA decoding recovers more discrete points than the illustrated PCA projection at equal latent dimension. **(a)** The hypercube $\mathcal{X}{=}\{0,1\}^3$. Two-dimensional affine projection with rounding recovers at most four vertices exactly. **(b)** A two-dimensional *geometric latent* GPCA subspace recovers six vertices using *real* coordinates. **(c)** The *nonlinear* manifold $\mathcal{M}$ induced by GPCA. **(d)** since $\mathcal{X}$ is isotropic, PCA reveals no dominant direction. The illustrated projection yields four nonintegral points outside $\mathcal{X}$. Appendix E compares GPCA and PCA reconstruction errors across datasets at equal latent dimensions.

6.2 Generation quality

6.2.1 Binarized MNIST

We evaluate generation quality on binarized MNIST LeCun et al. (2010), with 28^2 binary pixels. Table 2 compares our method with D3PM Austin et al. (2021), DDSM Avdeyev et al. (2023), DirichletFM Stark et al. (2024), DFM Gat et al. (2024), and variants of SFM Cheng et al. (2024), using baseline FID values from Cheng et al. (2024). Since learning is performed in the low-dimensional space $\mathcal{Z}$, we use a different architecture, as theirs is not suited to this representation. This reduces training time to under 20 minutes, compared with the 6 hours reported in Cheng et al. (2024). Appendix F provides hardware and training details. Table 3 compares MHD, e-distance, and FID for our model across regularization weights. Increasing λ reduces the Riemannian distance but increases the Hamming distance, reflecting the trade-off discussed in Section 6.1. The lowest FID occurs at $\lambda{=}10^{-4}$, indicating that moderate regularization benefits generation quality. Appendix H provides additional qualitative classifier metrics across latent dimensions in a separate experiment.

Table 2: FID on binarized MNIST.

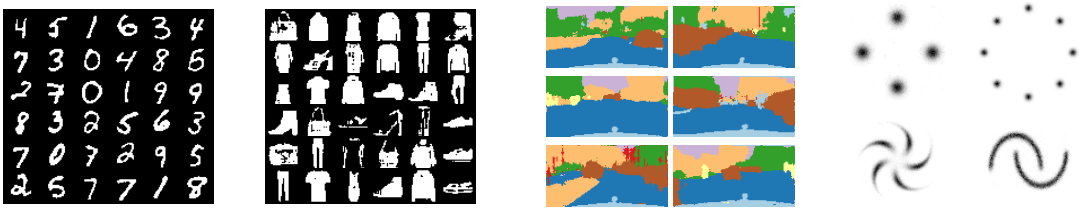
Model	**FID** $\downarrow$
SFM w/ OT	4.62
SFM w/o OT	5.15
SFM w/o π	8.21
LinearFM	5.91
DirichletFM	77.35
DDSM	7.79
D3PM	67.36
DFM	15.34
GPCA + FM (ours), $d = 32$, $\lambda = 10^{-4}$	**1.89**

Table 3: Metrics for binarized MNIST at $d = 32$.

λ	**MHD** $\downarrow$	$d_e^2 \downarrow$	**FID** $\downarrow$
0	0.0077	14.453	2.39
10^{-5}	0.0078	13.743	2.10
10^{-4}	0.0083	10.065	**1.89**
10^{-3}	0.0102	4.011	2.35
10^{-2}	0.0143	1.699	3.30
10^{-1}	0.0202	1.374	5.74

6.2.2 SAMPLE GENERATION

Figure 3 shows samples generated by our FM model (Section 4). Appendix I assesses novelty by comparing additional samples with their nearest training examples. Appendices G.1 and J provide dataset configurations and further experiments on promoter and enhancer DNA datasets following the experimental setups from Avdeyev et al. (2023) and Stark et al. (2024), respectively.



(a) (b) (c) (d)

Figure 3: **Samples of FM model described in Section 4**. Generated samples for (a) MNIST with $d = 32$, $\lambda = 10^{-2}$, (b) binarized Fashion-MNIST with $d = 64$, $\lambda = 10^{-2}$, (c) Cityscapes with $d = 512$, $\lambda = 10^{-3}$, and (d) the empirical joint distribution with $d = 16$, $\lambda = 10^{-3}$.

6.2.3 TOY DATA KL

We compare our method (GPCA + FM) with Linear FM, DirichletFM Stark et al. (2024), FisherFM Davis et al. (2024), and GSFM Tang et al. (2025) on synthetic categorical data with varying category counts K. The reference experiment uses length-four sequences drawn from a random categorical distribution and evaluates the KL divergence between generated and target distributions. Setup and baselines follow Tang et al. (2025, Table 7). Our curves report the minimum KL divergence over five seeds for $d = \lfloor K/2 \rfloor$ and $\lambda \in 10^{-3}, 10^{-4}$, testing generation quality in a lower-dimensional GPCA subspace.

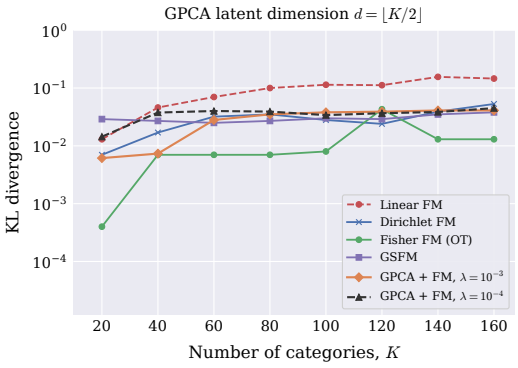


Figure 4: KL divergence across categories.

6.3 FLEXIBLE APPLICABILITY, COMPUTATIONAL EFFICIENCY AND LIMITATIONS

Once the GPCA representation is learned, FM training uses a unified, dataset-independent formulation, allowing the same architecture and training configuration to be reused across datasets. This simplifies implementation and facilitates reusability. Computationally, the FM objective in equation 19 offers numerical advantages by applying the network to low-dimensional coordinates in $\mathcal{Z}$. Appendix F reports GPCA compute and memory requirements.

Our approach is limited by the compressibility of the data. For the DNA datasets for example, low e-metric and low Hamming distance cannot be achieved simultaneously even at relative small compression factors, cf. Tables 21–23. A second limitation is the regularization parameter λ in equation 13, which controls this reconstruction–distance trade-off and is selected empirically. A systematic, data-dependent choice is left for future work.

7 CONCLUSION

We introduced a novel generative model based on geometric latent subspaces for discrete data, combining nonlinear decoding, compression, and efficient flow matching in the low-dimensional latent space. An isometry links latent and data manifolds. Future work will explore GPCA-based consistency models and discrete-data analysis via this isometry.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, clarity checks, and revisions to LaTeX formatting and layout. The scientific arguments, experimental results, and original text, tables, and plots were developed by the authors. AI assistance did not generate the scientific arguments or experimental results. The authors reviewed all AI-assisted revisions and take full responsibility for the final manuscript.

ETHICS STATEMENT

This work develops a general method for generative modeling of discrete data and evaluates it on established benchmark datasets, including promoter and enhancer DNA sequence datasets following Avdeyev et al. (2023) and Stark et al. (2024). We do not conduct new biological experiments or collect new human-subject data. The DNA experiments assess computational modeling performance, they do not establish the biological function or safety of generated sequences.

REPRODUCIBILITY STATEMENT

Code for reproducing the experiments is included in the supplementary material and will be made publicly available upon acceptance. The assumptions and proofs underlying our theoretical results are provided in Appendix B, and details of the GPCA optimization are given in Appendix D. Computational requirements and training details are reported in Appendix F. Dataset specifications, preprocessing, experimental configurations, and extended results are provided in Appendices G–I. Additional experiments on promoter and enhancer DNA datasets are presented in Appendix J.

REFERENCES

J. Aitchison. The Statistical Analysis of Compositional Data. *J. Royal Statistical Soc. B*, 44(2): 139–177, 1982.

Michael S Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. *arXiv preprint arXiv:2209.15571*, 2022.

S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. Amer. Math. Soc. and Oxford Univ. Press, 2000.

Shun-ichi Amari. *Information Geometry and Its Applications*, volume 194 of *Applied Mathematical Sciences*. Springer, Tokyo, 2016. ISBN 978-4-431-55978-8. doi: 10.1007/978-4-431-55978-8.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured Denoising Diffusion Models in Discrete State-Spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet Diffusion Score Model for Biological Sequence Generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

N. Ay, J. Jost, H. V. Lê, and L. Schwachhöfer. *Information Geometry*. Springer, 2017.

O. E. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. Wiley, Chichester, 1978.

M. Basseville. Divergence Measures for Statistical Data Processing – An Annotated Bibliography. *Signal Proc.*, 93(4):621–633, 2013.

B. Boll, D. Gonzalez-Alvarado, and C. Schnörr. Generative Modeling of Discrete Joint Distributions by E-Geodesic Flow Matching on Assignment Manifolds. *arXiv:2402.07846*, 2024.

B. Boll, D. Gonzalez-Alvarado, S. Petra, and C. Schnörr. Generative Assignment Flows for Representing and Learning Joint Distributions of Discrete Data. *J. Math. Imaging Vision*, 67(34), 2025.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, volume 1. Cambridge university press Cambridge, 2004.

L. D. Brown. *Fundamentals of Statistical Exponential Families*. Institute of Mathematical Statistics, Hayward, CA, 1986.

O. Calin and C. Udriste. *Geometric Modeling in Probability and Statistics*. Springer, 2014.

Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A Sequence-Based Global Map of Regulatory Activity for Deciphering Human Genetics. *Nature genetics*, 54(7):940–949, 2022a.

R. T. Q. Chen and Y. Lipman. Riemannian Flow Matching on General Geometries. In *ICLR*, 2024.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog Bits: Generating Discrete Data Using Diffusion Models with Self-Conditioning. *arXiv preprint arXiv:2208.04202*, 2022b.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical Flow Matching on Statistical Manifolds. *arXiv:2405.16441*, 2024.

Chaoran Cheng, Jiahan Li, Jiajun Fan, and Ge Liu. α-Flow: A Unified Framework for Continuous-State Discrete Flow Matching Models. *arXiv:2504.10283*, 2025.

M. Collins, S. Dasgupta, and R. E. Schapire. A Generalization of Principal Component Analysis to the Exponential Family. In *NIPS*, 2001.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

F. Critchley, P. Marriott, and M. Salmon. Preferred Point Geometry and the Local Differential Geometry of the Kullback-Leibler Divergence. *Ann. Statistics*, 22(3):1587–1602, 1994.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow Matching in Latent Space. *arXiv:2307.08698*, 2023.

Oscar Davis, Samuel Kessler, Mircea Petrache, Ismail Ilkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher Flow Matching for Generative Modeling over Discrete Data. *arXiv:2405.14664*, 2024.

Mathias Drton, Bernd Sturmfels, and Seth Sullivant. *Lectures on Algebraic Statistics*, volume 39 of *Oberwolfach Seminars*. Birkhäuser, 2009. doi: 10.1007/978-3-7643-8905-5.

I. Erb and N. Ay. The Information-Geometric Perspective of Compositional Data Analysis. In P. Filzmoser, K. Hron, J. A. Martín-Fernández, and J. Palarea-Albaladejo (eds.), *Advances in Compositional Data Analysis*, pp. 21–43. Springer, 2021.

C. Fefferman, S. Mitter, and H. Narayanan. Testing the Manifold Hypothesis. *J. Amer. Math. Soc.*, 29(4):983–1049, 2016.

P. Filzmoser, K. Hron, and M. Templ. *Applied Compositional Data Analysis*. Springer, 2018.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete Flow Matching. *Advances in Neural Information Processing Systems*, 37:133345–133385, 2024.

S. Goldt, M. Mézard, F. Krzakala, and L. Zdeborová. Modeling the Influence of Data Structure on Learning in Neural Networks: The Hidden Manifold Model. *Physical Review X*, 10(041044), 2020.

I. T. Jolliffe. *Principal Component Analysis*. Springer, 2nd edition, 2002.

J. Kivinen and M. Warmuth. Exponentiated Gradient versus Gradient Descent for Linear Predictors. *Inform. Comput.*, 132:1–63, 1997.

J. Kivinen and M. K. Warmuth. Relative Loss Bounds for Multidimensional Regression Problems. *Machine Learning*, 45:301–329, 2001.

S. L. Lauritzen. Chapter 4: Statistical Manifolds. In Shanti S. Gupta, S. I. Amari, O. E. Barndorff-Nielsen, R. E. Kass, S. L. Lauritzen, and C. R. Rao (eds.), *Differential Geometry in Statistical Inference*, pp. 163–216. Institute of Mathematical Statistics, Hayward, CA, 1987.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. MNIST Handwritten Digit Database. *ATT Labs [Online]*, 2010.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

R. E. Mahony and R. C. Williamson. Prior Knowledge and Preferential Structures in Gradient Descent Learning Algorithms. *J. Mach. Learning Res.*, 1:311–355, 2001.

T. Matumoto. Any Statistical Manifold has a Contrast Function—on the C^3-Functions Taking the Minimum at the Diagonal of the Product Manifold. *Hiroshima Math. J.*, 23(2):327–332, 1993.

P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

M. Meilă and H. Zhang. Manifold Learning: What, How, and Why. *Annual Review of Statistics and Its Application*, 11:393–417, 2024.

Anirban Samaddar, Yixuan Sun, Viktor Nilsson, and Sandeep Madireddy. Efficient flow matching using latent variables. *arXiv:2505.04486*, 2025.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet Flow Matching with Applications to DNA Sequence Design. *arXiv:2402.05841*, 2024.

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Gumbel-Softmax Flow Matching with Straight-Through Guidance for Controllable Biological Sequence Generation. *arXiv:2503.17361*, 2025.

Bernardo Williams, Victor M Yeom-Song, Marcelo Hartmann, and Arto Klami. Simplex-to-Euclidean Bijections for Categorical Flow Matching. *arXiv:2510.27480*, 2025.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-Mnist: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747*, 2017.

