LEARNING DIRECTIONAL RANDERS GEOMETRY FOR LATENT-SPACE EXPLORATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks extract powerful latent representations from images, enabling fast and efficient downstream analysis. However, these latent spaces lack explicit geometric structure, leading to possibly poor interpolation fidelity and misleading extrapolation. Latent interpolation techniques assume symmetric transition costs which are often not the case for real world data. Their symmetric nature cannot capture directional processes, such as disease progression, where certain transitions are impossible (*e.g.* tumor growth dynamics or cognitive improvement in Alzheimer's disease). Although Riemannian metrics extend Euclidean geometry by incorporating curvature, they remain symmetric. We show that understanding asymmetry and not curvature alone directly addresses these problems. To do so we propose a new Finsler-Randers metric learning framework. It incorporates direction-dependent distances, thus enabling the computation of trajectories respecting monotonic sequential constraints. We demonstrate its potential for interpolation and extrapolation on controlled cross-sectional datasets.

1 INTRODUCTION

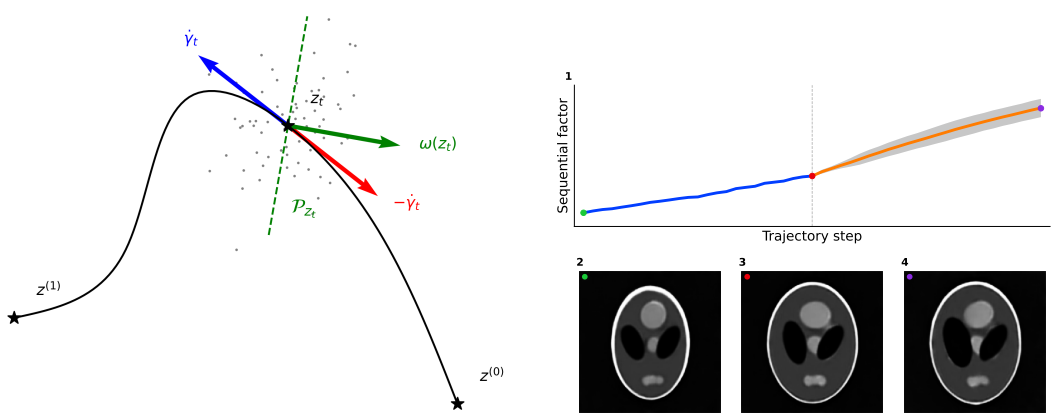
The progressive switch in image processing from hand-crafted features to features extracted from deep neural networks has significantly improved performance in various fields such as radiology (Astaraki et al., 2021), computational pathology (McGenity et al., 2024), image classification (Xian et al., 2017), and others. Once equipped with a decoder, latent space exploration serves as a proxy for exploration of the image space. In clinical settings, this approach is motivated by the analysis of patient over time, whether through interpolation or extrapolation of trajectories, or synthesis of longitudinal data. However, these gains come at the cost of interpretability. The latent spaces in which these features live are not naturally equipped with an explicit geometric structure: the assumption of Euclidean geometry falls short when exploring the latent space (Mi et al., 2021). While the lipschitzness of standards networks (Virmaux & Scaman, 2018) enables local understanding of latent neighborhoods, this local view is not enough to characterize the global structure needed for reliable long-range interpolations and extrapolation.

Working under the manifold hypothesis, a body of work around metric learning (Gruffaz & Sassen, 2025) aims to endow latent spaces with relevant geometric structures. Yet the symmetric nature of these frameworks, *i.e.* that $\text{dist}(a, b) = \text{dist}(b, a)$ and that the underlying trajectory is identical regardless of direction, fails to describe many real world situations. In clinical settings, some pathologies display monotonous evolution, e.g. cognitive scores on Alzheimer's disease. But, when considering the study of possible evolution, there are no guarantees that the extrapolated trajectories will respect this constraint. Clinical markers such as cognitive scores may vary non-monotonically along the computed path, which is violating a known monotonicity constraint. We use this setting to motivate our approach and validate it on controlled datasets exhibiting the same monotonicity structure.

In this work we propose to embed the dynamic of the data directly as a static learned geometry. This geometry needs to take into account asymmetry to inject prior knowledge on the structure about sequential constraints. It ultimately allows longitudinal trajectories to be built from cross-sectional datasets.

To sum up our contributions, we introduce Finsler geometry and Randers metrics as both a principled model and a minimal extension of current frameworks to model asymmetric processes. We then

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.



(a) (b)

Figure 1: **a.** Example of trajectory γ between z^0 and z^1. At $\gamma_t = z_t$ we consider the forward and backward trajectory. Because $\|\dot{\gamma}_t\|_{G(z)} = \|-\dot{\gamma}_t\|_{G(z)}$, the only difference in energy is induced by $\langle \omega, . \rangle$. The angle between $\omega(z_t)$ and $-\dot{\gamma}_t$ is small, so the dot product is positive, hence the associated energy is high. On the contrary, because $\omega(z_t)$ and $\dot{\gamma}_t$ are almost of opposite direction, the dot product is negative, hence the associated energy is lower : this direction is preferred. **b.** Interpolation then extrapolations. *1* : Sequential factors along the trajectory. In *blue* the interpolated trajectory, in *orange* the mean extrapolated sequential factors with $\pm 2\sigma$ on 50 trajectories. *2* : Start sample, *3* : End of interpolation, *4* : One extrapolated sample.

propose a lightweight algorithm to learn latent Randers metrics a posteriori from local scalar ordering. Finally, we evaluate our approach on controlled interpolation and extrapolation settings. We show that if Riemannian geometry can interpolate adequately when endpoints are observed, symmetric geometry cannot reliably extrapolate under directional dynamics. Randers geometry supplies the missing directional inductive bias.

2 A PRIMER ON GEOMETRY

2.1 RIEMANN MANIFOLDS

We consider $\mathcal{Z}$ a smooth complete sub-manifold of $\mathbb{R}^d$. Hence $\mathcal{Z}$ is endowed with a Riemannian metric G that defines a smooth inner product on the tangent space $T_z\mathcal{Z}$, $\langle \cdot, \cdot \rangle_z : (u, v) \mapsto u^T G(z) v$, where $G(z)$ is a symmetric positive definite matrix for all $z \in \mathcal{Z}$. We note $\|v\|_{G(z)} = \sqrt{\langle v, v \rangle_z}$. We can consider Riemannian manifolds as curved vector spaces where the energy to cross a certain region is given locally by the metric. We can then define the length L and energy E of a curve $\gamma \in \mathcal{C}^1([0, 1], \mathcal{Z})$ on the manifold:

$$L(\gamma) = \int_0^1 \|\dot{\gamma}_t\|_{G(\gamma_t)} dt \quad \text{and} \quad E(\gamma) = \int_0^1 \|\dot{\gamma}_t\|^2_{G(\gamma_t)} dt \ , \tag{1}$$

which in turn allows us to define the geodesic distance between points on the manifold as path of least distance. These minimizers coincide with the critical points of the energy functional for unit speed curves. For a more in depth introduction, one may refer to Guigui et al. (2023).

2.2 FINSLER MANIFOLDS

Finsler manifolds are a natural extension of Riemannian ones. They provide a way to smoothly encode asymmetric local distances whilst broadly keeping the same formulation as in Riemannian geometry. One might refer to Ohta (2021) for a more complete presentation.

Definition 1 (Finsler metric). A function $\mathcal{F}_z : T_z\mathcal{Z} \to \mathbb{R}_+$ at point z is a Finsler metric if: it is $\mathcal{C}^\infty$ on $T\mathcal{Z} \setminus 0$; positive homogeneity : $\mathcal{F}_z(\lambda v) = \lambda \mathcal{F}_z(v)$ for $\lambda > 0$; and the fundamental tensor $g_{ij}(z, v) = \frac{\partial^2 \mathcal{F}_z}{\partial v_i \partial v_j}$ is positive definite.

These metrics induce potentially asymmetric distances. In particular, we do not always have $\mathcal{F}_z(v) = \mathcal{F}_z(-v)$ since the homogeneity only applies to positive scalars. Using Euler's theorem on homogeneous function, we have that $\mathcal{F}_z^2(v) = v^T g(z, v) v = \|v\|^2_{g(z,v)}$.

Similarly to Riemannian geometry, shortest paths under the Finsler metric are also minimizers of the energy of the curve under unit speed assumption. Consequently, we can define the length and energy of curves and thus geodesic distances on Finsler manifolds by replacing $G(\gamma_t)$ by $g(\gamma_t, \dot{\gamma}_t)$ in Equations 1. This enables to use the same class of algorithm as in the Riemannian case to solve the shortest path problem for marginal additional cost (Rygaard & Hauberg, 2025; Arvanitidis et al., 2019).