J. Zhang. Divergence Function, Duality, and Convex Analysis. *Neural Computation*, 16(1):159–195, 2004.

A BASIC NOTATION

Spaces

N, n, c:	number of data points, nodes, and categories, respectively
$\mathcal{C} := \{1, \dots, c\} =: [c], \quad c \in \mathbb{N}$	set of categories
$\mathcal{X} := \{e_1, \dots, e_c\} \subset \{0,1\}^c$	one-hot encoding of $\mathcal{C}$
$\mathcal{X}^n := \mathcal{X} \times \cdots \times \mathcal{X}$	one-hot encoding of $\mathcal{C}^n$
$\mathcal{X}_N^n := \{x_1, \dots, x_N\} \subset$ (either $\mathcal{X}^n$ or $\mathcal{S}_c^n$)	given data, training set consisting of N data points
$\mathcal{S}_c := \{p \in \mathbb{R}^c_{>0} : \langle \mathbb{1}_c, p \rangle = 1\}$	discrete distributions with full support
$T_0 := T_0\mathcal{S}_c = \{v \in \mathbb{R}^c \colon \langle \mathbb{1}_c, v \rangle = 0\}$	tangent space of $\mathcal{S}_c$ in ambient coordinates
$\mathbb{1}_{\mathcal{S}_c} := \dfrac{1}{c}(1, 1, \dots, 1)^\top \in \mathcal{S}_c,$	barycenter of $\mathcal{S}_c$
$\mathcal{S}_c^n := \mathcal{S}_c \times \cdots \times \mathcal{S}_c$	product space of factorizing distributions on $\mathcal{C}^n$
$T_0^n := T_0 \times \cdots \times T_0$	tangent space to $\mathcal{S}_c^n$ in ambient coordinates

Parametrizations

$\theta := \varphi(p) \in \mathbb{R}^{c-1}$	natural parameters of $p \in \mathcal{S}_c$
$\Theta := \mathbb{R}^{n \times (c-1)}$	natural-parameter space for n nodes
$\psi(\theta) := \log\Big(1 + \sum_{l=1}^{c-1} e^{\theta_l}\Big)$	log-partition function, cf. equation 2
$\eta := \nabla\psi(\theta) \in \mathbb{R}^{c-1}$	first $c-1$ components of $p_\theta \in \mathcal{S}_c$
$\psi^*(\eta)$:	convex conjugate of ψ, negative entropy
$\varphi \colon \mathcal{S}_c \to \mathbb{R}^{c-1}$	coordinate map (encoder), cf. equation 1
$\varphi^{-1} \colon \mathbb{R}^{c-1} \to \mathcal{S}_c$	inverse coordinate map (decoder)
$p_\theta := \varphi^{-1}(\theta) \in \mathcal{S}_c$	categorical distribution with natural parameters θ

The maps φ and φ^{-1} extend factorwise between $\mathcal{S}_c^n$ and Θ.

Latent subspace and approximation

d:	latent subspace dimension
$\mathcal{U} := \operatorname{span}\{V^1, \dots, V^d\} \subset \Theta$	GPCA subspace, cf. equation 4
$\mathcal{Z} := \mathbb{R}^d$	latent coefficient space
$V := (V^1, \dots, V^d)$	basis elements $V^\ell \in \Theta = \mathbb{R}^{n \times (c-1)}$ obtained from GPCA
$Z := (Z_1, \dots, Z_N)^\top \in \mathbb{R}^{N \times d}$	latent coefficients of training data obtained from GPCA
$\widehat{\theta} := V \odot z := \sum_{\ell=1}^{d} z_\ell V^\ell$	natural parameters associated with $z \in \mathcal{Z}$, cf. equation 5
$\mathcal{M} := \varphi^{-1}(\mathcal{U}) \subset \mathcal{S}_c^n$	induced data manifold, cf. Section 2.2
$\Pi_{\mathcal{M}}$: see equation 7	Bregman projection onto $\mathcal{M}$
$\widehat{p}_i := \varphi^{-1}(\widehat{\theta}_i)$	reconstruction of data point x_i, $\widehat{\theta}_i$ is the reconstructed natural parameter
$\lambda \colon \lambda \geq 0$	regularization weight in equation 13

Flow matching and e-geometry

$g_p^{(e)}$: see Definition 3.1	e-metric; also denoted by g_e
d_e: see equation 9	Riemannian distance induced by the e-metric
$\|\cdot\|_{e,p}$: $\|v\|_{e,p}^2 = g_p^{(e)}(v,v)$	norm on the tangent space at p
$\phi_t^{(e)}(p_0 \mid p_1)$: see equation 10	e-geodesic interpolant
u_t: $\mathcal{S}_c^n \to T_0^n$	flow-generating vector field, cf. equation 14
$v_t(\cdot; w)$:	learned vector field with network parameters w
$v_t^{\psi}(\cdot; w)$: see Proposition 4.1	pushforward of $v_t(\cdot; w)$ through φ
P_0, P_1, P_t:	base distribution, target distribution, and probability path
$\|\cdot\|_V$: $\|z\|_V^2 = \langle Vz, Vz\rangle$	induced latent norm in $\mathcal{Z}$, cf. equation 19

B PROOFS AND ADDITIONAL DETAILS

B.1 PROOFS OF SECTION 3

Proof of Proposition 3.2. By Proposition 1.9.1 in Calin & Udriste (2014), the Christoffel symbols of the e-connection $\nabla^e = \nabla^{(1)}$ vanish in natural coordinates $\theta = \varphi(p)$. In these coordinates, the e-metric has constant identity matrix, so the Christoffel symbols of its Levi–Civita connection also vanish. Hence, the two connections coincide, since their Christoffel symbols agree in the same global coordinate chart. The argument extends factorwise to $\mathcal{S}_c^n$. □

Proof of Proposition 3.3. The map $\varphi^{-1} : \mathcal{U} \to \mathcal{M}$ is the restriction of the inverse coordinate chart $\varphi^{-1} : \mathbb{R}^{n\times(c-1)} \to \mathcal{S}_c^n$ to the linear subspace $\mathcal{U}$. By definition of the e-metric, φ^{-1} is a Riemannian isometry on the full parameter space. Its restriction therefore defines an isometric embedding of $\mathcal{U}$ into $\mathcal{S}_c^n$, with image $\mathcal{M}$. Since straight lines in $\mathcal{U}$ are also geodesics of the ambient Euclidean space, their images under φ^{-1} are e-geodesics in $\mathcal{S}_c^n$. □

Proof of Proposition 3.4. Since φ is a Riemannian isometry, it maps e-geodesics to Euclidean straight lines. Consequently, the e-geodesic in equation 10 satisfies

$$\varphi\big(\phi_t^{(e)}(p_0 \mid p_1)\big) = (1-t)\varphi(p_0) + t\varphi(p_1), \qquad t \in [0,1]. \tag{22}$$

Writing $\theta_i = \varphi(p_i)$ and $\widehat{\theta}_i = \varphi(\widehat{p}_i)$, the assumption gives

$$\|\theta_i - \widehat{\theta}_i\|_2 = d_e(p_i, \widehat{p}_i) \le \varepsilon, \qquad i = 0, 1. \tag{23}$$

Hence,

$$d_e\big(\phi_t^{(e)}(p_0 \mid p_1), \phi_t^{(e)}(\widehat{p}_0 \mid \widehat{p}_1)\big) \overset{equation\ 9}{=} \|\varphi\big(\phi_t^{(e)}(p_0 \mid p_1)\big) - \varphi\big(\phi_t^{(e)}(\widehat{p}_0 \mid \widehat{p}_1)\big)\|_2 \tag{24a}$$

$$\tag{24b}$$

$$\overset{equation\ 22}{=} \|(1-t)\varphi(p_0) + t\varphi(p_1) - \big((1-t)\varphi(\widehat{p}_0) + t\varphi(\widehat{p}_1)\big)\|_2 \tag{24c}$$

$$\overset{\theta_i=\varphi(p_i),\widehat{\theta}_i=\varphi(\widehat{p}_i)}{=} \big\|(1-t)(\theta_0 - \widehat{\theta}_0) + t(\theta_1 - \widehat{\theta}_1)\big\|_2 \tag{24d}$$

$$\le (1-t)\|\theta_0 - \widehat{\theta}_0\|_2 + t\|\theta_1 - \widehat{\theta}_1\|_2 \tag{24e}$$

$$\le (1-t)\varepsilon + t\varepsilon = \varepsilon. \tag{24f}$$

This proves the bound uniformly for $t \in [0,1]$. □

Proof of Proposition 3.5. First note that, since K is compact and φ is continuous, $U_K := \varphi(K)$ is compact. Moreover, U_K is convex by assumption. The Hessian $\nabla^2\psi(\theta)$ is the coordinate representation of the Fisher-Rao metric in natural coordinates, cf. Amari (2016, Section 2.1), and hence it is positive definite for the minimal categorical family.

The smallest and largest eigenvalues of $\nabla^2\psi(\theta)$ depend continuously on θ. Hence, by compactness of U_K, they attain their extrema, and we may define

$$m_K := \min_{\theta\in U_K} \lambda_{\min}\big(\nabla^2\psi(\theta)\big) > 0, \qquad M_K := \max_{\theta\in U_K} \lambda_{\max}\big(\nabla^2\psi(\theta)\big) < \infty. \tag{25}$$

To obtain the corresponding matrix bounds, fix $\theta \in U_K$ and let $q_1, \dots, q_{c-1}$ be an orthonormal eigenbasis of $\nabla^2\psi(\theta)$, with corresponding eigenvalues $\lambda_1, \dots, \lambda_{c-1}$. Writing $v = \sum_{k=1}^{c-1} a_k q_k$, we obtain

$$v^\top \nabla^2\psi(\theta) v = \sum_{k=1}^{c-1} \lambda_k a_k^2, \qquad \|v\|_2^2 = \sum_{k=1}^{c-1} a_k^2. \tag{26}$$

Since $m_K \le \lambda_k \le M_K$ for every $k \in [c-1]$, it follows that

$$m_K\|v\|_2^2 \le v^\top \nabla^2\psi(\theta) v \le M_K\|v\|_2^2, \qquad v \in \mathbb{R}^{c-1}. \tag{27}$$

Equivalently,

$$m_K I_{c-1} \preceq \nabla^2 \psi(\theta) \preceq M_K I_{c-1}, \qquad \theta \in U_K. \tag{28}$$

For $p, p' \in K$, write $\theta = \varphi(p)$ and $\theta' = \varphi(p')$. Since U_K is convex, the Hessian lower bound implies strong convexity in U_K, and hence (Boyd & Vandenberghe, 2004, Equation 9.8) gives

$$\psi(\theta') \geq \psi(\theta) + \langle \nabla\psi(\theta), \theta' - \theta \rangle + \frac{m_K}{2}\|\theta' - \theta\|_2^2. \tag{29}$$

Rearranging terms yields

$$D_\psi(\theta', \theta) = \psi(\theta') - \psi(\theta) - \langle \nabla\psi(\theta), \theta' - \theta \rangle \geq \frac{m_K}{2}\|\theta' - \theta\|_2^2. \tag{30}$$

Likewise, the upper Hessian bound implies the quadratic upper bound in (Boyd & Vandenberghe, 2004, Equation 9.13)

$$\psi(\theta') \leq \psi(\theta) + \langle \nabla\psi(\theta), \theta' - \theta \rangle + \frac{M_K}{2}\|\theta' - \theta\|_2^2, \tag{31}$$

which gives

$$D_\psi(\theta', \theta) \leq \frac{M_K}{2}\|\theta' - \theta\|_2^2. \tag{32}$$

Using $D_{\mathrm{KL}}(p, p') = D_\psi(\theta', \theta)$, cf. Amari (2016, Equation (2.16)) and $d_e(p, p') = \|\theta - \theta'\|_2$ from equation 9, it follows that

$$\frac{m_K}{2} d_e^2(p, p') \leq D_{\mathrm{KL}}(p, p') \leq \frac{M_K}{2} d_e^2(p, p'). \tag{33}$$

The claim follows with $C_1 = m_K/2$ and $C_2 = M_K/2$. □

B.2 Proofs of Section 4

Proof of Proposition 4.1. The natural parameters corresponding to $x_0, x_1 \in \mathcal{S}_c^n$ are given factorwise by

$$\theta_{k,j} = \varphi(x_{k,j}) = \left(\log \frac{x_{k,jl}}{x_{k,jc}}\right)_{l \in [c-1]}, \qquad k = 0, 1, \quad j \in [n]. \tag{34}$$

Since φ is a Riemannian isometry, it maps e-geodesics to Euclidean straight lines. Writing $x_t = \phi_t^{(e)}(x_0 \mid x_1)$, we have

$$\varphi(x_t) = (1-t)\varphi(x_0) + t\varphi(x_1) = (1-t)\theta_0 + t\theta_1 = \theta_t. \tag{35}$$

In particular,

$$\partial_t \theta_t = \theta_1 - \theta_0. \tag{36}$$

Moreover, by the chain rule we have

$$\partial_t \theta_t = d\varphi_{x_t}[\partial_t x_t]. \tag{37}$$

By Definition 3.1, extended factorwise to $\mathcal{S}_c^n$, we compute

$$\begin{aligned}
\|v_t(x_t; w) - \partial_t x_t\|_{e,x_t}^2 &= \left\| d\varphi_{x_t}\left[v_t(x_t; w) - \partial_t x_t\right] \right\|_2^2 && (38)\\
&= \left\| d\varphi_{x_t}\left[v_t(x_t; w)\right] - d\varphi_{x_t}[\partial_t x_t] \right\|_2^2 && (39)\\
&\overset{equation\ 37}{=} \left\| d\varphi_{x_t}\left[v_t(x_t; w)\right] - \partial_t \theta_t \right\|_2^2 && (40)\\
&= \left\| v_t^\psi(\theta_t; w) - (\theta_1 - \theta_0) \right\|_2^2. && (41)
\end{aligned}$$

where the last equation uses the definition $v_t^\psi(\theta_t; w) := d\varphi_{x_t}\left[v_t(x_t; w)\right]$. Taking expectations on both sides, with $\theta_k = \varphi(x_k)$ distributed according to $\varphi_\sharp P_k$, $k = 0, 1$, concludes the proof. □

Proof of Corollary 4.3. We write $x_t = \phi_t^{(e)}(x_0 \mid x_1)$, $\widehat{x}_t = \phi_t^{(e)}(\widehat{x}_0 \mid \widehat{x}_1)$, and denote the corresponding natural parameters by

$$\theta_i = \varphi(x_i), \qquad \widehat{\theta}_i = \varphi(\widehat{x}_i), \qquad i = 0, 1. \tag{42}$$

Since φ preserves distances as a Riemannian isometry, the reconstruction assumption equation 20 gives

$$\|\theta_i - \widehat{\theta}_i\|_2 \le \sqrt{\varepsilon}, \qquad i = 0, 1. \tag{43}$$

Writing $\theta_t = \varphi(x_t)$ and $\widehat{\theta}_t = \varphi(\widehat{x}_t)$, the straight-line representation of the interpolants gives

$$\begin{aligned} \|\theta_t - \widehat{\theta}_t\|_2 &= \|((1-t)\theta_0 + t\theta_1) - ((1-t)\widehat{\theta}_0 + t\widehat{\theta}_1)\|_2 && \text{(44a)} \\ &= \|(1-t)(\theta_0 - \widehat{\theta}_0) + t(\theta_1 - \widehat{\theta}_1)\|_2 && \text{(44b)} \\ &\le (1-t)\|\theta_0 - \widehat{\theta}_0\|_2 + t\|\theta_1 - \widehat{\theta}_1\|_2 \le \sqrt{\varepsilon}. && \text{(44c)} \end{aligned}$$

Furthermore the L-Lipschitz assumption implies

$$\|v_t^\psi(\theta_t; w) - v_t^\psi(\widehat{\theta}_t; w)\|_2 \le L\|\theta_t - \widehat{\theta}_t\|_2 \overset{equation\ 44}{\le} L\sqrt{\varepsilon}. \tag{45}$$

Adding and subtracting the reconstructed vector field and target velocity, we compute

$$\begin{aligned} &\|v_t^\psi(\theta_t; w) - (\theta_1 - \theta_0)\|_2 && \text{(46a)} \\ &= \left\| v_t^\psi(\theta_t; w) - v_t^\psi(\widehat{\theta}_t; w) + v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0) + (\widehat{\theta}_1 - \widehat{\theta}_0) - (\theta_1 - \theta_0) \right\|_2 && \text{(46b)} \\ &\le \|v_t^\psi(\theta_t; w) - v_t^\psi(\widehat{\theta}_t; w)\|_2 + \|v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0)\|_2 + \|(\widehat{\theta}_1 - \widehat{\theta}_0) - (\theta_1 - \theta_0)\|_2 && \text{(46c)} \\ &\overset{equation\ 45}{\le} \|v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0)\|_2 + L\sqrt{\varepsilon} + \|\widehat{\theta}_1 - \theta_1\|_2 + \|\widehat{\theta}_0 - \theta_0\|_2 && \text{(46d)} \\ &\overset{equation\ 43}{\le} \|v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0)\|_2 + (L+2)\sqrt{\varepsilon}. && \text{(46e)} \end{aligned}$$

Since, for any $\eta > 0$,

$$(a+b)^2 \le (1+\eta)a^2 + \left(1 + \frac{1}{\eta}\right) b^2, \tag{47}$$

setting

$$a = \|v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0)\|_2, \qquad b = (L+2)\sqrt{\varepsilon}, \qquad \eta = \sqrt{\varepsilon}, \tag{48}$$

we obtain

$$\|v_t^\psi(\theta_t; w) - (\theta_1 - \theta_0)\|_2^2 \le (1 + \sqrt{\varepsilon})\|v_t^\psi(\widehat{\theta}_t; w) - (\widehat{\theta}_1 - \widehat{\theta}_0)\|_2^2 + (L+2)^2(\varepsilon + \sqrt{\varepsilon}). \tag{49}$$

Taking expectations on both sides and applying Proposition 4.1 to the original and reconstructed interpolants concludes the proof. □

C EXAMPLE: REPRESENTATION OF SYNTHETIC DATASETS

We identify points in the product space $\mathcal{S}_c^n$ with their images in the *meta-simplex* $\mathcal{S}_{c^n}$ through the standard identification of marginal distributions with their corresponding product distributions, cf. (Drton et al., 2009, Section 5.2),

$$\mathcal{T} : \mathcal{S}_c^n \hookrightarrow \mathcal{S}_{c^n}, \qquad p = (p_1, \ldots, p_n) \mapsto p_1 \otimes \cdots \otimes p_n =: P. \tag{50}$$

The entries of $P = p_1 \otimes \cdots \otimes p_n$ are indexed by configurations $(i_1, \ldots, i_n) \in [c]^n$ and given by

$$P(i_1, \ldots, i_n) = \prod_{k=1}^{n} p_k(i_k), \qquad i_k \in \{1, \ldots, c\}. \tag{51}$$

In this representation, $P \in \mathcal{S}_{c^n}$ is the joint probability vector over the c^n possible configurations, while the image $\mathcal{T}(\mathcal{S}_c^n) \subset \mathcal{S}_{c^n}$ consists of those probability vectors admitting the tensor-product factorization above. As a simple illustrative example, consider $c = n = 2$. In this case, the product simplex $\mathcal{S}_2^2$ is embedded as a two-dimensional surface in the three-dimensional meta-simplex $\mathcal{S}_4$. Figure 5 shows this embedded product manifold from three different viewing angles.

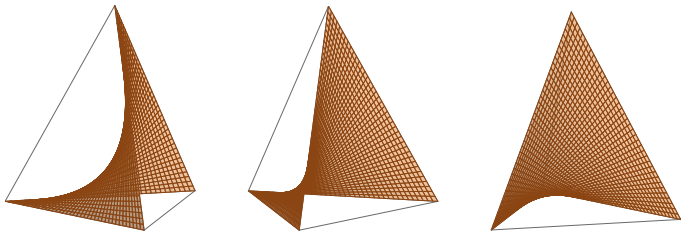
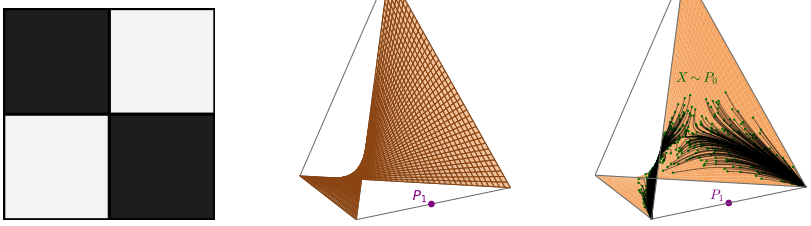


Figure 5: Three views of the embedded product manifold $\mathcal{T}(\mathcal{S}_2^2) \subset \mathcal{S}_4$.

Flow-based models following Riemannian Flow Matching on the product space $\mathcal{S}_c^n$, such as those in Stark et al. (2024); Boll et al. (2024; 2025); Davis et al. (2024); Cheng et al. (2024; 2025); Williams et al. (2025), learn a vector field on $\mathcal{S}_c^n$ that defines a flow ϕ_t transporting a base distribution $P_0 \in \mathcal{P}(\mathcal{S}_c^n)$ into a distribution $\widehat{P}_1 \in \mathcal{P}(\mathcal{S}_c^n)$, whose samples simulate samples from the data distribution P_1. As an illustrative example, we consider again $c = n = 2$ and, for $0 < \varepsilon < 1/4$, the target joint distribution

$$P_1 = (0.5 - \varepsilon, \varepsilon, \varepsilon, 0.5 - \varepsilon)^\top \in \mathcal{S}_4. \tag{52}$$

Although $P_1 \notin \mathcal{T}(\mathcal{S}_2^2)$, samples from P_1 are represented by vertices of the closure of $\mathcal{S}_2^2$. The flow generates samples whose empirical distribution approximates P_1 rather than representing P_1 as a single product distribution. Figure 6a shows the empirical distribution of samples drawn from P_1. Figure 6b shows P_1 in the meta-simplex $\mathcal{S}_4$, emphasising that $P_1 \notin \mathcal{T}(\mathcal{S}_2^2)$. Figure 6c illustrates how a flow ϕ_t on $\mathcal{S}_2^2$ transports samples from the base distribution to samples approximately distributed according to P_1.



(a) Empirical approximation of P_1.

(b) Target P_1 in the meta-simplex.

(c) Flow on $\mathcal{S}_2^2$.

Figure 6: Illustration of the target distribution P_1 defined in equation 52 and its approximation by a flow-based model. (a) Histogram of $N = 10^4$ samples drawn from P_1. (b) The target $P_1 \in \mathcal{S}_4$ lies outside the embedded product manifold $\mathcal{T}(\mathcal{S}_2^2)$. (c) A flow on $\mathcal{S}_2^2$ transports base samples to samples approximately distributed according to P_1.

Our approach learns a latent subspace $\mathcal{U} = \mathrm{span}(V)$ and the corresponding low-dimensional manifold $\mathcal{M} = \varphi^{-1}(\mathcal{U})$. This provides a lower-dimensional space in which to learn the flow. Continuing the example above, we apply GPCA to samples from P_1 to learn the one-dimensional submanifold $\mathcal{M} \subset \mathcal{S}_2^2$, whose embedded image $\mathcal{T}(\mathcal{M})$ is shown in blue in Figure 7.

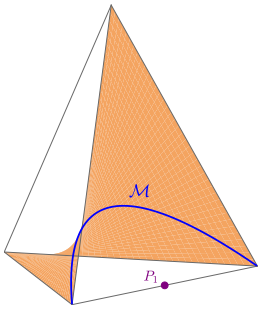


Figure 7: The embedded image $\mathcal{T}(\mathcal{M})$ of the one-dimensional manifold learned by GPCA from samples of P_1 defined in equation 52, shown in blue in the meta-simplex $\mathcal{S}_4$.

The same geometric picture provides intuition for the discrete synthetic datasets used throughout this work, shown in Figure 8, where we use $n = 2$ and $c = 92$. We refer to Appendices G.1 and G.4 for data specifications and visual experiments on these datasets. These histograms should be interpreted analogously to the empirical distribution in Figure 6a, representing discrete realizations of the underlying continuous distributions. Figure 9 illustrates, on the left, the coordinate representations Z of the data, shown in blue, obtained by GPCA, see Section 2.2. The middle image shows the interpolations between base samples and selected data points, while the rightmost image shows the corresponding samples as empirical histograms. By choosing the geometry as described in Section 3, these geodesic interpolations correspond to straight lines in the objective equation 17. Choosing a different geometry, such as the Fisher–Rao geometry, would result in curved interpolations and reduce the computational gains noted in Remark 4.2.

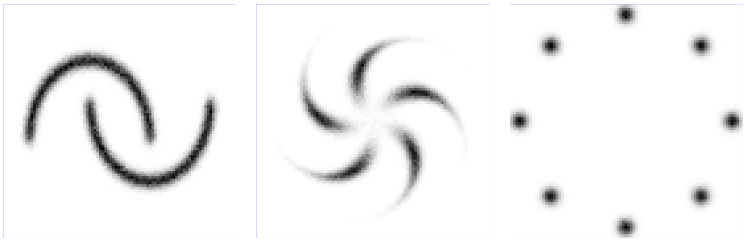


Figure 8: Synthetic datasets: two moons, pinwheel, and Gaussian mixture (left to right).

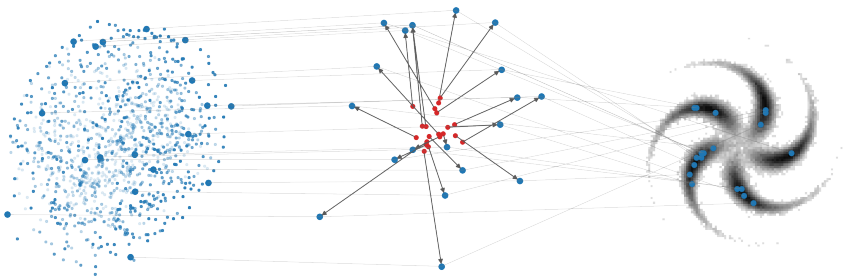


Figure 9: (left) Coordinate representations of the $N = 10^4$ samples of the pinwheel dataset, (center) geodesic interpolations from normal samples to selected coordinate representations, and (right) the corresponding pinwheel samples.