Definition 2 (Dual metric) The dual metric $\mathcal{F}_z^* : T_z^*\mathcal{Z} \to \mathbb{R}_+$ of a Finsler metric is given by

$$\forall p \in T_z^*\mathcal{Z},\ \mathcal{F}_z^*(p) = \sup_{\|v\| \neq 0} \frac{\langle v, p \rangle}{\mathcal{F}_z(v)}. \tag{2}$$

Definition 3 (Legendre transform) The *Legendre transform* $\mathcal{L} : T\mathcal{M} \to T^*\mathcal{M}$ is uniquely defined by the conditions

$$\mathcal{F}_z^*(\mathcal{L}_z(v)) = \mathcal{F}_z(v), \qquad \langle \mathcal{L}_z(v), v \rangle = \mathcal{F}_z^2(v). \tag{3}$$

The Legendre transform is a diffeomorphism between the tangent bundle $T\mathcal{M}$ and its dual $T^*\mathcal{M}$. We can compute it using the fundamental tensor as $\mathcal{L}_z(v) = g(z, v)v$. In the case where the Finsler metric is Riemannian, the dual metric reduces to the usual co-metric G^{-1} used *e.g.* in Riemannian geodesic shooting (Girolami & Calderhead, 2011) and $\|p\|_z^* = \sqrt{p^T G(z)^{-1} p}$.

2.3 Randers-Finsler manifolds

Definition 4 (Randers-Finsler metric). Randers-Finsler metrics are a particular case of Finsler metrics (Randers, 1941). They are simple extension of Riemannian norm such that, for a Riemannian manifold $(\mathcal{Z}, G)$, we have:

$$\mathcal{F}_z(v) = \|v\|_{G(z)} + \langle \omega(z), v \rangle\,, \tag{4}$$

where $\omega : \mathcal{Z} \to T_\cdot^*\mathcal{Z}$ is such that $\|\omega(z)\|_{G^{-1}(z)} < 1$ to ensure positivity. These metrics are Finsler metrics. Note that for such metric we have an explicit form of g and g^{-1}, enabling efficient implementation. Also the dual metric is still a Randers metric, with explicit and easy to compute expression (Appendix A).

With these metrics, the direction only influences the dot product part of the metric. Thus, they can be seen as regular Riemannian norms regularized by a measure of agreement with a low amplitude vector field ω (Cheng & Shen, 2012). In particular, we can isolate *sequential hyperplanes* $\mathcal{P}_z = \{v \in T_z\mathcal{Z} \mid \langle \omega(z), v \rangle = 0\}$. Along these hyperplanes, the *1-form* has no influence and the local displacement's energy is only influenced by the Riemannian norm, see Figure 1a. Therefore, in order to minimize the energy along a trajectory, the tangent vector needs to be on the correct side of $\mathcal{P}_z$ so that $\langle \omega(z), v \rangle \leq 0$, *i.e.* the trajectory needs to be antipodal to ω.

3 Related work

3.1 Metric Learning

Metric learning is a long-standing field in machine learning (Friedman, 1994; Lebanon, 2002; Yang & Jin, 2006). It involves finding new data representations that better reflect the task at hand. Although initially focused on linear or at least analytical projections (Kulis et al., 2013), it has evolved to accommodate the rise of deep neural networks for data projection (Kaya & Bilge, 2019). However, these methods tend to fail when considering long-range interactions. Their lack of granularity drove the introduction of mixes of different spatially varying metrics to model relevant distances (Hauberg et al., 2012).

Hence, equipped with a Riemannian framework, it becomes possible to describe *local to global* interactions. The difficulty lies in designing a Riemannian metric G that is both computable and expressive enough to characterize the data (Gruffaz & Sassen, 2025). Building on the data manifold hypothesis (Fefferman et al., 2016), a body of work aims to pull-back the geometry from the image manifold to the latent space (Arvanitidis et al., 2018; Sun et al., 2025b). Yet these approaches suffer two drawbacks. First, albeit these metrics are defined on any Riemannian metrics to be pulled back, they often work on the underlying hypothesis that the local Euclidean distance is a correct specification of the image space, while a practical assumption, it is known to be false in general (Wang et al., 2005). Then, from a computational standpoint, they amount to computing the Jacobians of neural networks which scales horrendously poorly with the network size. Consequently other authors have aimed to learn more efficient representation, either by directly working in the latent space (Arvanitidis et al., 2016; Kapusniak et al., 2024; Chadebec & Allassonnière, 2022), learning the metric end-to-end in the latent space (Chadebec et al., 2022), or taking an information geometry point of view (González-Duque et al., 2022).

3.2 Finsler Geometry

This work aims to further the use of Finsler geometry in machine learning. While these tools have received extensive attention from their inception in mathematics (Finsler, 1918; Ohta, 2021; Yeon, 2015), they were quickly adopted in general relativity to model spacetime asymmetries (Randers, 1941; Pfeifer, 2019). This framework lends itself to model other potentially asymmetric physical phenomenons such as computing the shortest navigation path under low amplitude wind (Zermelo, 1931; Markvorsen et al., 2025), fire-front propagation (Markvorsen, 2016) or distances on sloped surfaces (Matsumoto, 1989). Such metrics have also been used to model anisotropy in shape analysis and image processing (Yang et al., 2018; Weber et al., 2024; Chen et al., 2024). Recently the links between stochastic Riemannian metric and associated Finslerian geometry have been investigated by Pouplin et al. (2022), along with the associated algorithm to compute geodesics (Rygaard & Hauberg, 2025). However, learning a Finsler metric from the data has only been scarcely explored, notable examples include: Dagès et al. (2026) uses a constant Randers metrics to inject a prior in dimension reduction algorithms ;Zweig et al. (2026) learns transport maps using a lineage guided tree Finsler metric; and Gahtan et al. (2026) learns the Randers metric associated to fire-front propagation based on wind and topological data.

4 Methods

From now on, we will consider a dataset $\mathcal{D} = \{(x_i, z_i, s_i)_i\}$ where $x_i \in \mathcal{X} \subset \mathbb{R}^D$ are the images, $z_i \in \mathcal{Z} \subset \mathbb{R}^d$ their latent features, and $s_i \in \mathbb{R}$ monotonous scalar values. We also require the s_i to be totally ordered. These values are scores reflecting the sequential dynamic we want to follow in the latent space: time in longitudinal studies, cognitive score for neurological diseases etc. We limit our investigation to phenomena modellable by a monotonous score, that is, for a series of T measurements from a single individual, we require $s_i^1 \leq \cdots \leq s_i^T$. Being totally ordered is a strong hypothesis. For example, in a clinical setting, it means that we can with certainty compare disease progression between two *different* patients given their scores. We emphasize here that our framework only requires a *cross-sectional dataset* fulfilling the above hypothesis.

In this section, we first present the underlying Riemannian metrics that will be used. The presented metrics are either extracted or adapted from the Riemannian metric learning literature. We then introduce our core contribution: a learning algorithm for the *1-form* ω of the Randers metric following the insights from Figure 1a.

4.1 The Riemannian metrics

We now assume that the latent space $(\mathcal{Z}, G)$ is a Riemannian manifold with metric G. Usually, the implicit Euclidean assumptions considers that we have $G = I_d$ and thus that geodesics are straight lines. Riemannian metric learning aims to learn appropriate G.

We consider a VAE (Kingma & Welling, 2014) $\mathcal{X} \xrightarrow{g} \mathcal{Z} \xrightarrow{f} \mathcal{X}$, where g is the encoder such that $g(x_k) = (z_k, \Sigma_k)$ with Σ_k the latent covariance matrix associated to the features z_k. It allows to

define various metric on the latent space. We first use the diagonal metric from Arvanitidis et al. (2016)(**LAND**):

$$G_{ii}^{-1}(z) = \sum_k \exp\left(-\frac{\|z - z_k\|_2^2}{2\tau_k^2}\right)(z_k^{(i)} - z^{(i)})^2 + \lambda\,. \tag{5}$$

It enforces the data manifold support using Gaussian-weighted centroids. We also consider the variants from Kapusniak et al. (2024)(**LAND RBF**), where the interpolation weights are learned, and from Chadebec & Allassonnière (2022)(**CovCentroids**) that uses the learned covariance matrix Σ_k instead of the local covariance of the latent samples. The parameters τ_k control each centroids' importance while λ is a regularization parameter[1]. Also, given the Jacobian of the decoder $J_f(z) = \frac{\partial f}{\partial z}$, we have that $G_{\text{pb}}(z) = J_f(z)J_f(z)^T$ defines a Riemannian metric. To be used in practice, these pullback-based metric need specific calibration Arvanitidis et al. (2018); González-Duque et al. (2022); Sun et al. (2025a). Still, they require to differentiate through potentially large networks. In our setup, these metric don't fit on our available compute memory to compute geodesics.