D GPCA LOSS, e-DISTANCE & DATA REPRESENTATION

Throughout this section, let $i \in [N]$ index data points, $j \in [n]$ index nodes, and $l \in [c]$ index classes. We represent factorizing distributions $p \in \mathcal{S}_c^n$ in the minimal parameterization as

$$p_{\theta;j}(l) = \exp\big(\langle \theta_j, T(l)\rangle - \psi(\theta_j)\big), \quad \psi(\theta_j) = \log\Big(1 + \sum_{l=1}^{c-1} e^{\theta_{jl}}\Big), \quad j \in [n], \tag{53}$$

cf. Section 2, where θ_j denote the natural parameters, ψ is the log-partition function, and $T(l) = (\delta_{lk})_{k\in[c-1]}$ denote the sufficient statistic of class l. The mean parameters are given by the gradient of the log-partition function

$$\eta_j := \nabla\psi(\theta_j) = \Big(\frac{e^{\theta_{jl}}}{1 + \sum_{k=1}^{c-1} e^{\theta_{jk}}}\Big)_{l\in[c-1]}, \quad j \in [n], \tag{54}$$

and the last component is obtained by normalization, i.e.

$$\eta_{jc} = 1 - \sum_{l=1}^{c-1} \eta_{jl}. \tag{55}$$

The conjugate of the log-partition function is the negative entropy (Amari, 2016, Section 2.1)

$$\psi^*(\eta_j) = \sup_{\theta_j \in \mathbb{R}^{c-1}} (\langle \theta_j, \eta_j\rangle - \psi(\theta_j)) = \sum_{l=1}^{c-1} \eta_{jl} \log \eta_{jl} + \eta_{jc} \log \eta_{jc}. \tag{56}$$

For hard-label data $x_1, \dots, x_N \in \mathcal{X}_N^n$, Collins et al. (2001) proposed to learn $V \in \mathbb{R}^{n\times(c-1)\times d}$ and $Z \in \mathbb{R}^{N\times d}$ such that the natural parameters are represented as $\widehat{\theta}_i = V \odot Z_i$, cf. equation 5. The representation is estimated by minimizing the Bregman divergence objective

$$\frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} D_{\psi^*}(x_{ij}^-, \nabla\psi((V \odot Z_i)_j)), \tag{57}$$

where $x_{ij}^- = (x_{ijl})_{l\in[c-1]}$ contains the first $c-1$ components of the one-hot encoded data points. Collins et al. (2001) point out that this objective equals the negative log likelihood (NLL) of the data under the model. Indeed, since

$$\psi^*(x_{ij}^-) = \sum_{l=1}^{c} x_{ijl} \log x_{ijl} = 0 \tag{58}$$

for one-hot data, with the convention that $0 \log 0 = 0$, Legendre duality gives

$$D_{\psi^*}(x_{ij}^-, \nabla\psi((V \odot Z_i)_j)) = \psi^*(x_{ij}^-) + \psi(\widehat{\theta}_{ij}) - \langle \widehat{\theta}_{ij}, x_{ij}^-\rangle \tag{59}$$

$$= \psi(\widehat{\theta}_{ij}) - \langle \widehat{\theta}_{ij}, x_{ij}^-\rangle = -\sum_{l=1}^{c} x_{ijl} \log p_{\widehat{\theta};ij}(l). \tag{60}$$

The last expression is the negative logarithm of the probability assigned to the observed class, and hence the objective studied in Collins et al. (2001), for hard-label data, is equivalent to

$$\mathrm{NLL}(V, Z) = -\frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} \sum_{l=1}^{c} x_{ijl} \log p_{\widehat{\theta};ij}(l). \tag{61}$$

The node-wise reconstruction error is measured by the mean Hamming distance (MHD) between the original data points $x_1, \dots, x_N \in \mathcal{X}_N^n$ and their reconstructions, and is defined as

$$\mathrm{MHD} = \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} d_H(x_{ij}, \mathrm{round}\,(\varphi^{-1}(\widehat{\theta}_{ij}))), \quad d_H(p, q) = \frac{1}{2} \sum_{l=1}^{c} \mathbb{1}_{p_l \neq q_l}, \; p, q \in \{e_1, \dots, e_c\}, \tag{62}$$

where $\widehat{\theta}_i = V \odot Z_i$, and (V, Z) are the parameters estimated by minimizing the geometric or generalized PCA loss. Here, round returns the one-hot encoding of the class with largest probability.

The optimization of the NLL is numerically stable and can be carried out efficiently using reverse-mode automatic differentiation. For small latent dimensions, the Hamming distance decreases together with the NLL, but generally does not reach zero. For larger latent dimensions, which still provide a substantial dimensionality reduction, zero Hamming distance is reached much faster in practice. Section G provides a more detailed empirical analysis of the relation between the NLL, Hamming distance, and e-distance for different latent dimensions on benchmark datasets.

Data representation. As discussed in the main text, hard-label data points $x \in \mathcal{X}_N^n$ are embedded into the interior of the product simplex $\mathcal{S}_c^n$ by a smoothing transformation. In practice, we use

$$\widetilde{x}_{ij} = \epsilon \mathbb{1}_{\mathcal{S}_c} + (1-\epsilon)x_{ij}, \qquad 0 < \epsilon < 1, \tag{63}$$

where $\mathbb{1}_{\mathcal{S}_c}$ denotes the barycenter of $\mathcal{S}_c$. This transformation ensures that all entries of $\widetilde{x}_{ij}$ are strictly positive, so that KL divergences and logarithms are well-defined.

For points in $p, \widetilde{p} \in \mathcal{S}_c$, let $\eta, \widetilde{\eta} \in \mathbb{R}^{c-1}$ contain their first $c-1$ components, respectively. By equation 3, the Bregman divergence generated by the convex conjugate ψ^* coincides with the Kullback-Leibler divergence

$$D_{\psi^*}(\eta, \widetilde{\eta}) = D_{\mathrm{KL}}(p, \widetilde{p}). \tag{64}$$

Consequently, denoting the smoothed data $\widetilde{x}$ again by x for simplicity, and writing $\eta_{ij} = (x_{ijl})_{l \in [c-1]}$, the generalized PCA loss reads

$$\mathcal{L}(V, Z) = \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} D_{\psi^*}(\eta_{ij}, \widehat{\eta}_{ij}) = \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} D_{\mathrm{KL}}(x_{ij}, \widehat{p}_{ij}), \tag{65}$$

where $\widehat{\eta}_{ij} = \nabla\psi(\widehat{\theta}_{ij})$, and $\widehat{p}_{ij} = \varphi^{-1}(\widehat{\theta}_{ij})$, with $\widehat{\theta}_i = V \odot Z_i$.

Since

$$D_{\mathrm{KL}}\big(x_{ij}, \widehat{p}_{ij}\big) = \sum_{l=1}^{c} x_{ijl} \log \frac{x_{ijl}}{\widehat{p}_{ijl}} = -\sum_{l=1}^{c} x_{ijl} \log \widehat{p}_{ijl} + \sum_{l=1}^{c} x_{ijl} \log x_{ijl}, \tag{66}$$

the KL divergence decomposes into a cross-entropy term and a negative entropy term that depends only on the data. Therefore, the entropy term is constant with respect to the optimization variables V and Z. Minimizing the KL loss is thus equivalent to minimizing the cross-entropy loss

$$\text{Cross-Entropy}(V, Z) = -\frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} \sum_{l=1}^{c} x_{ijl} \log \widehat{p}_{ijl}, \tag{67}$$

which is the objective that we minimize in practice.

In practice, for the flow matching framework, we want the reconstruction error between data points x and its reconstruction $\widehat{p} \in \mathcal{M}$ to be small, with respect to underlying Riemannian distance in equation 9. In our setting, this means that the mean squared e-distance

$$e\text{-Distance} = \frac{1}{Nn} \sum_{i=1}^{N} \sum_{j=1}^{n} d_e^2(x_{ij}, \widehat{p}_{ij}) \tag{68}$$

should be small. When the initial and terminal points are close to their reconstructions, Proposition 3.4 guarantees that the e-geodesic interpolants in equation 10 on the submanifold $\mathcal{M} = \varphi^{-1}(\mathcal{U})$ remain close to e-geodesic interpolants on the data space. Moreover, Corollary 4.3 provides a corresponding approximation bound for the Flow Matching objective on the data space.

Empirically, we find that the Hamming distance decreases along with the GPCA loss. The e-distance, however, does not decrease as consistently. In particular, minimizing the cross-entropy on the smoothed data instead of the NLL on the hard-label data leads to larger Hamming distances. We refer to Section G for a more detailed empirical analysis of the relation between the NLL, cross-entropy, Hamming distance, and e-distance for different latent dimensions on benchmark datasets.

E EXPERIMENTS: GPCA VS PCA

We compare PCA and GPCA in terms of discrete reconstruction quality on different categorical datasets: binarized MNIST LeCun et al. (2010), binarized FashionMNIST Xiao et al. (2017), Cityscapes Cordts et al. (2016), and the PROMOTER, MELANOMA and FLYBRAIN datasets. For further specifications on the datasets and experimental configurations, see Sections G.1 and J.0.1, which provide the corresponding references.

For PCA, we determine the latent dimension needed to retain at least 99% of the total variance, following the cumulative percentage of total variation criterion in (Jolliffe, 2002, Section 6.1.1). More concretely, each data point is represented by nc one-hot coordinates, and its empirical covariance matrix has size $nc \times nc$. Let $\ell_1 \geq \cdots \geq \ell_{nc} \geq 0$ denote its eigenvalues, where ℓ_k is the variance of the corresponding principal component. The ratio of the sum of the first d eigenvalues to the sum of all eigenvalues gives the cumulative proportion of total variance explained by these components. We write

$$d_{99} := \min \left\{ d \in \{1, \ldots, nc\} : \frac{\sum_{k=1}^{d} \ell_k}{\sum_{k=1}^{nc} \ell_k} \geq 0.99 \right\}. \tag{69}$$

Table 4 compares the discrete reconstruction errors of PCA and GPCA. The first group of columns reports d_{99} and the mean Hamming distance equation 62, between the original data and their discrete PCA reconstructions at this dimension (PCA MHD). The second group compares PCA and GPCA at the same latent dimension d, with PCA MHD and GPCA MHD denoting their respective reconstruction errors. All GPCA results use $\lambda = 10^{-4}$.

The MHD measures the mean fraction of mislabeled nodes after rounding or categorical label selection. Thus, an MHD with value x for a dataset with n nodes per sample corresponds to xn mislabeled nodes per sample on average. For instance, for the Melanoma dataset with $n = 500$, the reconstruction errors at $d = 500$ correspond to an average of 101.5 mislabeled nodes per sample for PCA and 0.0015 for GPCA.

Table 4: PCA and GPCA reconstruction errors. The comparison at fixed d uses the same latent dimension for both methods. GPCA uses $\lambda = 10^{-4}$.

		PCA: 99% explained variance		**Mean Hamming distance**		
Dataset	$n \times c$	d_{99}	**PCA MHD**	d	**PCA MHD**	**GPCA MHD**
Binarized MNIST	1024×2	395	$3.51 \cdot 10^{-4}$	128	$3.37 \cdot 10^{-3}$	$47 \cdot 10^{-6}$
Binarized FashionMNIST	1024×2	594	$5.29 \cdot 10^{-4}$	256	$7.06 \cdot 10^{-3}$	$6 \cdot 10^{-6}$
Cityscapes	8192×8	2515	$5 \cdot 10^{-6}$	512	$2.66 \cdot 10^{-2}$	0
Promoter	1024×4	3017	$3.2 \cdot 10^{-5}$	1024	$2.05 \cdot 10^{-1}$	$41 \cdot 10^{-6}$
Melanoma	500×4	1475	0	500	$2.03 \cdot 10^{-1}$	$3 \cdot 10^{-6}$
Flybrain	500×4	1477	0	500	$2.36 \cdot 10^{-1}$	0

At equal latent dimensions, GPCA achieves smaller Hamming reconstruction errors than PCA on all six datasets. These results indicate that GPCA provides more accurate discrete reconstructions than PCA at the same level of compression. The differences are particularly pronounced for the biological datasets: at the selected latent dimensions, PCA misclassifies approximately 20–24% of nodes, whereas GPCA achieves nearly error-free discrete reconstructions. GPCA achieves these errors with substantially fewer latent dimensions than PCA requires to retain 99% of the total variance.

F EXPERIMENTS: COMPUTATIONAL COSTS AND TIME COMPLEXITY

We briefly discuss the computational cost of our method. We report wall-clock runtimes separately for the GPCA optimization and the subsequent flow-matching procedure. All experiments were conducted on a single NVIDIA GeForce RTX 4090 GPU with 24 GB of VRAM. Peak GPU memory usage was approximately 15 GB. We use PyTorch ≥ 2.10 with a CUDA 12.8.

For the optimization of the objective in equation 13, we employ full-batch Adam and compute gradients via PyTorch automatic differentiation. The regularization weight λ varies across experimental settings, as specified in the corresponding results. The resulting runtimes for the datasets used in our experiments are summarized in Table 5. Empirically, we find that only a moderate number of iterations, approximately 5000 in most cases, is required to attain values close to the optima reported in Tables 6–18. The optimization dynamics are stable, with gradient norms decreasing quickly and smoothly, as illustrated in Figures 11 and 15. Since this optimization is not computationally demanding, we run it for 30000 iterations in our experiments to ensure convergence.

We refer to Section G.1 for details on the datasets used, and to Avdeyev et al. (2023); Stark et al. (2024) for a description of the configurations for the PROMOTER and ENHANCER DNA datasets.

Table 5: Estimated GPCA optimization speed and wall-clock time for 30000 iterations.

Dataset	**iter/s**	**Time for 30000 iterations**
MNIST, $d = 64$	39.88	12m 32s
Fashion-MNIST, $d = 64$	39.85	12m 33s
Cityscapes, $d = 512$	27.52	18m 10s
Synthetic Large-Class datasets, $d = 16$	420.98	1m 11s
MELANOMA dataset, $d = 1024$	46.02	10m 52s
FLYBRAIN dataset, $d = 1024$	44.60	11m 12s
PROMOTER dataset, $d = 1024$	10.54	47m 26s

Regarding the flow-matching procedure, we emphasize that the computational cost depends on the chosen architecture. In particular, there is an important distinction between minimizing the objective in equation 16 and minimizing the reduced objective in equation 19. While learning directly in the data space may lead to faster convergence in terms of the number of epochs, it incurs a substantially higher computational cost per iteration. By contrast, when working in the latent coordinate space $\mathcal{Z}$, the computational cost is significantly reduced.

As an illustrative example, we trained on MNIST with latent dimension $d = 64$, using an MLP with approximately 8M parameters in the latent coefficient space $\mathcal{Z}$. Both the latent MLP and the data-space U-Net were trained for 600 epochs with batch size 256, using Adam with an initial learning rate of 10^{-4} and a cosine-annealing learning-rate schedule. The MLP uses eight layers, hidden dimension 512, and an expansion ratio of 4. The U-Net uses 32 base channels, channel multipliers $(1, 2, 2, 2)$, two residual blocks per level, and attention at spatial resolution 16. Including evaluation, the corresponding wall-clock runtimes were approximately 11 minutes for the latent MLP and around 1 hour for the data-space U-Net. Despite its larger parameter count, the MLP benefits from the reduced objective in equation 19, which expresses flow matching entirely in the latent coordinates z. In this formulation, the vector field $v_t(\cdot; w)$ acts on d-dimensional inputs rather than on the full ambient representation. Consequently, the forward and backward passes operate on compact latent representations, contributing to the observed reduction in computational cost. The learned transport dynamics are thus modeled directly in the reduced coordinate space $\mathcal{Z}$, with the decoder applied when mapping the resulting latent samples back to the data manifold. For sampling, we integrate the learned vector field using the adaptive Heun solver over $t \in [0, 1]$, with absolute and relative tolerances both set to 10^{-4}.

G EXPERIMENTS: EMPIRICAL PERFORMANCE OF GEOMETRIC PCA

G.1 DATASETS AND GPCA CONFIGURATION

We compare our approach, which minimizes the regularized GPCA objective in equation 13, with NLL minimization in equation 61 on one-hot data and cross-entropy minimization on smoothed data. These objectives and data representations are described in Appendix D. The experiments are conducted on the following datasets:

1. **Synthetic data with many classes and few nodes.** We consider four synthetic datasets, each represented by a histogram of a discrete distribution on $n = 2$ variables with $c = 92$ categories. For each dataset, we draw $N = 3 \cdot 10^4$ samples and then apply GPCA, as described in Section 2.2.
2. **MNIST and Fashion-MNIST.** We use MNIST LeCun et al. (2010) and Fashion-MNIST Xiao et al. (2017), each with $N = 6 \cdot 10^4$ training samples. The images are padded to $n = 32 \cdot 32$ pixels, and we use $c = 2$ classes. Since the pixel values lie in $[0, 1]$, we apply a rounding transformation to obtain binary data. For MNIST, this transformation has only a minor visual effect, and the digits remain clearly recognizable. For Fashion-MNIST, however, it provides another valuable benchmark for generating *discrete* data, cf. in Figure 10.
3. **Cityscapes.** We use Cityscapes Cordts et al. (2016), consisting of $N = 2975$ training samples. The images are resized to $n = 256 \cdot 256$ nodes, and we use $c = 8$ semantic classes.



(a) Original Fashion-MNIST samples. (b) Samples of binarized Fashion-MNIST.

Figure 10: Visual comparison of Fashion-MNIST samples before and after rounding. We included the binarized Fashion-MNIST data into our set of benchmark experiments.

For the GPCA optimization, the latent codes $Z \in \mathbb{R}^{N \times d}$ and basis parameters $V \in \mathbb{R}^{n \times (c-1) \times d}$ are learned jointly using full-batch Adam with learning rate 10^{-2} for between 10^4 and $3 \cdot 10^4$ iterations, depending on the experiment. For cross-entropy minimization and the regularized GPCA objective, we apply the smoothing transformation described in Appendix D with $\epsilon = 10^{-2}$. Gradients with respect to both Z and V are computed by automatic differentiation using PyTorch.

G.2 NEGATIVE LOG-LIKELIHOOD & KL MINIMIZATION

We first illustrate the behavior of the GPCA loss of Collins et al. (2001) on one-hot data and on data smoothed using equation 63 with $\epsilon = 10^{-2}$. As discussed in Appendix D, the objective equals the NLL in equation 61 for one-hot data. For smoothed data, we minimize the corresponding KL

divergence, equivalently the cross-entropy loss. We observe that NLL minimization leads to stable gradients and a consistent decrease in Hamming distance, with zero Hamming distance achieved at relatively small latent dimensions.

Figure 11 shows an optimization example on Cityscapes with latent dimension $d = 256$ over the first 450 iterations. We compare the NLL, the Hamming distance, and the Riemannian distance, also referred to as the e-distance. We observe that both the NLL and the Hamming distance decrease consistently and quickly reach very small values, whereas the e-distance remains comparatively high.

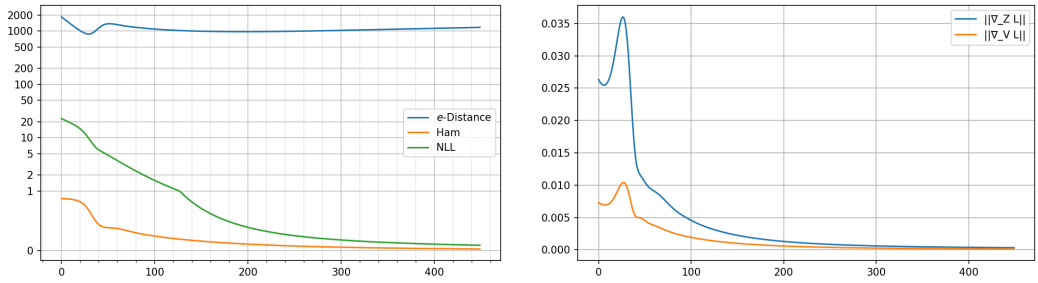


Figure 11: GPCA optimization on resized Cityscapes dataset with latent dimension $d = 256$. The left plot shows the evolution of the NLL, Hamming distance, and e-distance. The right plot shows the gradient norms during optimization.

Figure 12 shows the final mean Hamming distance as a function of the latent dimension after GPCA training for 10^4 iterations. The two curves correspond to Cityscapes and MNIST. Interestingly, zero Hamming distance on the entire training set is achieved with only $d = 44$ latent dimensions for MNIST and $d = 133$ latent dimensions for Cityscapes.

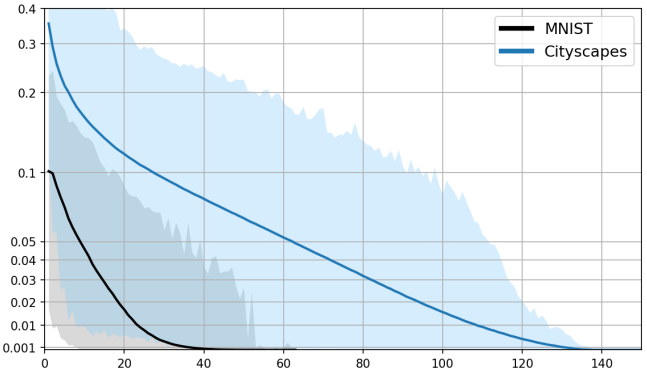


Figure 12: Hamming reconstruction error across latent dimension. The y-axis uses inverse-sinh scaling $y' = \sinh^{-1}(y/s)$ to emphasize small errors, with $s = 0.05$.

Figures 13 and 14 provide additional visualizations of the GPCA representation obtained on MNIST. The visual comparisons in Appendix G.4 show that a latent dimension of $d = 32$ already captures the main structure of the MNIST digits in the GPCA reconstructions. Figure 13 shows a two-dimensional embedding of the latent points obtained from a $d = 32$ dimensional GPCA subspace. The embedding is computed purely from the geometry of the latent representation and reveals a meaningful organization of the data without using label information. Finally, Figure 14 shows a three-dimensional version of the latent embedding, providing a more detailed view of the same organization.

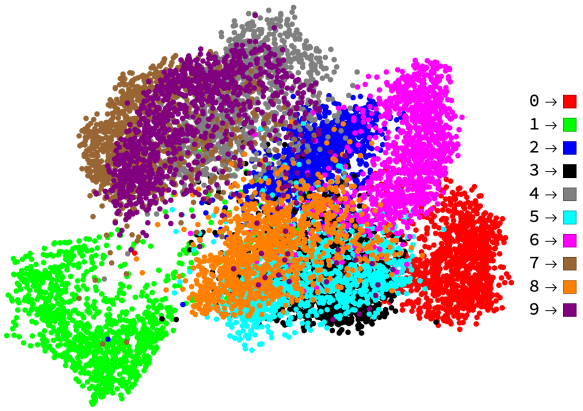


Figure 13: A 2D embedding of the $d = 32$ dimensional latent points $\widehat{\theta}_i = V \odot Z_i \in \mathcal{U}$, $i \in [N]$ corresponding to 10.000 samples $x_i \approx \varphi^{-1}(\widehat{\theta}_i)$ of the MNIST data base. This semantically meaningful and *fully unsupervised* embedding of the 4-nearest neighbor graph with nodes $\widehat{\theta}_i$ is *purely geometric*: no learning is involved besides determining the GPCA subspace $\mathcal{U}$. This result underlines the suitability of latent GPCA representations of discrete data, as illustrated by Figure 2.

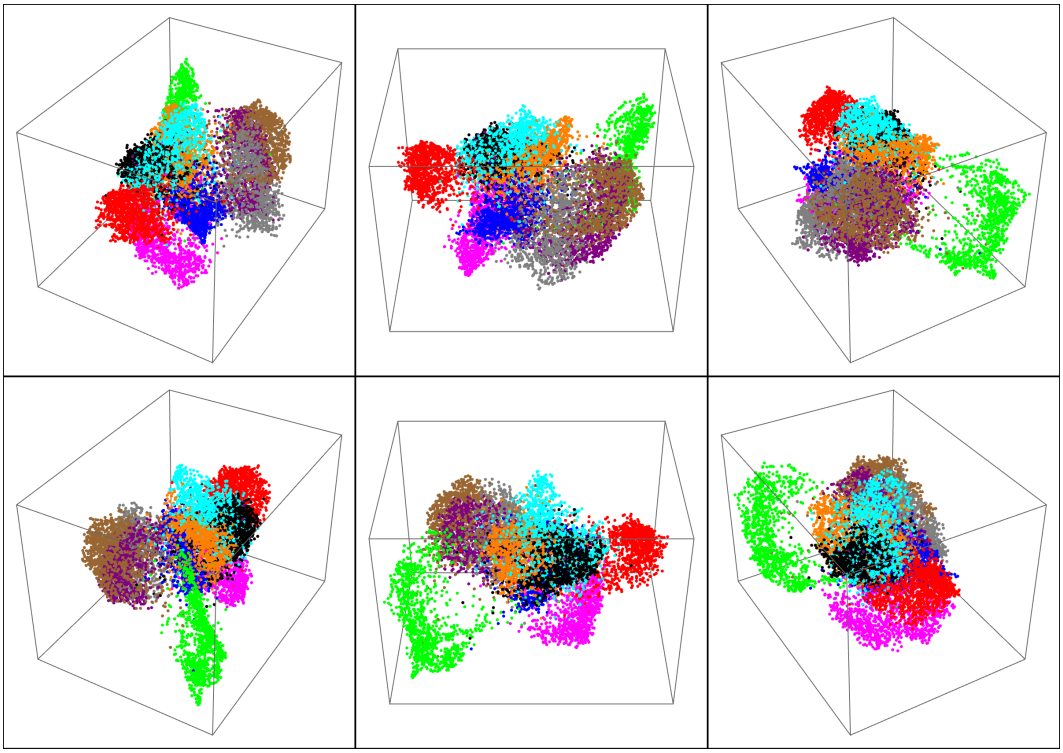


Figure 14: A more detailed 3D embedding of the $d = 32$-dimensional latent data points representing a corresponding sample from the MNIST data base.

Tables 6 and 7 show how the Hamming distance in equation 62 decreases together with the NLL in equation 61 for different latent dimensions on MNIST and Cityscapes. For example, on Cityscapes, even a strongly reduced latent dimension $d = 256$, corresponding to a reduction factor of 512, is sufficient for the NLL to become small and for the Hamming distance to drop to zero within seconds, after only a few iterations. The e-distance in equation 68, however, does not decrease as consistently and remains on the order of 10^3 when optimizing the NLL.

By contrast, Tables 8 and 9 show that optimizing the KL divergence substantially reduces the e-distance, although the resulting values may still remain relatively large. Figure 15 illustrates this behavior for GPCA optimization on Cityscapes with latent dimension $d = 256$. As in Figure 11, the loss and the Hamming distance decrease at a similar rate. However, when optimizing the KL divergence, the e-distance is considerably smaller, while the final Hamming distance is larger than in the NLL case. The right panel shows that the gradient norms decay stably throughout optimization.

Table 6: **NLL minimization.** Comparison of the Hamming distance, NLL, and the squared e-distance for the **MNIST** dataset across different latent dimensions.

	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
NLL $(\times 10^{-6})$	110123	57950	3803	0.0	0.0
MHD $(\times 10^{-6})$	54501	26023	1352	0.0	0.0
e-Distance $(\times 10^{3})$	73.333	171.252	168.111	3.817	1.713

Table 7: **NLL minimization.** Comparison of the Hamming distance, NLL, and the squared e-distance for the **Cityscapes** dataset across different latent dimensions.

	$d = 32$	$d = 64$	$d = 128$	$d = 256$	$d = 512$
NLL $(\times 10^{-6})$	254804	123109	4	0.0	0.0
MHD $(\times 10^{-6})$	88305	42746	0.0	0.0	0.0
e-Distance $(\times 10^{3})$	493.297	1192.090	212.277	8.332	6.593

Table 8: **Cross-entropy minimization.** Comparison of the Hamming distance, cross-entropy, and the squared e-distance for the **MNIST** dataset across different latent dimensions.

	$d = 8$	$d = 16$	$d = 32$	$d = 64$	$d = 128$
Cross-Entropy $(\times 10^{-6})$	143713	102126	63448	44578	36420
MHD $(\times 10^{-6})$	55357	29450	7705	746	37
e-Distance	4.670	8.117	14.454	7.647	2.550

Table 9: **Cross-entropy minimization.** Comparison of the Hamming distance, cross-entropy, and the squared e-distance for the **Cityscapes** dataset across different latent dimensions.

	$d = 32$	$d = 64$	$d = 128$	$d = 256$	$d = 512$
Cross-Entropy $(\times 10^{-6})$	353188	251113	150246	92236	76277
MHD $(\times 10^{-6})$	96587	57206	12970	0.0	0.0
e-Distance	91.55	118.65	157.32	58.23	17.33

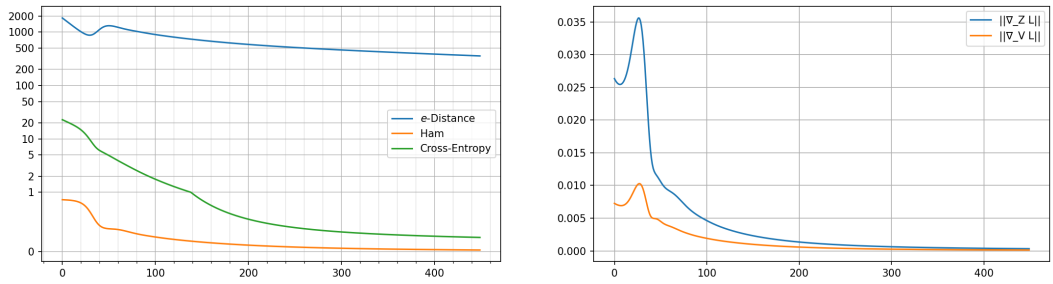


Figure 15: GPCA optimization on resized Cityscapes dataset with latent dimension $d = 256$. The left panel shows the evolution of the KL divergence, Hamming distance, and e-distance. The right panel shows the gradient norms during optimization.