To alleviate the significant cost of recomputing the Jacobian matrices at each point we consider a computationally tractable variant that we subsequently call the **PBCentroids** metric:

$$G^{-1}(z) = \sum_k \exp\left(-\frac{\|z - z_k\|^2}{2\tau_k^2}\right)G_{\text{pb}}(z_k)^{-1} + \lambda I_d\,. \tag{6}$$

We provide a more detailed overview of these methods and their parameters in Appendix C.

4.2 The Randers metric

For ease of training, we parametrize ω by decomposing the direction and amplitude such that

$$\omega_\theta(z) = \rho_\theta(z)\phi_\theta(z)\,. \tag{7}$$

This factorization decorrelates the sequential orientation from its strength. While $\phi_\theta : \mathcal{Z} \to T_z\mathcal{Z}$ is meant to showcase the sequential dynamic. $\rho_\theta : \mathcal{Z} \to [0, 1]$ aims to reflect how locally orientable the space is, effectively weighting the influence of the *1-form* in the metric $\mathcal{F}$. To enforce the positivity constraint 2.2, we learn ϕ_θ as:

$$\phi_\theta(z) = \frac{1}{\|\tilde{\phi}_\theta(z)\|_{G^{-1}(z)} + \varepsilon}\tilde{\phi}_\theta(z)\ , \tag{8}$$

where $\varepsilon << 1$ and $\tilde{\phi}_\theta$ is a neural network such that we enforce $\|\omega_\theta(z)\|_{G^{-1}(z)} < 1$.

Following the discussion of 2.3, we want to learn locally separating hyperplanes $\mathcal{P}_z$ for $z \in \mathcal{Z}$. Inspired by the SVM literature (Hearst et al., 1998; Xu et al., 2012), we thus learn $\tilde{\phi}_\theta$ through a local hinge loss such that :

$$l_\phi(z_i) = \frac{1}{K_\phi}\sum_{z_j \in \text{KNN}(z_i)} \max(0, 1 - y_i^j \times \tilde{\phi}_\theta^\top(z_i)G(z_i)(z_j - z_i))\ , \tag{9}$$

where $y_i^j = 2 \times \mathbf{1}_{(s_j < s_i)} - 1 \in \{\pm 1\}$. The loss is only computed on the K_ϕ closest neighbors of each point to enforce locality and use the Riemannian dot product.

This results in a vector field ϕ_θ on the manifold that is able to locally discriminate the "before" from the "after" effectively orienting the latent space. Yet usually not all the latent space can be correctly orientated. To account for this we regularize the strength of the asymmetry of the metric through ρ_θ by regressing the local accuracy of the separating hyperplane. That is, at point z_i we compute the local accuracy of the hyperplane and we consider the region orientable if it is above 0.5. We thus learn to regress its significant portion mapped to $[0, 1]$:

$$l_\rho(z_i) = \left\|\rho_\theta(z_i) - 2 \times \max\left(0, \text{ACC}_\phi(z_i) - \frac{1}{2}\right)\right\|^2 \tag{10}$$

[1]For larger dataset, the sum can be computed on a subset of points using *K-Medoids*

$$\text{where} \quad \text{ACC}_\phi(z_i) = \frac{1}{K_\phi} \# \left\{ y_i^j \times \phi_\theta(z_i)^\top G(z_i)(z_j - z_i) > 0 \mid z_j \in \text{KNN}(z_i) \right\}. \tag{11}$$

Consequently, when ϕ_θ produces a good separating hyperplane, that means that the latent space is locally orientable and we will have $\rho_\theta \approx 1$. On the contrary, when the sequential dynamic is not structuring enough, we will have $\rho_\theta \approx 0$, effectively falling back to the regular Riemannian geometry.

We first learn ϕ_θ by optimizing l_ϕ then we learn ρ_θ by optimizing l_ρ. This yields in practice a lightweight algorithm as it essentially amounts to learning two small networks ϕ and ρ in small dimensions with possibly pre-extracted latent features. We investigate the influence of the polar decomposition in Appendix F.

4.3 Interpolation

To assess the validity of our framework, we quantify the regression of the sequential factors along interpolated trajectories. So we sample z_0 and z_1 such that $s_0 < s_1$. We then compute the latent geodesic γ_t from z_0 to z_1 using $T = 25$ time points. The geodesic are initialized by the shortest path of the data's KNN-graph (Kim et al., 2024). They are then refined using the GEORCE solver from Rygaard & Hauberg (2025) for diagonal metrics, or by minimizing energy of the curve under BSpline parametrization for dense metrics (Shtoff, 2025). Computations details are available in Appendix E.

Proposition 1 (Path invariance) Let U be a simply connected domain of $\mathcal{Z}$. If there exists a smooth potential function ψ such that $\omega_\theta = \nabla\psi$, then the Randers and Riemannian geodesics between fixed endpoints coincide up to reparametrization. In particular $dst_{\mathcal{F}}(z_0, z_1) = dst_G(z_0, z_1) + \psi(z_1) - \psi(z_0)$.

This results, proved in Appendix B, demonstrates that if the learned vector field ω_θ approximate the gradient of the sequential factors s, ie $\omega \approx \nabla s$, then the interpolated Randers trajectory will be identical to the Riemannian ones.

4.4 Extrapolation

The learned vector field enables the computation of dynamic aware extrapolation. Here we use geodesic guided shooting techniques. Indeed, given the dual metric $\mathcal{F}_*$, we can show that, for the Hamiltonian $H(p, q) = \frac{1}{2}\mathcal{F}_*(q, p)^2$ with $(q, p) \in T^*\mathcal{M}$, the flow

$$\dot{p}_t = -\frac{\partial H}{\partial q}(p_t, q_t) \quad \dot{q}_t = \frac{\partial H}{\partial t}(p_t, q_t) \tag{12}$$

yields geodesic trajectories (Hryniewicz & Salomão, 2013, section 2.3). As the functional is not separable, we integrate the system using the *generalized leapfrog algorithm* with implicit steps (Girolami & Calderhead, 2011; Chadebec et al., 2022; Cobb et al., 2019).

We can initialize this flow using the learned dynamic encoded by ω. So if we note $\tilde{\omega}_0 = \frac{\omega(q_0)}{\|\omega(q_0)\|_2}$, the initial speed $\widetilde{v}_0 \in T_{q_0}\mathcal{M}$ is sampled such that $\widetilde{v}_0 \mid q_0 \sim \text{vMF}(-\tilde{\omega}_0, \kappa)$ follows a von Mises Fisher distribution (Pinzón & Jung, 2023) centered around the direction of evolution given by $-\tilde{\omega}_0$ and of concentration κ. We essentially sample $\tilde{v}_0$ in the cone of direction $-\tilde{\omega}_0$ and aperture κ. We then rescale the initial speed to have a norm α, $v_0 = \alpha\tilde{v}_0$. This initial speed $v_0 \in T_{q_0}\mathcal{M}$ is then pushed to the dual tangent space using the Legendre transform $p_0 = g(q_0, v_0)v_0 \in T^*_{q_0}\mathcal{M}$.

We also choose κ to match a target concentration η that verifies $\eta = \mathbb{E}_{v\sim\text{vMF}(-\tilde{\omega}_0,\kappa)}[\text{cosim}(-\tilde{\omega}_0, v)]$. The parameter η allows to tune the angle between the sampled direction of evolution $-\tilde{\omega}_0$ and the initial speed v_0. A value of $\eta = 1$ means that the sampled direction is always aligned with $-\tilde{\omega}_0$, while $\eta = 0$ means that the sampled direction is uniformly distributed on the sphere. The following proposition proved in in Appendix G allows to find the expression of κ given η.

Proposition 2 For a given concentration parameters η, we have that

$$\eta = I_{\frac{d}{2}}(\kappa)/I_{\frac{d}{2}-1}(\kappa), \quad \text{such that} \quad \kappa \approx \frac{\eta d - \eta^3}{1 - \eta^2}, \tag{13}$$

where I_d is the modified Bessel function of the first kind of order d.

To ensure fair comparison with Riemannian settings, we follow the usual setup and sample $\widetilde{v_0} \mid q_0 \sim \mathcal{N}(0, G^{-1}(q_0))$. We also rescale the initial speed so that $v_0 = \alpha \frac{\widetilde{v}_0}{\|\widetilde{v}_0\|_2}$ which yields $p_0 = G(q_0)v_0$.