G.3 GPCA OBJECTIVE WITH e-DISTANCE REGULARIZATION

Proposition 3.5 states that, on a fixed compact set of the assignment manifold $\mathcal{S}_c^n$, the KL divergence controls the squared e-distance between data points and their reconstructions.Consequently, along the GPCA optimization, decreasing the KL divergence reduces the upper bound on the squared e-distance, up to constants depending on a fixed compact set satisfying the assumptions of Proposition 3.5, provided the data and their reconstructions remain in this set. To prevent the reconstructions from approaching the boundary of the simplex too closely, we propose to regularize the GPCA objective by adding an explicit e-distance penalty and minimize the objective in equation 13 which reads

$$\mathcal{L}_\lambda(V,Z)=\frac{1}{Nn}\sum_{i=1}^{N}\sum_{j=1}^{n} D_{\mathrm{KL}}(x_{ij},\varphi^{-1}((V\odot Z_i)_j))+\lambda\,\mathrm{d}_e^2(x_{ij},\varphi^{-1}((V\odot Z_i)_j)).$$

Here, the second term in equation 13 penalizes reconstructions that are far from the data in e-distance, encouraging agreement with the smoothed target probabilities. The parameter $\lambda > 0$ balances the contribution of the regularization term relative to the KL term.

Empirically, we observe that smaller λ tend to yield lower Hamming distances, whereas larger values enforce stronger control of the e-distance. Hence, the main trade-off is between achieving highly accurate discrete reconstructions, as measured by the Hamming distance, and maintaining sufficiently small e-distance. This behavior is illustrated in Tables 17, and 18. For example, choosing $\lambda = 10^{-3}$ appears to provide a reasonable compromise for the MNIST dataset with latent dimension $d = 128$. It yields a Hamming distance of $\mathrm{MHD} = 9.6 \cdot 10^{-5}$ together with a small squared e-distance. Interpreted per image, this corresponds on average to about 0.1 misclassified pixels. We also observe that, for fixed λ, increasing the latent dimension decreases the GPCA loss, the Hamming distance, and the e-distance. This is consistent with the fact that a higher-dimensional representation allows for sharper and more accurate reconstructions, cf. Tables 10– 16.

We also considered an alternative regularized objective in which the e-distance penalty is replaced by a negative entropy penalty on the reconstructions. Empirically, this leads to qualitatively similar behavior, i.e. the entropy penalty prevents reconstructions from becoming too close to the corners of the simplex. In addition, smaller regularization weights tend to improve the Hamming distance, whereas larger weights provide stronger control of the e-distance. Overall, the results are comparable to those obtained with the e-distance penalty, so this modification does not lead to substantial changes in performance.

Table 10: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions $\boldsymbol{d}$ on the MNIST dataset with regularization parameter $\boldsymbol{\lambda = 10^{-2}}$

	$\boldsymbol{d=8}$	$\boldsymbol{d=16}$	$\boldsymbol{d=32}$	$\boldsymbol{d=64}$	$\boldsymbol{d=128}$	$\boldsymbol{d=256}$
Cross-Entropy $(\times\mathbf{10^{-6}})$	173149	129212	89401	62162	45868	36243
MHD $(\times\mathbf{10^{-6}})$	56267	33396	14260	3604	495	97
e-Distance	2.724	2.206	1.699	1.147	0.689	0.268589

Table 11: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the MNIST dataset with regularization parameter $\lambda = 10^{-3}$

	$d=8$	$d=16$	$d=32$	$d=64$	$d=128$	$d=256$
Cross-Entropy $(\times 10^{-6})$	173149	129212	89401	62162	38158	36243
MHD $(\times 10^{-6})$	56267	33396	14260	3602	96	0
e-Distance	2.72	2.21	1.70	1.48	1.38	0.27

Table 12: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the Fashion-MNIST dataset with regularization parameter $\lambda = 10^{-2}$

	$d=16$	$d=32$	$d=64$	$d=128$
Cross-Entropy $(\times 10^{-6})$	159478	128362	99688	71593
MHD $(\times 10^{-6})$	44509	31548	19176	6812
e-Distance	2.771	2.292	1.855	1.417

Table 13: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the Fashion-MNIST dataset with regularization parameter $\lambda = 10^{-3}$

	$d=16$	$d=32$	$d=64$	$d=128$
Cross-Entropy $(\times 10^{-6})$	132078	104965	79654	54938
MHD $(\times 10^{-6})$	41995	28759	15590	3128
e-Distance	3.964	3.647	3.543	3.424

Table 14: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the **Cityscapes** dataset with regularization parameter $\lambda=10^{-1}$

	$d=32$	$d=64$	$d=128$	$d=256$	$d=512$	$d=1024$
Cross-Entropy $(\times 10^{-6})$	1405488	1162139	928690	698497	477215	283546
MHD $(\times 10^{-6})$	125910	98440	71155	41466	13118	752
e-Distance	9.556	7.990	6.530	5.067	3.548	2.007

Table 15: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the **Cityscapes** dataset with regularization parameter $\lambda=10^{-2}$

	$d=32$	$d=64$	$d=128$	$d=256$	$d=512$	$d=1024$
Cross-Entropy $(\times 10^{-6})$	510419	404977	301960	207198	137594	96735
MHD $(\times 10^{-6})$	112242	80375	46083	11937	128	0.00
e-Distance	11.619	10.311	8.97	7.24	4.65	2.25

Table 16: Cross-entropy, Hamming distance and squared e-distance for different latent dimensions d on the **Cityscapes** dataset with regularization parameter $\lambda=10^{-3}$

	$d=32$	$d=64$	$d=128$	$d=256$	$d=512$	$d=1024$
Cross-Entropy $(\times 10^{-6})$	385214	287447	192681	119147	86937	74136
MHD $(\times 10^{-6})$	102815	66227	26065	412	0.00	0.00
e-Distance	23.837	22.684	21.705	18.204	8.418	3.286

Table 17: Cross-entropy, Hamming distance and squared e-distance for different values of the regularization parameter $\boldsymbol{\lambda}$ on the MNIST dataset with latent dimension $\boldsymbol{d = 128}$

	$\boldsymbol{\lambda = 1}$	$\boldsymbol{\lambda = 0.1}$	$\boldsymbol{\lambda = 0.01}$	$\boldsymbol{\lambda = 0.001}$	$\boldsymbol{\lambda = 0}$
Cross-Entropy $(\times \mathbf{10^{-6}})$	586701	97695	45868	38158	36420
MHD $(\times \mathbf{10^{-6}})$	3122	1841	495	96	37
e-Distance	0.54	0.55	0.69	1.38	2.55

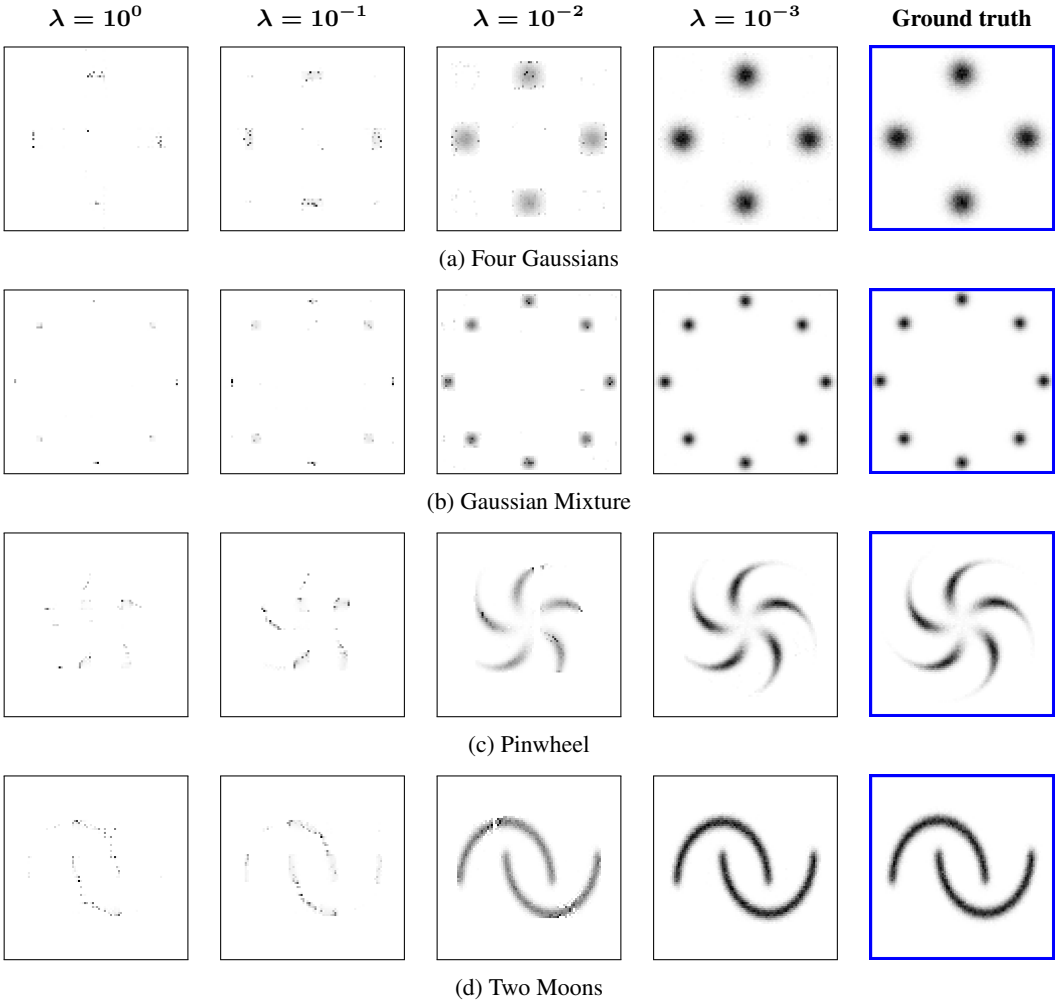
Table 18: Cross-entropy, Hamming distance and squared e-distance for different values of the regularization parameter $\boldsymbol{\lambda}$ on the **Cityscapes** dataset with latent dimension $\boldsymbol{d}$**=256**

	$\boldsymbol{\lambda}$**=1**	$\boldsymbol{\lambda}$**=0.1**	$\boldsymbol{\lambda}$**=0.01**	$\boldsymbol{\lambda}$**=0.001**	$\boldsymbol{\lambda}$**=0**
Cross-Entropy $(\times \mathbf{10^{-6}})$	5133136	698159	207198	119147	99236
MHD $(\times \mathbf{10^{-6}})$	143927	41458	11937	412	00
e-Distance	4.91	5.06	7.24	18.20	58.23

G.4 VISUAL COMPARISON OF GPCA RECONSTRUCTIONS

We further illustrate the effect of the regularization parameter λ on the GPCA reconstructions by visualizing the approximations $\widehat{p}_i$ for different values of λ and different latent dimensions d.

Synthetic large-class datasets. We consider four synthetic datasets, each represented by a histogram of a discrete distribution on $n = 2$ variables with $c = 92$ categories. From each dataset, we sample $N = 3 \cdot 10^4$ points and then apply GPCA, see Section 2.2. This yields a low-dimensional representation, which is subsequently used to train our model via Flow Matching, as described in Section 4.



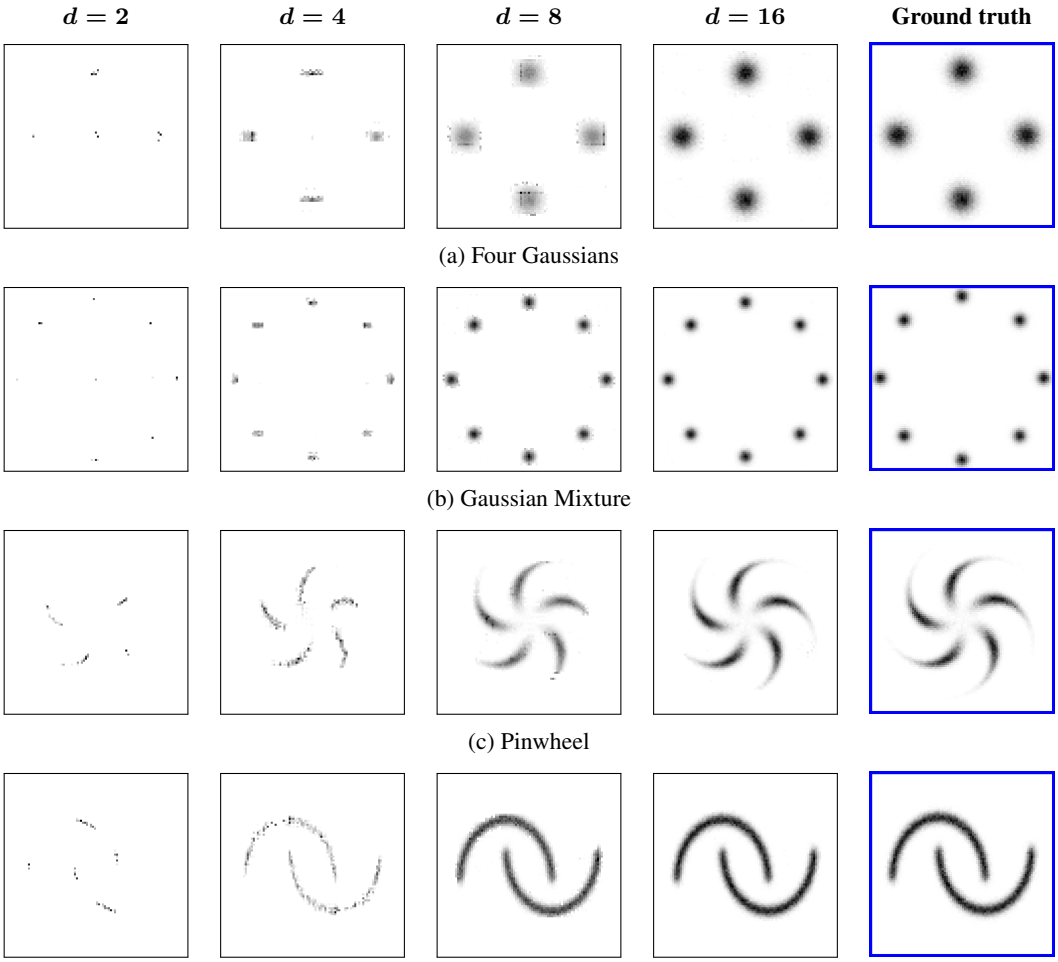
(a) Four Gaussians

(b) Gaussian Mixture

(c) Pinwheel

(d) Two Moons

Figure 16: Visual comparison of GPCA reconstructions of synthetic datasets, with $c = 92$ categories, and regularization parameters λ, cf. Eq. equation 13, with **fixed latent dimension $d = 16$**. Each row corresponds to one dataset, and each column shows the empirical joint distribution estimated from $N = 3 \cdot 10^4$ samples using $\lambda \in \{10^0, 10^{-1}, 10^{-2}, 10^{-3}\}$. The rightmost image, highlighted in blue, shows the empirical joint distribution estimated from the ground-truth distribution.



(a) Four Gaussians

(b) Gaussian Mixture

(c) Pinwheel

(d) Two Moons

Figure 17: Visual comparison of GPCA reconstructions of synthetic datasets, with $c = 92$ categories, and latent dimensions. We use overall a **fixed $\boldsymbol{\lambda} = \mathbf{10^{-3}}$** for the GPCA optimization, cf. Eq. equation 13. Each row corresponds to one dataset, and each column shows the empirical joint distribution estimated from $N = 3 \cdot 10^4$ samples using latent dimension $d \in \{2, 4, 8, 16\}$. The rightmost image, highlighted in blue, shows the empirical joint distribution estimated from the ground-truth distribution.

Cityscapes Dataset. We illustrate the effect of the regularization parameter λ and the latent dimension d on the GPCA reconstructions of the Cityscapes dataset. Figures 18–20 correspond to selected values from Tables 14–16. In particular, for $\lambda = 10^{-3}$ and $d = 256$, we observe almost perfect reconstructions. This is consistent with the very low Hamming distance of $412 \cdot 10^{-6}$ reported in Table 16, which corresponds, on average, to about 27 mislabeled pixels per image.

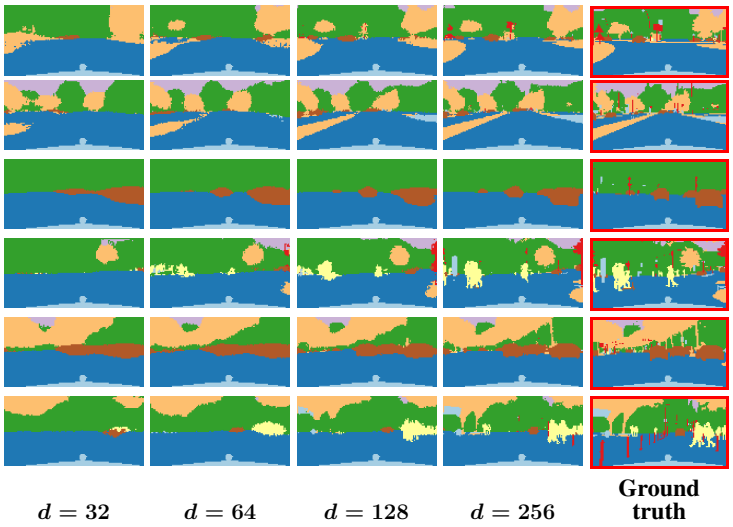


Figure 18: Visual comparison of Cityscapes GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{32, 64, 128, 256\}$ for **fixed regularization parameter $\boldsymbol{\lambda = 10^{-1}}$**. The rightmost column, highlighted in red, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

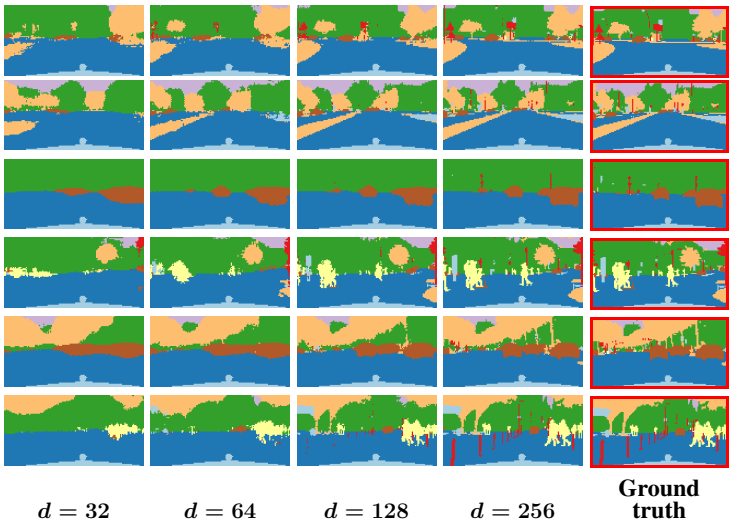


Figure 19: Visual comparison of Cityscapes GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{32, 64, 128, 256\}$ for **fixed regularization parameter $\boldsymbol{\lambda = 10^{-2}}$**. The rightmost column, highlighted in red, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

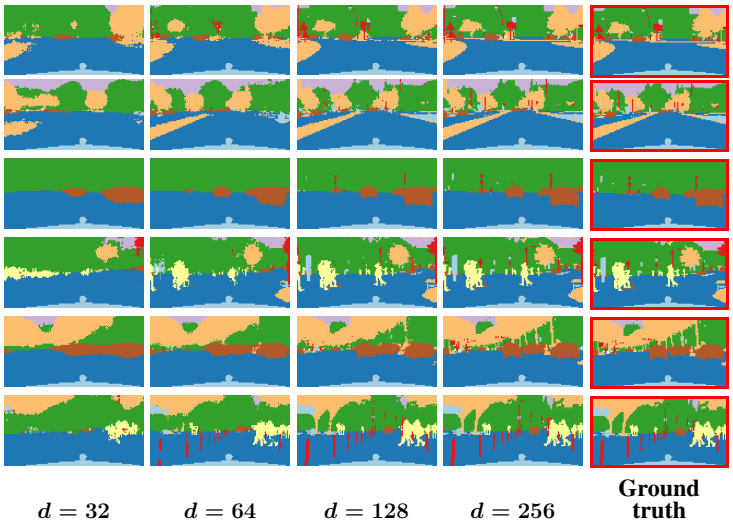
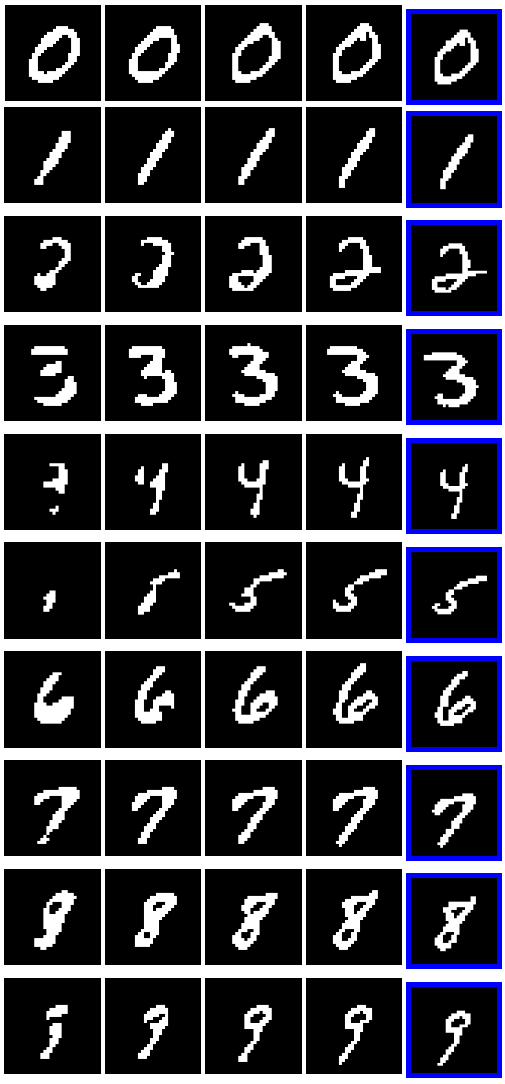


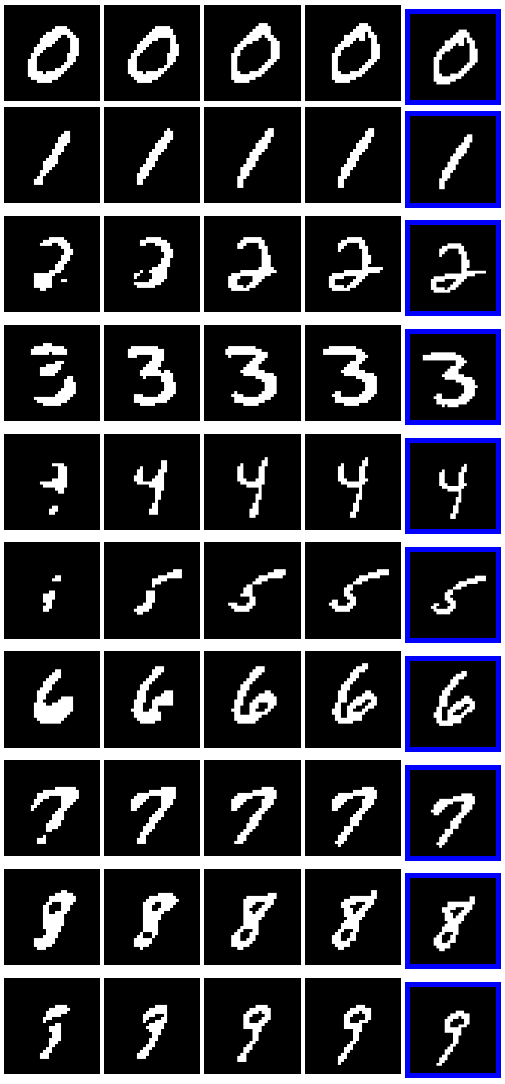
Figure 20: Visual comparison of Cityscapes GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{32, 64, 128, 256\}$ for **fixed regularization parameter $\boldsymbol{\lambda = 10^{-3}}$**. The rightmost column, highlighted in red, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

MNIST and Fashion-MNIST Datasets. We illustrate the effects of the regularization parameter λ and the latent dimension d on the GPCA reconstructions of the MNIST and Fashion-MNIST datasets. Figures 21 and 22 correspond to selected values from Tables 10 and 11, respectively. In particular, we observe that, for both $\lambda = 10^{-2}$ and $\lambda = 10^{-3}$, the reconstructions capture the overall structure of the data for dimensions $d = 32$ and $d = 64$. These are the latent dimensions used to train our flow matching model, which yielded good performance, cf. Section H. Figures 23 and 24 correspond to selected values from Tables 12 and 13, respectively.



$d = 8$ $d = 16$ $d = 32$ $d = 64$ **Ground truth**

Figure 21: Visual comparison of MNIST GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{8, 16, 32, 64\}$ for **fixed regularization parameter $\lambda = 10^{-2}$**. The rightmost column, highlighted in blue, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.



$d = 8$ $d = 16$ $d = 32$ $d = 64$ **Ground truth**

Figure 22: Visual comparison of MNIST GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{8, 16, 32, 64\}$ for **fixed regularization parameter $\lambda = 10^{-3}$**. The rightmost column, highlighted in blue, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

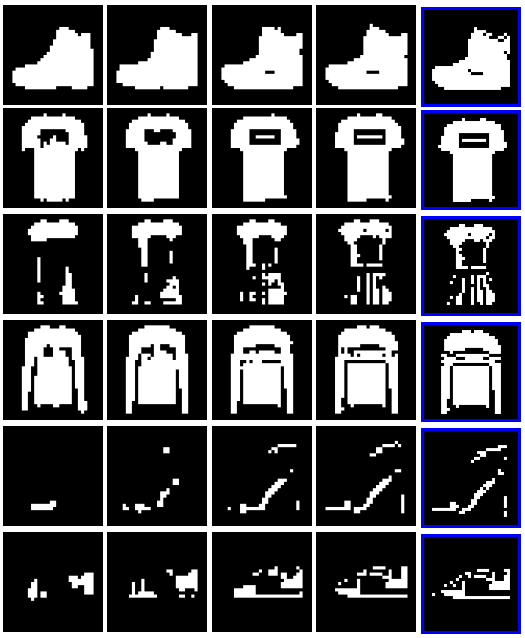


Figure 23: Visual comparison of Fashion-MNIST GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{16, 32, 64, 128\}$ for **fixed regularization parameter $\lambda = 10^{-2}$**. The rightmost column, highlighted in blue, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

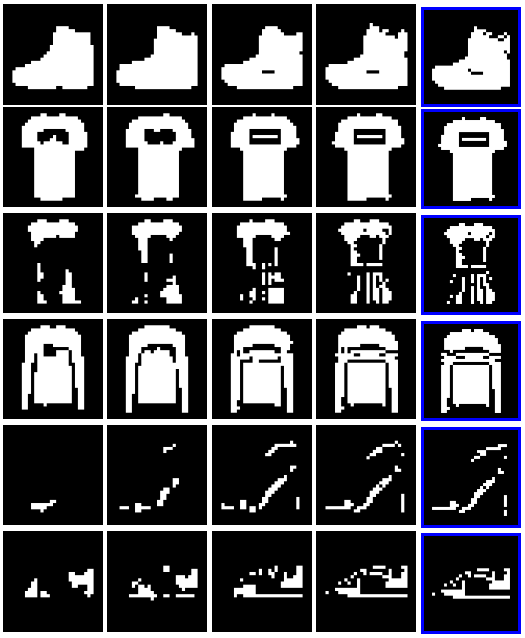


Figure 24: Visual comparison of Fashion-MNIST GPCA reconstructions for randomly selected training samples. Each column corresponds to one latent dimension $d \in \{16, 32, 64, 128\}$ for **fixed regularization parameter $\lambda = 10^{-3}$**. The rightmost column, highlighted in blue, shows the corresponding ground-truth data $x \in \mathcal{X}_N^n$.

H EXPERIMENTS: GENERATION QUALITY FOR MNIST WITH BINARY INPUTS

We evaluate unconditional samples produced by models trained on the MNIST handwritten-digit dataset LeCun et al. (2010) with binary inputs. To study the effect of latent dimensionality, we vary $d \in \{8, 16, 32, 64, 128\}$ while fixing the regularization weight to $\lambda = 10^{-4}$. We also vary $\lambda \in \{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ at fixed latent dimension $d = 32$. In both experiments, we assess the quality and diversity of generated samples using a pretrained classifier.