In order to mitigate the effects of numerical approximation, we also resample the speed after each integration step with momentum ζ, so that at time step t, we have :

$$v_{t+1} = \zeta v_t + (1 - \zeta)v_{\text{sampled}}. \tag{14}$$

4.5 Evaluation setup

Given a trajectory in the latent space, either from interpolation or extrapolation, it is first decoded back to the image space. Then we can recover the unknown sequential factors s_t^γ. We evaluate the discrete trajectories of sequential factors $s^\gamma = (s_k^\gamma)_{k\in[T]}$ using the monotonicity score $\tau(\gamma)$ defined as the Kendall rank correlation coefficient between s^γ and $\{0, \cdots, T\}$. In the extrapolation setup, we also report the relative variation of the sequential factors as $100 \times \frac{s_T^\gamma - s_0^\gamma}{s_0^\gamma}$. This allows to monitor the ability of the extrapolated trajectory to actually produce meaningful diverse trajectories. We also report the Fréchet Wavelet Distance (FWD) (Veeramacheneni et al., 2023) to ensure that the generated images are within the data distribution. A summary of the learning algorithms and evaluation procedure is available in Appendix I.

5 Experiments

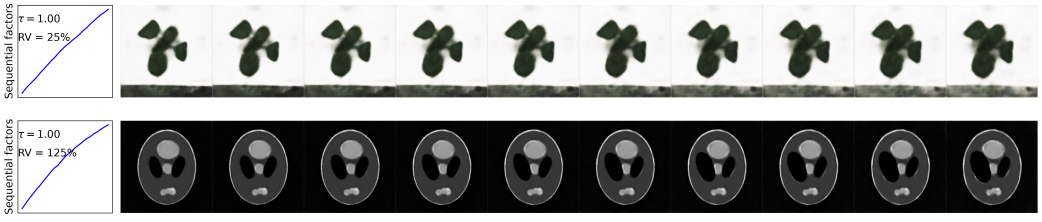


Figure 2: Example of extrapolated trajectory on the KOMATSUNA dataset (*top*) and Shepp-Logan (*bottom*) using the asymmetric framework with *LAND* base metric. Left columns displays the sequential factors along the trajectory where τ is the Kendall Tau correlation and *RV* is the relative variation.

As the latent space is structured by the metric G and the sequential dynamics are described by ω, we can provide dynamic aware possible evolutions given a single acquisition point (Figure 1b), or even create synthetic longitudinal cohorts given a cross-sectional cohort (Figure 2).

Following other work in the metric learning literature Diepeveen et al. (2025); Arvanitidis et al. (2018); Sun et al. (2025a), we first focus on a controlled synthetic setting, here fulfilling the framework's hypothesis : monotonous dynamics, totally ordered and continuous sequential factor easily retrievable from the image and no label noise. This setting and the provided experiences permit to isolate the causes of failure of usual approaches whilst showcasing the desired behavior achieved by the introduced Randers metric. The framework is motivated by clinical progression problems; our experiments establish feasibility in controlled medical-imaging-like and biological-growth settings.

We experiment using the 5 metrics described in Section 4.1 using a VAE trained from scratch. Architecture and training details are available in Appendix 7. For each metric we report the results using the base metric and its Finsler variant with the learned vector field ω. We also compare our approach to other baselines : *Wasserstein gradient flow learning* (Huizing et al., 2026) and *HyperSphericalVAE* (Sablica & Hornik, 2025), see details at Appendix D. We ablate the hyper-parameters influence and noise robustness in Appendix F.

5.1 SHEPP-LOGAN DATASET

To simulate a medical imaging dataset fitting our criterion we use variations of the Shepp-Logan phantom (Shepp & Logan, 1974). It consists of a model of a human head with the skull, the brain, and several ellipsis representing different brain sections. We generate images ($n_{\text{train}} = 1000$ and $n_{\text{test}} = 500$) by varying the colour and shape of the ellipsis and adding random gaussian noise. The sequential factors (s_i) are here the area of the top left black ellipsis. They aim to represent tumour size.

5.2 KOMATSUNA DATASET

The KOMATSUNA dataset (Uchiyama et al., 2017) consists of a series of images ($n_{\text{train}} = 720$ and $n_{\text{test}} = 180$) of the growth process of the komatsuna plant. We use as sequential factors s_i the leaf area of the plant as it fulfills the framework's hypothesis.

5.3 RESULTS

Table 1: Combined extrapolation and interpolation results on the Shepp-Logan dataset. Here, $d = 64, \alpha = 5.0, \eta = 0.999$, and $\zeta = 0.9$.

		Interpolation		Extrapolation	
Base metric	Finsler	Kendall Tau (↑)	FWD (↓)	Kendall Tau (↑)	Relative Variation (%)
Euclidean	✗	0.94 ± 0.19	0.35	0.02 ± 0.81	0.09 ± 5.72
	✓	1.0 ± 0.02	0.41	1.0 ± 0.02	85.59 ± 33.52
LAND [1]	✗	0.46 ± 0.29	0.23	-0.04 ± 0.8	-0.14 ± 6.04
	✓	0.46 ± 0.29	0.19	1.0 ± 0.0	90.00 ± 44.26
LAND RBF [29]	✗	0.46 ± 0.29	0.21	0.0 ± 0.8	-0.11 ± 5.59
	✓	0.46 ± 0.3	0.22	1.0 ± 0.02	83.47 ± 30.76
CovCentroids [9]	✗	0.46 ± 0.29	0.17	0.03 ± 0.79	0.15 ± 6.68
	✓	0.45 ± 0.29	0.2	1.0 ± 0.03	74.53 ± 74.71
PBCentroids	✗	0.96 ± 0.09	0.21	0.01 ± 0.79	0.32 ± 4.55
	✓	0.77 ± 0.31	0.23	0.59 ± 0.55	28.75 ± 70.48
Wasserstein [28]	✗	$\cdots$	$\cdots$	0.97 ± 0.15	25.54 ± 24.04
HyperSpherical [47]	✗	1.0 ± 0.04	0.35	-0.03 ± 0.75	-0.48 ± 53.77

The results are displayed in Table 1 for SheppLogan dataset and in Table 2 for the KOMATSUNA dataset. We can draw similar conclusions for both datasets. The naive symmetric *Euclidean* framework reaches top performances in interpolation. This is expected as the Gaussian prior on the latent space tends to convexify the latent space, which mitigates data manifold support issues. Furthermore, VAE are known to have relatively disentangled representations (Burgess et al., 2018), hence the good interpolation performances. In most settings the addition of the asymmetry doesn't change the interpolation results. This is expected per *Proposition 1*. Indeed, if we can assume $\omega \approx \nabla s$, then the Finslerian and Riemannian interpolated trajectories are the same, which yields the observed identical scores. The discrepancies observed on *PBCentroids* in Table 1 and *LAND* in Table 2 are to attributed to the base metric. Indeed the learning of ω_θ is induced by the angles defined by $\langle \cdot, \cdot \rangle_G$. A miss specification of the metric for a given dataset will yield a spurious learning of the vector field.

The proposed method proves to be more useful in the extrapolation regime. The results demonstrate that the asymmetric frameworks yield the correct expected behavior. Indeed, the symmetric shooting methods actually amount to a non homogeneous random walk on the manifold, thus on average describing a diffusion process. Incorporating prior information in the geometry allows to provide sequentially consistent trajectories. The results hold even in the presence of moderate noise on the labels, see details in Appendix F.

Table 2: Combined extrapolation and interpolation results on the KOMATSUNA dataset. Here, $d = 64, \alpha = 5.0, \eta = 0.999$, and $\zeta = 0.9$.

		Interpolation		Extrapolation	
Base metric	Finsler	Kendall Tau (↑)	FWD (↓)	Kendall Tau (↑)	Relative Variation (%)
Euclidean	✗	0.98 ± 0.06	4.67	-0.05 ± 0.78	0.16 ± 8.63
	✓	0.98 ± 0.06	4.71	1.0 ± 0.05	138.56 ± 130.66
LAND [1]	✗	0.76 ± 0.26	4.02	-0.11 ± 0.78	-1.01 ± 8.01
	✓	0.57 ± 0.29	4.29	1.0 ± 0.01	112.84 ± 97.52
LAND RBF [29]	✗	0.98 ± 0.04	4.28	-0.12 ± 0.76	-0.36 ± 7.52
	✓	0.99 ± 0.03	4.53	1.0 ± 0.01	141.37 ± 140.31
CovCentroids [9]	✗	0.42 ± 0.33	3.76	-0.05 ± 0.78	-0.02 ± 10.08
	✓	0.43 ± 0.41	3.47	0.99 ± 0.13	332.15 ± 584.35
PBCentroids	✗	0.44 ± 0.38	6.07	-0.06 ± 0.76	1.53 ± 18.50
	✓	0.48 ± 0.38	5.65	0.36 ± 0.58	63.83 ± 192.03
Wasserstein [28]	✗	$\cdots$	$\cdots$	0.87 ± 0.37	351.62 ± 489.63
HyperSpherical [47]	✗	0.99 ± 0.03	7.63	0.18 ± 0.72	98.25 ± 170.43

The directional component dominates the choice of base metric in the tested regimes, whereas interpolation remains sensitive to the base geometry. Apart for the *PBCentroids* one, they all achieve high monotonicity score whilst exploring a wide range of extrapolated data as shown by the high and diverse relative variation score. The hyper-parameter search in Appendix F reveals that the tuning of the concentration parameter κ is crucial to get good performances. For example, on the SheppLogan dataset the relative variation drop from $74.53 \pm 74.71\%$ to $37.48 \pm 12.45\%$ on the *CovCentroids* geometry when η changes from 0.999 to 0.5 while still maintaining high Kendall Tau. It also demonstrates that, unsurprisingly, trajectories reaching further, or equivalently with higher speed magnitude, achieve broader exploration of the space whilst respecting the learned dynamic : the relative variation drops from $74.53 \pm 74.71\%$ to $18.88 \pm 5.63\%$ on the *CovCentroids* geometry when α changes from 5.0 to 1.0, while still maintaining high Kendall Tau again.