For each latent dimension d, we generate M unconditional samples $w_1, \ldots, w_M$ and compute the mean classifier confidence

$$Q(d) = \frac{1}{M} \sum_{i=1}^{M} \max_{k=0,\ldots,9} p_\phi(k \mid w_i), \tag{70}$$

where $p_\phi(k \mid w_i)$ denotes the predicted class probability assigned by the pretrained classifier. This quantity measures how confidently the generated samples are recognized as valid digits. Larger values of $Q(d)$ indicate that samples are, on average, more digit-like according to the classifier. To assess class diversity, we also compute the entropy of the empirical predicted class histogram

$$H(d) = -\sum_{k=0}^{9} \widehat{p}_d(k) \log \widehat{p}_d(k), \qquad \widehat{p}_d(k) = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}[\arg\max_c p_\phi(c \mid w_i) = k], \tag{71}$$

which measures the class coverage. The maximal class confidence is $Q(d) = 1$, the maximal class coverage is $H(d) = \log(10) \approx 2.3026$.

Table 19 reports the metrics for varying latent dimension d at fixed regularization weight $\lambda = 10^{-4}$.

Table 19: Classifier-based metrics for different latent dimensions d at fixed regularization weight $\lambda = 10^{-4}$.

	$d = 8$	$d = 16$	$d = 32$	$d = 64$	**Test data**
$Q(d)$	0.831	0.944	0.982	0.981	0.991
$H(d)$	2.224	2.293	2.301	2.301	2.301

For fixed $d = 32$, Table 20 reports the classifier confidence and class-histogram entropy as λ varies. The highest confidence is obtained for $\lambda = 10^{-4}$, while class-histogram entropy remains close to the test-data reference across all settings. We use Q and H for the same metrics defined above, evaluated for each regularization weight.

Table 20: Classifier-based metrics for varying regularization weight λ at fixed latent dimension $d = 32$. The test-data reference is evaluated using the same classifier.

	$\lambda = 0$	$\lambda = 10^{-5}$	$\lambda = 10^{-4}$	$\lambda = 10^{-3}$	$\lambda = 10^{-2}$	$\lambda = 10^{-1}$	**Test data**
$Q(d)$	0.972	0.978	0.982	0.981	0.978	0.969	0.991
$H(d)$	2.300	2.300	2.301	2.301	2.301	2.299	2.301

I EXPERIMENTS: GENERATION OF NOVEL DATA POINTS

To assess whether the generative model produces novel samples rather than reproducing training examples, we compare randomly generated outputs with their nearest images from the training set. Models are trained with regularization parameter $\lambda = 10^{-2}$ and latent dimensions 32, 64, and 512 for MNIST, FashionMNIST, and Cityscapes, respectively. For each generated sample, we retrieve the four nearest training images using the Hamming distance in equation 62 and display them alongside the generated sample. The results indicate that the generated samples are not direct copies of training images, suggesting that the model does not simply memorize the training set.

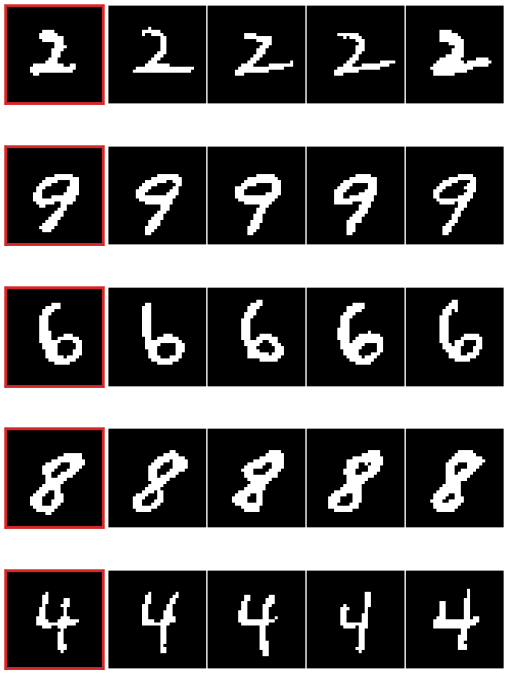


Figure 25: **MNIST.** Generated samples (left, marked red) and their four nearest training images, with $d = 32$ and $\lambda = 10^{-2}$.

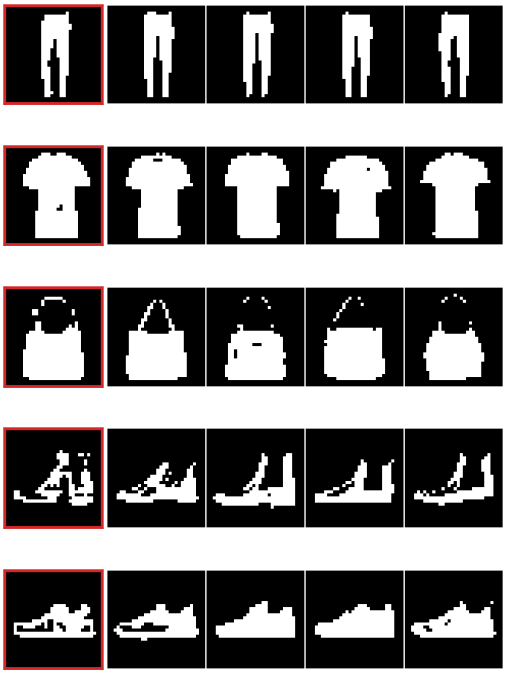


Figure 26: **FashionMNIST.** Generated samples (left, marked red) and their four nearest training images, with $d = 64$ and $\lambda = 10^{-2}$.

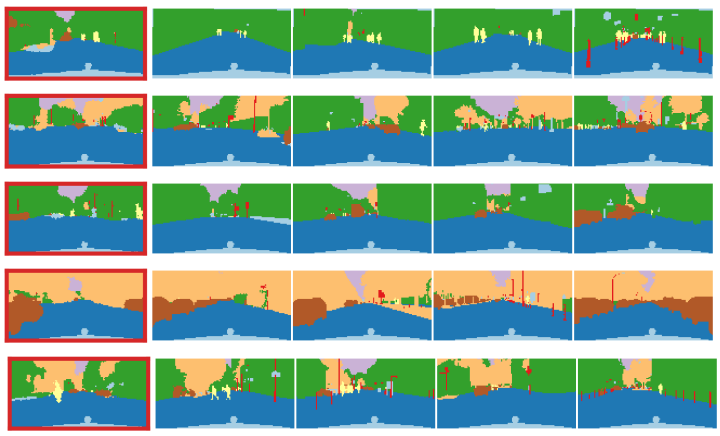


Figure 27: **Cityscapes.** Generated samples (left, marked red) and their four nearest training images, with $d = 512$ and $\lambda = 10^{-2}$.

J EXPERIMENTS: CONDITIONAL FLOW MATCHING ON ENHANCER DNA DATASETS

J.0.1 GPCA RECONSTRUCTIONS ON DNA DATASETS

We consider three DNA datasets, with each nucleotide represented by one of $c = 4$ categories. The PROMOTER dataset contains 100000 human DNA sequences of length $n = 1024$, centered at annotated transcription start sites and paired with transcription initiation signal profiles. The MELANOMA and FLYBRAIN datasets contain 88870 and 104665 sequences of length $n = 500$ from human melanoma and fly brain cells, respectively. Further details on these datasets, preparation generation tasks see Avdeyev et al. (2023, Appendices C.4 and F) and Stark et al. (2024, Sections 4.2–4.3 and Appendices B.1–B.2).

A limitation of GPCA concerns the compressibility of the data. Overall, we observe that the image datasets considered in this work admit more accurate reconstructions than the DNA datasets when using the same latent dimension and regularization parameters, compare Tables 21, 22, and 23 below with the results in Section G.3.

A further limitation concerns extending our formulation of flow matching in the coefficient space $\mathcal{Z}$ to conditional generation. The conditioning information need not be categorical or share the same representation as the sequences, but may instead be continuous, as in the PROMOTER dataset. In this case, we learn GPCA from the sequences alone, while the conditioning profiles are subsequently used for conditional flow matching. We incorporate the conditioning information using network architectures employed in the $\mathcal{U}$ space, as such architectures are typically used by conditional generation models in practice. Consequently, we cannot directly take advantage of operating in $\mathcal{Z}$ and minimizing the conditional flow matching objective in equation 19. Developing a formulation that jointly incorporates the latent representation in $\mathcal{Z}$ and the conditioning information therefore remains a direction for future work.

Promoter dataset. We use $N = 88570$ training sequences to learn the GPCA representation. Table 21 reports results for $d = 256, 512, 1024$ using the regularization parameter $\lambda = 10^{-3}$ in equation 13. These dimensions correspond to compression factors of 16, 8, and 4, respectively, relative to the 4096 one-hot coordinates. We observe that accurate label reconstruction requires a relatively large latent dimension and does not by itself ensure a small geometric approximation error.

Table 21: GPCA reconstruction metrics for PROMOTER with $\lambda = 10^{-3}$ across latent dimensions.

	$d = 256$	$d = 512$	$d = 1024$
Cross-Entropy	1.0925	0.7977	0.2442
MHD	0.4583	0.2903	0.0041
e**-distance**	7.9725	9.0613	9.9992

Enhancer datasets. Tables 22 and 23 report GPCA reconstruction results for FLYBRAIN and MELANOMA at $d = 500$, varying the regularization parameter λ in equation 13. This dimension corresponds to a compression factor of 4 relative to the $4 \cdot 500 = 2000$ one-hot coordinates. Increasing λ reduces the e-distance but generally increases the reconstruction error, illustrating the trade-off between accurate label reconstruction and a small geometric approximation error mentioned in Section 6.

Table 22: GPCA reconstruction metrics for FLYBRAIN with $d = 500$.

	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Cross-Entropy	0.5643	0.4312	0.2507
MHD	0.1951	0.0100	0.0100
e**-distance**	3.8992	4.6061	9.5124

Table 23: GPCA reconstruction metrics for MELANOMA with $d = 500$.

	$\lambda = 0.1$	$\lambda = 0.01$	$\lambda = 0.001$
Cross-Entropy	0.5121	0.2927	0.2305
MHD	0.1610	0.0342	0.0100
e**-distance**	3.8571	6.5329	9.2930

J.0.2 CONDITIONAL FLOW MATCHING ON DNA DATASETS

Promoter dataset. We compare our method (GPCA + FM) with the promoter generation results reported by Cheng et al. (2024), using latent dimension $d = 1024$ and regularization parameter $\lambda = 10^{-3}$. Generation is conditioned on transcription initiation signal profiles. Table 24 reports SP-MSE, the mean squared difference between the promoter activities predicted by Sei Chen et al. (2022a) for generated sequences and the corresponding reference sequences. Lower values indicate closer agreement in predicted promoter activity. We use the experimental configurations described in Cheng et al. (2024, Appendix C.6), select the checkpoint with the lowest validation SP-MSE and report its test SP-MSE. Baseline values are taken from their Figure 4, with the Language Model and DirichletFM results attributed to Stark et al. (2024). Our method achieves a test SP-MSE of 0.0335, close to the value of 0.0315 obtained in our reproduction of SFM without OT. We note that we could not reproduce its published SP-MSE of 0.0258.

Table 24: SP-MSE on generated promoter DNA experiment. Baseline values are taken from Cheng et al. (2024), who report the entries marked † from Stark et al. (2024).

Model	**SP-MSE↓**
SFM w/ OT Cheng et al. (2024)	0.0279
SFM w/o OT Cheng et al. (2024)	**0.0258**
LinearFM Cheng et al. (2024)	0.0282
DDSM Avdeyev et al. (2023)	0.0334
D3PM-uniform Austin et al. (2021)	0.0375
Bit-Diffusion (one-hot) Chen et al. (2022b)	0.0395
Bit-Diffusion (bit) Chen et al. (2022b)	0.0414
Language Model	$0.0333^{\dagger}$
DirichletFM Stark et al. (2024)	$0.0269^{\dagger}$
SFM w/o OT (our reproduction)	0.0315
GPCA + FM (ours), $d = 1024,\ \lambda = 10^{-3}$	0.0335

Enhancer datasets. We compare our method (GPCA + FM) with the enhancer generation results reported by Davis et al. (2024) on the MELANOMA and FLYBRAIN datasets, using latent dimension $d = 500$ and regularization parameter $\lambda = 10^{-3}$. Training is unconditional. Table 25 reports generation quality measured by the Fréchet Biological Distance (FBD), with lower values indicating closer agreement between the distributions of generated and reference sequences in the feature space used for evaluation. We use the same neural network architectures and hyperparameters as Davis et al. (2024), performing flow matching in the natural-parameter subspace $\mathcal{U}$. Consequently, this experiment does not realize the computational savings from training directly in the latent coefficient space $\mathcal{Z}$.

For Fisher-Flow, we report the mean FBD obtained in our reproduction. On MELANOMA, our value of 28.4 is close to the published result of 27.5 ± 2.6 Davis et al. (2024). On FLYBRAIN, we identified a dataset-loading issue in the released experiment configuration, which selects the Melanoma dataset instead of FlyBrain. With the corrected configuration, we obtain an FBD of 16.9 and could not reproduce the published value of 3.8. This configuration issue may account for the discrepancy.

Table 25: Generation quality on enhancer DNA datasets measured by FBD. Published baseline results are taken from Davis et al. (2024).

Method	**Melanoma FBD↓**	**FlyBrain FBD↓**
Random Sequence	619.0 ± 0.8	832.4 ± 0.3
Language Model	35.4 ± 0.5	25.7 ± 1.0
DirichletFM Stark et al. (2024)	$\mathbf{7.3 \pm 1.2}$	6.8 ± 1.8
Fisher-Flow Davis et al. (2024)	27.5 ± 2.6	$\mathbf{3.8 \pm 0.3}$
Fisher-Flow (our reproduction)	28.4 ± 0.7	16.9 ± 0.5
GPCA + FM (ours)	29.6 ± 1.1	17.3 ± 0.4