6 Limitations and Future work

We see two important lines of future work. First we need to assess to scalability of the presented method to bigger datasets. To open up more potential datasets, we need to investigate how to relax the total ordering hypothesis on the scalar values s_i. So far it is only used to compute local pseudo-labels in the learning of ω. One could imagine computing the loss only on a single patient samples. But this would require very important sampling frequency so as to maintain the desired locality. We could extend this framework to work with binned discrete clinical score, such as LI-RADS or MMSE, as they enable inter-patient comparison. The problem lies in having adequate evaluation since these quantities are hard to retrieve from images (Wang et al., 2023) and are subject to noisy labels (Low et al., 2025). Although we hypothesize that the demonstrated robustness to noise could alleviate noisy labels issues.

7 Conclusion and discussion

Faithful latent extrapolation under sequential dynamics constraints requires modelling directionality beyond curvature. We embed such dynamics as a static learned asymmetric geometry through Finsler-Randers metrics, allowing us to model asymmetric transition cost whilst avoiding the need for longitudinal datasets. This lightweight directional regularization outperforms symmetric frameworks in our experiments. It suggests that asymmetry is an important ingredient in modelling progression. This opens the door to richer geometric priors for trajectory extrapolation in clinical settings.

AI USE STATEMENT

In this work, we used generative AI tools for to provide feedback on research. We have not used generative AI tools to develop the conceptual framework developed in the paper, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit software code, summarize existing literature, edit the research paper to improve readability, identify relevant literature, format references. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code will be available on GitHub and is provided in the supplementary material. It includes a toolkit to work with learned Riemannian/Finslerian manifold (metrics definition, geodesic solvers and samplers) as well as the experiments detailed in the paper. The main algorithm are included as pseudo-code in Appendix I.

REFERENCES

Georgios Arvanitidis, Lars K Hansen, and Søren Hauberg. A locally adaptive normal distribution. *Advances in Neural Information Processing Systems*, 29, 2016.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: On the curvature of deep generative models. In *6th International Conference on Learning Representations, ICLR 2018*, 2018.

Georgios Arvanitidis, Soren Hauberg, Philipp Hennig, and Michael Schober. Fast and robust shortest paths on manifolds learned from data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1506–1515. PMLR, 2019.

Mehdi Astaraki, Guang Yang, Yousuf Zakko, Iuliana Toma-Dasu, Oerjan Smedby, and Chunliang Wang. A comparative study of radiomics and deep-learning based methods for pulmonary nodule malignancy prediction in low dose ct images. *Frontiers in oncology*, 11:737368, 2021.

Arindam Banerjee, Inderjit S Dhillon, Joydeep Ghosh, Suvrit Sra, and Greg Ridgeway. Clustering on the unit hypersphere using von mises-fisher distributions. *Journal of Machine Learning Research*, 6(9), 2005.

D. Bao, S.-S. Chern, and Z. Shen. *An Introduction to Riemann-Finsler Geometry*, volume 200 of *Graduate Texts in Mathematics*. Springer, 2000. ISBN 978-1-4612-7070-6 978-1-4612-1268-3. doi: 10.1007/978-1-4612-1268-3.

J Frederic Bonnans, Guillaume Bonnet, and Jean-Marie Mirebeau. A linear finite-difference scheme for approximating randers distances on cartesian grids. *ESAIM: Control, Optimisation and Calculus of Variations*, 28:45, 2022.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Clément Chadebec and Stéphanie Allassonnière. A geometric perspective on variational autoencoders. *Advances in neural information processing systems*, 35:19618–19630, 2022.

Clément Chadebec, Elina Thibeau-Sutre, Ninon Burgos, and Stéphanie Allassonnière. Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2879–2896, 2022.

Da Chen, Jean-Marie Mirebeau, Huazhong Shu, and Laurent D Cohen. A region-based randers geodesic approach for image segmentation. *International Journal of Computer Vision*, 132(2): 349–391, 2024.

Xinyue Cheng and Zhongmin Shen. *Finsler Geometry*. Springer, 2012. ISBN 978-3-642-24887-0 978-3-642-24888-7. doi: 10.1007/978-3-642-24888-7.

Adam D Cobb, Atılım Güneş Baydin, Andrew Markham, and Stephen J Roberts. Introducing an explicit symplectic integration scheme for riemannian manifold hamiltonian monte carlo. *arXiv preprint arXiv:1910.06243*, 2019.

Thomas Dagès, Simon Weber, Daniel Cremers, and Ron Kimmel. Harnessing data asymmetry: Manifold learning in the finsler world. *arXiv preprint arXiv:2603.11396*, 2026.

Willem Diepeveen, Georgios Batzolis, Zakhar Shumaylov, and Carola-Bibiane Schönlieb. Score-based pullback riemannian geometry: Extracting the data manifold geometry using anisotropic flows. In *International Conference on Machine Learning*, pp. 13746–13773. PMLR, 2025.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2681–2690. PMLR, 2019.

Paul Finsler. *Ueber kurven und Flächen in allgemeinen Räumen*. Philos. Fak., Georg-August-Univ., 1918.

Jerome H Friedman. Flexible metric nearest neighbor classiżcation. Technical report, Citeseer, 1994.

Barak Gahtan, Jacob Shpund, and Alex M Bronstein. Wildfire simulation with differentiable randers-finsler eikonal solvers. *arXiv preprint arXiv:2603.00035*, 2026.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, 2011. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2010.00765.x.

Miguel González-Duque, Georgios Arvanitidis, Alison Pouplin, Dimitris Kalatzis, and Søren Hauberg. Pulling back information geometry. In *25th International Conference on Artificial Intelligence and Statistics*, 2022.

Samuel Gruffaz and Josua Sassen. Riemannian Metric Learning: Closer to You than You Imagine. *arXiv preprint arXiv:2503.05321*, 2025. doi: 10.48550/arXiv.2503.05321.

Nicolas Guigui, Nina Miolane, and Xavier Pennec. Introduction to Riemannian Geometry and Geometric Statistics: From Basic Theory to Implementation with Geomstats. *Foundations and Trends® in Machine Learning*, 16(3):329–493, 2023. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000098.

Søren Hauberg, Oren Freifeld, and Michael Black. A geometric take on metric learning. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Umberto L Hryniewicz and Pedro AS Salomão. Global properties of tight reeb flows with applications to finsler geodesic flows on s2. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 154, pp. 1–27. Cambridge University Press, 2013.

Geert-Jan Huizing, Jules Samaran, Daniele Capocefalo, Anna Audit, Gabriel Peyré, and Laura Cantini. Stories: learning cell fate landscapes from spatial transcriptomics using optimal transport. *Nature Methods*, 23(3):522–531, 2026.

Kacper Kapusniak, Peter Potaptchik, Teodora Reu, Leo Zhang, Alexander Tong, Michael Bronstein, Joey Bose, and Francesco Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *Advances in Neural Information Processing Systems*, 37:135011–135042, 2024.

Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

Beomsu Kim, Michael Puthawala, Jong Chul Ye, and Emanuele Sansone. (Deep) Generative Geodesics. *arXiv preprint arXiv:2407.11244*, 2024. doi: 10.48550/arXiv.2407.11244.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *Internation Conference on Learning Representations*, 2014.

Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4): 287–364, 2013.

Guy Lebanon. Learning riemannian metrics. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pp. 362–369, 2002.

Gavin Low, Tyler Pfanner, Xu Jing Qian, Ali Ramji, Karim Samji, and Mitchell P Wilson. Lr-m for ct/mri on li-rads v2018: a review of imaging criteria, performance, challenges and future directions from an end-user perspective. *Abdominal Radiology*, pp. 1–15, 2025.

Steen Markvorsen. A finsler geodesic spray paradigm for wildfire spread modelling. *Nonlinear Analysis: Real World Applications*, 28:208–228, 2016.

Steen Markvorsen, Enrique Pendás-Recondo, and Frederik Möbius Rygaard. Time-dependent zermelo navigation with tacking. *arXiv preprint arXiv:2508.07274*, 2025.

Makoto Matsumoto. A slope of a mountain is a finsler surface with respect to a time measure. *Journal of Mathematics of Kyoto University*, 29(1):17–25, 1989.

Clare McGenity, Emily L Clarke, Charlotte Jennings, Gillian Matthews, Caroline Cartlidge, Henschel Freduah-Agyemang, Deborah D Stocken, and Darren Treanor. Artificial intelligence in digital pathology: a systematic review and meta-analysis of diagnostic test accuracy. *NPJ digital medicine*, 7(1):114, 2024.

Lu Mi, Tianxing He, Core Francisco Park, Hao Wang, Yue Wang, and Nir Shavit. Revisiting latent-space interpolation via a quantitative evaluation framework. *arXiv preprint arXiv:2110.06421*, 2021.

Shin-ichi Ohta. *Comparison Finsler Geometry*. Springer Monographs in Mathematics. Springer International Publishing, 2021. ISBN 978-3-030-80649-1 978-3-030-80650-7. doi: 10.1007/978-3-030-80650-7.

Christian Pfeifer. Finsler spacetime geometry in physics. *International Journal of Geometric Methods in Modern Physics*, 16(supp02):1941004, 2019.

Carlos Pinzón and Kangsoo Jung. Fast python sampler for the von mises fisher distribution. *preprint*, 2023.

Alison Pouplin, David Eklund, Carl Henrik Ek, and Søren Hauberg. Identifying latent distances with finslerian geometry. *arXiv preprint arXiv:2212.10010*, 2022.

Gunnar Randers. On an asymmetrical metric in the four-space of general relativity. *Physical Review*, 59(2):195, 1941.

Frederik Möbius Rygaard and Søren Hauberg. GEORCE: A Fast New Control Algorithm for Computing Geodesics. *arXiv preprint arXiv:2505.05961v1*, 2025.

Lukas Sablica and Kurt Hornik. Hyperspherical variational autoencoders using efficient spherical cauchy distribution. *arXiv preprint arXiv:2506.21278*, 2025.

Erich Schubert and Lars Lenssen. Fast k-medoids clustering in rust and python. *Journal of Open Source Software*, 7(75):4183, 2022.

Lawrence A Shepp and Benjamin F Logan. The fourier reconstruction of a head section. *IEEE Transactions on nuclear science*, 21(3):21–43, 1974.

Alex Shtoff. torchcurves: Differentiable parametric curves in pytorch. *GitHub*, 2025. URL https://github.com/alexshtf/torchcurves.

Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim GJ Rudner, and Smita Krishnaswamy. Geometry-aware generative autoencoders for warped riemannian metric learning and generative modeling on data manifolds. In *International Conference on Artificial Intelligence and Statistics*, pp. 1018–1026. PMLR, 2025a.

Xingzhi Sun, Danqi Liao, Kincaid MacDonald, Yanlei Zhang, Chen Liu, Guillaume Huguet, Guy Wolf, Ian Adelstein, Tim G. J. Rudner, and Smita Krishnaswamy. Geometry-Aware Generative Autoencoders for Warped Riemannian Metric Learning and Generative Modeling on Data Manifolds. *arXiv preprint arXiv:2410.12779*, 2025b. doi: 10.48550/arXiv.2410.12779.

Hideaki Uchiyama, Shunsuke Sakurai, Masashi Mishima, Daisaku Arita, Takashi Okayasu, Atsushi Shimada, and Rin-ichiro Taniguchi. An easy-to-setup 3d phenotyping platform for komatsuna dataset. In *2017 IEEE international conference on computer vision workshops (ICCVW)*, pp. 2038–2045. IEEE, 2017.

Lokesh Veeramacheneni, Moritz Wolter, Hildegard Kuehne, and Juergen Gall. Fr\'echet wavelet distance: A domain-agnostic metric for image generation. *arXiv preprint arXiv:2312.15289*, 2023.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in Neural Information Processing Systems*, 31, 2018.

Ke Wang, Yuehua Liu, Hongxin Chen, Wenjin Yu, Jiayin Zhou, and Xiaoying Wang. Fully automating li-rads on mri with deep learning-guided lesion segmentation, feature characterization, and score inference. *Frontiers in Oncology*, 13:1153241, 2023.

Liwei Wang, Yan Zhang, and Jufu Feng. On the euclidean distance of images. *IEEE transactions on pattern analysis and machine intelligence*, 27(8):1334–1339, 2005.

Simon Weber, Thomas Dages, Maolin Gao, and Daniel Cremers. Finsler-laplace-beltrami operators with application to shape analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3131–3140, 2024.

Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4582–4591, 2017.

Zhixiang Xu, Kilian Q Weinberger, and Olivier Chapelle. Distance metric learning for kernel machines. *arXiv preprint arXiv:1208.3422*, 2012.

Fang Yang, Li Chai, Da Chen, and Laurent Cohen. Geodesic via asymmetric heat diffusion based on finsler metric. In *Asian Conference on Computer Vision*, pp. 371–386. Springer, 2018.

Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. *Michigan State Universiy*, 2(2):4, 2006.

Won Dae Yeon. On the History of the Birth of Finsler Geometry at Göttingen. *Journal of the Korean Society for the History of Mathematics*, 2015.

Ernst Zermelo. Über das navigationsproblem bei ruhender oder veränderlicher windverteilung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 11(2):114–124, 1931.

Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.

Aaron Zweig, Mingxuan Zhang, David A Knowles, and Elham Azizi. Learning lineage-guided geodesics with finsler geometry. *arXiv preprint arXiv:2603.16708*, 2026.

TABLE OF CONTENTS (APPENDIX)

A Properties of Randers metric **14**

B Path invariance **15**

C Description of the metrics used **16**

D Wasserstein gradient flow learning **18**

E Geodesics computations **19**

F Ablations **19**

G Concentration parameter tuning **21**

H Experiments details **22**

I Pseudo code **23**

J Interpolation and Extrapolation example **25**

A PROPERTIES OF RANDERS METRIC

A.1 RANDERS-FINSLER METRICS ARE FINSLER METRICS

The proof is adapted from Weber et al. (2024). We aim to show that Randers metric are Finsler metric. Triangle inequality and positive homogeneity are trivial. Let's prove the positivity.

Let $(z, u) \in T\mathcal{Z}$. We note $v = G(z)u$. Then :

$$\begin{aligned}
\mathcal{F}(z, u) &= \mathcal{F}(z, G(z)^{-1}v) && (15)\\
&= \|G(z)^{-1}v\|_{G(z)} + \langle \omega(z), G(z)^{-1}v\rangle && (16)\\
&= \sqrt{v^T G(z)^{-1} v} + \omega(z)^T G(z)^{-1} v && (17)\\
&= \|G(z)^{-\frac{1}{2}}v\| + \langle G(z)^{-\frac{1}{2}}\omega(z), G(z)^{-\frac{1}{2}}v\rangle && (18)\\
& && (19)
\end{aligned}$$

Using the Cauchy-Schwartz inequality we get :

$$|\langle \omega(z), G(z)^{-1}v\rangle| \leq \|G(z)^{-\frac{1}{2}}\omega(z)\| \, \|G(z)^{-\frac{1}{2}}v\| \tag{20}$$

Because by hypothesis we have $\|\omega(z)\|_{G(z)^{-1}} < 1$, we have $\|G(z)^{-\frac{1}{2}}\omega(z)\| < 1$. Thus :

$$|\ \langle G(z)^{-\frac{1}{2}}\omega(z), G(z)^{-\frac{1}{2}}v\rangle\ | = |\ \langle \omega(z), G(z)^{-1}v\rangle\ | \leq \|G(z)^{-\frac{1}{2}}v\| \tag{21}$$

Thus, if $u \neq 0$, we get $\mathcal{F}(z, u) > 0$. So finally Randers metric are Finsler metric.

A.2 FUNDAMENTAL TENSOR OF A RANDERS METRIC

For clarity we use the same notation as in Ohta (2021). That is, we assume a Randers-Finsler metric in the class of (α, β)-metric where $\alpha_z(u, v) = a_{ij}(z)u^i v^j$ is a Riemannian metric and $\beta(z, v) = b_i(z)v^i$ is a *1-form* such that $\mathcal{F}(z, v) = \sqrt{\alpha_z(v, v)} + \beta(x, v)$. If we note $A = (a_{ij})$ and $b = (b_i)$, we have $\mathcal{F}(z, v) = \sqrt{v^T A(z) v} + b(z)^T v$.

Then, following the work of Bao et al. (2000), if we note :

$$\tilde{l}_i = \frac{a_{ij}v^j}{\|v\|_\alpha} \quad \text{and} \quad l_i = \tilde{l}_i + b_i \,. \tag{22}$$

Then we have :

$$g_{ij} = \frac{\mathcal{F}(z,v)}{\|v\|_\alpha}(a_{ij} - \tilde{l}_i\tilde{l}_j) + l_i l_j. \tag{23}$$

And if we note :

$$(a^{ij}) = (a_{ij})^{-1} \; ; \; b^i = \sum_{j=1}^{n} a^{ij}b_j \; ; \; \|\beta\|_\alpha = \sqrt{\sum_{i=1}^{n} b^i b_j} \tag{24}$$

We have:

$$g^{ij} = \frac{\|v\|_\alpha}{\mathcal{F}(z,v)}a^{ij}(x) + \frac{\beta(v) + \|\beta\|_\alpha^2(z)\|v\|_\alpha}{\mathcal{F}^3(z,v)}v^i v^j - \frac{\|v\|_\alpha}{\mathcal{F}^2(z,v)}(b^i(z)v^j + b^j(z)v^i) \tag{25}$$

A.3 Dual Randers Metric

The dual metric $\mathcal{F}_*$ of a Finsler metric is defined as :

$$\forall (q,p) \in T_*\mathcal{Z}, \quad \mathcal{F}_*(q,p) = \sup_{\|u\|\neq 0} \frac{\langle v, p\rangle}{\mathcal{F}(q,v)}. \tag{26}$$

Fortunately, the dual metric of a Randers metric is also a Randers metric (Yang et al., 2018). More precisely :

$$\forall (q,p) \in T_*\mathcal{Z}, \quad \mathcal{F}_*(q,p) = \|p\|_* + \langle \omega_*(q), p\rangle. \tag{27}$$

Where we have :

$$\delta = 1 - \langle \omega, G^{-1}\omega\rangle \; , G_* = \frac{(G^{-1}\omega)(G^{-1}\omega)^\top + \delta G^{-1}}{\delta^2} \; , \omega_* = -\frac{G_*^{-1}G^{-1}\omega}{\delta}. \tag{28}$$

The dual metric depend on the amplitude of the vector field $\|\omega(x)\|_{G^{-1}(x)}$. Consequently, to reduce numerical instabilities in the extrapolation settings where we use the dual vector field ω_*, we limit the vector field amplitude by capping the value of $\|\omega(x)\|_{G^{-1}(x)}$ to 0.9.

B Path invariance

Proposition 1 (Path invariance) Let U be a simply connected domain of $\mathcal{Z}$. If there exists a smooth potential function ψ such that $\omega = \nabla\psi$, then the Randers and Riemannian geodesics between fixed endpoints coincide of to reparametrization.

Proof Let $z_0, z_1 \in U \subset \mathcal{Z}$. Consider a curve $\gamma : [0,1] \to \mathcal{Z}$ such that $\gamma(0) = z_0$ and $\gamma(1) = z_1$. Then if we suppose that $\omega = d\psi$:

$$\begin{aligned}
L_{\mathcal{F}}(\gamma) &= \int_0^1 \mathcal{F}_{\gamma_t}(\dot{\gamma}_t)dt \\
&= \int_0^1 \sqrt{\dot{\gamma}_t^\top G(\gamma_t)\dot{\gamma}_t} + \langle \omega(\gamma_t), \dot{\gamma}_t\rangle dt \\
&= L_G(\gamma) + \int_0^1 \langle \nabla\psi(\gamma_t), \dot{\gamma}_t\rangle dt \\
&= L_G(\gamma) + \int_0^1 \frac{d}{dt}\psi(\gamma_t)dt \\
&= L_G(\gamma) + \psi(z_1) - \psi(z_0).
\end{aligned}$$

Consequently, minimizing $L_{\mathcal{F}}$ is equivalent to minimizing L_G. The Randers part of the metric only influences the distance not the trajectory.

C Description of the metrics used

C.1 Metrics expression

We here present the different metrics used to compare our method. We first consider the setup of a deterministic generator. In this setup we consider a VAE $\mathcal{X} \xrightarrow{g} \mathcal{Z} \xrightarrow{f}$, where for a point $x \in \mathcal{X}$, We have $g(x) = (z, \Sigma^{-1})$ and $\hat{x} = f(z + \varepsilon\Sigma^{-1})$ with $\varepsilon \sim \mathcal{N}(0, I)$.

LAND The diagonal elements of the LAND metric of Arvanitidis et al. (2016) are defined as :

$$G_{ii}^{-1}(z) = \sum_k \exp\left(-\frac{\|z - z_k\|_2^2}{2\tau_k^2}\right)(z_k^{(i)} - z^{(i)})^2 + \lambda\,. \tag{29}$$

Here $z^{(i)}$ denotes the i-th coordinates of the points z and $z_k^{(i)}$ the i-th coordinates of the k-th centroid.

LAND RBF This allow to introduce the LANDRBF metric of Kapusniak et al. (2024) :

$$G^{-1}(z) = \text{diag}(h(z)) + \lambda I\,, \tag{30}$$

where :

$$h(z) = \sum_k w_k \exp\left(-\frac{\|z - z_k\|_2^2}{2\tau_k^2}\right)\,. \tag{31}$$

The positive weights are then optimized by minimizing the following loss function :

$$L(\{w_k\}) = \sum_{z_i \in \mathcal{Z}} (h(z_i) - 1)^2\,. \tag{32}$$

CovCentroids The Covariance Centroids metric from Chadebec & Allassonnière (2022) replace the local covariance of the LAND metric by the learned covariances of the encoder :

$$G^{-1}(z) = \sum_k \exp\left(-\frac{\|z - z_k\|_2^2}{2\tau_k^2}\right)\Sigma_k^{-1} + \lambda I \tag{33}$$

PBCentroids We can pullback the geometry of the image space$\mathcal{X}$ to the latent space $\mathcal{Z}$ using the pullback of the metric tensor. Under the hypothesis that the decoder is a local diffeomorphism and that the geometry of the image space is Euclidean, we can then define the pullback metric tensor as follows:

$$G_{\text{pb}}(z) = J_f(z)^\top J_f(z) \tag{34}$$

where $J_f(z)$ is the Jacobian of the decoder at point z. As this is expensive to compute and ill-calibrated to support the data-manifold, we can instead compute the pullback metric tensor at centroids and then interpolate it in the latent space. The cometric is then defined as follows:

$$G^{-1}(z) = \sum_k \exp\left(-\frac{\|z - z_k\|_2^2}{2\tau_k^2}\right) G_{\text{pb}}^{-1}(z_k) + \lambda I \tag{35}$$

For the interested reader, we also provide a broader overview including pullback-based metrics. Whilst these metrics have favorable theoretical properties, they require loads of compute.

WarpedPB The centroid approach is ad-hoc and relies on correct tuning of the temperature parameters τ_k. Instead, it has been proposed by Sun et al. (2025a) to add an additional dimension to the decoder so that we now have $\tilde{f} = (f, s)$. It verifies $s(z) \approx 0$ for z in the data-manifold and $s(z) \gg 0$ for z off the data-manifold. We can then define a cometric as follows:

$$G(z) = J_{\tilde{f}}(z)^\top J_{\tilde{f}}(z) + \lambda I \tag{36}$$

The question is then to correctly design and learn such function s. The authors propose two approaches. First they propose to train a GAN from the latent space to the data space and use the learned

discriminator as a function s. The second approach is to use a Gaussian Process. We thus obtain s as the variance of the GP at point z. The GP here uses a RBF kernel with a temperature parameter τ such that :

$$s(z) = K(z,z) - K(z,Z)\left[K(Z,Z) + \tau I\right]^{-1} K(Z,z) \tag{37}$$

where $Z = [z_1, \cdots, z_n]$; $K(z,Z) = [K(z,z_1), K(z,z_2), \ldots, K(z,z_n)]$; $K(Z,z) = K(z,Z)^\top$; and $K(Z,Z) = [K(z_i,z_j)]_{i,j}$. With :

$$K(z,z') = \exp\left(-\frac{\|z-z'\|_2^2}{2\tau^2}\right) \tag{38}$$

RBF PB We now consider a VAE with a stochastic generator. That means that it now outputs a distribution $\mathcal{X} \xrightarrow[g_\sigma]{g_\mu} \mathcal{Z} \xrightarrow[f_\sigma]{f_\mu}$. We thus have that $z = g_\mu(x) + \varepsilon g_\sigma(x)$ with $\varepsilon \sim \mathcal{N}(0,I)$ and $\hat{x} = f_\mu(z) + \varepsilon f_\sigma(z)$ with $\varepsilon \sim \mathcal{N}(0,I)$.

Such networks are trained in two steps. First we train g_μ,g_σ and f_μ classically using the ELBO loss. Then we freeze g_μ,g_σ and f_μ and train f_σ using the same ELBO loss. The difference with a regular VAE is that instead of using $p(x|z) = \mathcal{N}(f_\mu(z), I)$, we now use $p(x|z) = \mathcal{N}(f_\mu(z), f_\sigma(z))$ as a prior.

The issue is that even if we take the pullback of such networks, we still don't enforce manifold support. Consequently, specific design for f_σ is needed to enforce that the variance is small on the manifold and large off the manifold. Arvanitidis et al. (2018) proposes to use a RBF network for f_σ. That is we instead learn $\beta_\psi = \frac{1}{f_\sigma^2}$ where:

$$\beta_\psi(z) = Wv(z) + \zeta \quad \text{with} \quad v_k(z) = \exp\left(-\frac{\|z-z_k\|_2^2}{2\tau_k^2}\right). \tag{39}$$

Where we use K centroids and positive weights W and bias ζ. Note that we can plug this function, coined *RBF Layer*, as a the first layer of the decoder and then add other layers to learn a more complex function. In any cases, we need to ensure that all of the weights are positive to ensure that $\beta_\psi(z) > 0$ for all z.

The metric is then defined as follows:

$$G(z) = J_{f_\mu}(z)^\top J_{f_\mu}(z) + J_{f_\sigma}(z)^\top J_{f_\sigma}(z). \tag{40}$$

PBIG In González-Duque et al. (2022), the authors adopt an information geometry point of view. As the decoder now outputs a distribution, its output is a statistical manifold naturally equipped with the Fisher-Rao metric. In this paper we only consider that the decoder outputs a multivariate diagonal Gaussian distribution. We consequently have direct access to the Fisher-Rao metric of the model and don't need to really on the KL divergence "black box tools" used in their paper. We still need to have a correctly calibrated estimate of the variance, here we follow the same design as for the **RBG PB** metric. The metric is thus now given by

$$G(z) = J_{f_\mu}(z)^\top \text{diag}\left(\frac{1}{f_\sigma(z_i)^2}\right) J_{f_\mu}(z) + J_{f_\sigma}(z)^\top \text{diag}\left(\frac{2}{f_\sigma(z_i)^2}\right) J_{f_\sigma}(z). \tag{41}$$

C.2 Remark on the pullback metrics

The computation how the pullback metrics is computationally very expensive and scales poorly with the size of the decoder network. To this end we avoid materializing the full dense jacobian as much as possible. When needed, we use finite differences approximation as they prove faster and less expensive than true autodiff because they only require two forward passes. That is, using e_j the j-th standard basis vector and ε as small constant, we have that the j-th column of the jacobian matrix is given by:

$$J_f(z)[:,j] = \frac{f(z+\varepsilon e_j) - f(z-\varepsilon e_j)}{2\varepsilon}. \tag{42}$$

Otherwise the computation of the norm and dot products can be computed using regular Jacobian vector dot products as implemented in automatic differentiation libraries as : $\langle u, v\rangle_{G_{pb}(z)} = \langle u, J^\top(z)J(z)v\rangle = \langle J(z)u, J(z)v\rangle$. In practice, apart from toy networks, this is still too expensive and we resort to materializing the full jacobian matrix using finite differences to compute dot products.

C.3 PARAMETER SELECTION

All of the presented above metrics require to tune some hyper-parameters.

We denoted by λ the hyper-parameter regularizing towards the identity. It is needed to avoid numerical errors and ensure a that the metric yields PSD matrices when evaluated. In practice it is chosen in all experiments as $\lambda = 10^{-3} \ll 1$.

Then these metrics are evaluated on a subset of K centroids that we extract using K-Medoids (Schubert & Lenssen, 2022). To tune the temperature parameters, we build on the approach described in Arvanitidis et al. (2018) and others. We first introduce a new hyper-parameter $a \in \mathbb{R}_*^+$. For $k \in [\![1, K]\!]$, we consider the temperature τ_k associated to the centroids z_k. Denote by C_k the cluster of points whose closest centroids is z_k. We thus have the following quantities :

$$r_k = \text{clamp}\left(\frac{\mid C_k \mid -1}{\text{min_c_size} - 1}, 0, 1\right) \tag{43}$$

$$\text{scale}_k^2 = r_k \times \text{local_scale}_k + (1 - r_k) \times \text{global_scale}_k \tag{44}$$

$$\tau_k^2 = a \times \text{quantile_clip}(\text{scale}_k^2, 0.25, 0.95) \tag{45}$$

The reliability score r_k allows to interpolate between a local scale and a more global scale. This enables to avoid the degenerated cases of very small clusters or high values of K. We also clip the values of scale_k based on the values of the quantiles to further avoid outliers. The value of local_scale_k is given similarly to Arvanitidis et al. (2018) :

$$\text{local_scale}_k = \frac{1}{\mid C_k \mid} \sum_{z_i \in C_k} \|z_i - z_k\|^2 . \tag{46}$$

And the global_scale_k is given by the distance squared to the n_{rank} closest other centroid.

In practice, we pick $\text{min_c_size} = 3$, $n_{\text{rank}} = 5$, and $a = 1$ for all metrics except for **CovCentroids** and **PBCentroids** where we set $a = 5$, .

The temperature of the Gaussian Process for the Warped Pullback metric Sun et al. (2025a) is given by the maximum distance to the second closest centroid.

We find that these heuristics provide consistent results without much of the hassle of tuning these parameters. In particular it doesn't yield large gap in the values of the local probability density $\sqrt{\det G(z)}$ between training and validation samples. We nevertheless acknowledge that it would be of interest to study these heuristics properties in further details.

D WASSERSTEIN GRADIENT FLOW LEARNING

We briefly present the method of Huizing et al. (2026) with their parameter α set to 0. We consider that the data are sampled according to a distribution μ_s. We model the evolution of μ_s as a Wasserstein gradient flow for a potential energy E. That is for a sample z, its Euclidean gradient flow z_s is given by

$$\frac{\mathrm{d}z}{\mathrm{d}s} = -\nabla E(z). \tag{47}$$

As we don't have access to the true potential E, we approximate it using a neural network E_θ. More precisely, using the discretized distribution $(\mu_{s_k})_k$, we aim to match $\mu_{s_{k+1}}$ and $(Id - \tau \nabla E_\theta)_{\#}\mu_{s_k}$. We compare the distribution using the debiased Wasserstein distance $\mathcal{W}_\varepsilon$ (Feydy et al., 2019). Hence the loss is :

$$\mathcal{L}(\theta) = \sum_k (s_{k+1} - s_k) \mathcal{W}_\varepsilon \left(\mu_{s_{k+1}}, (Id - \tau \nabla E_\theta)_{\#}\mu_{s_k}\right) \tag{48}$$

At training time we use $\tau = s_{k+1} - s_k$. At inference, we sample an initial point (z, s) and an arrival $s_{\text{max}} > s$. We then compute the extrapolated trajectory by integrating Eq.47 with E_θ where $\tau = \frac{s_{\text{max}} - s}{T - 1}$; where T is the desired number of time points. In the Shepp-Logan dataset setup, we discretize the distribution by binning the sequential factors in 10 bins of equal size.

E GEODESICS COMPUTATIONS

Geodesics are computed using a two stages approach. For the Riemannian case, first a weighted graph is computed from the embedded data. The edges are computed from a KNN graph using 8 neighbors. The edge $e_{i\to j}$ between z_i and z_j is weighted by the length of $\gamma_{i\to j}(t) = tz_i + (1-t)z_j$ under the metric. At inference we connect the point to the graph using again the length of the straight line to the closest points under the metric G. For the Finsler case, the graph is now directed and we can have different weights between $e_{i\to j}$ and $e_{j\to i}$, where the distance is computed using g. This yields a first geodesic approximation using Dijkstra's algorithm. For sufficiently dense uniform random graph, this first approximation converges towards the true geodesic (Kim et al., 2024).

We then proceeded differently for diagonal or dense metrics. Dense metrics induce prohibitive computational cost that prevent using sophisticated algorithms. This is further discouraged on pullback metric where the optimization process might need to differentiate through the pull-backed tensor, inducing additional instabilities.

For the diagonal metrics, we refine the first geodesic approximation from the graph approach using the GEORCE algorithm from Rygaard & Hauberg (2025).

For dense metric, we use the first geodesic approximation from the graph approach to define a BSline. We then optimize its control points with the Adam optimizer ($n_{\text{epochs}} = 500, lr = 0.01$) by minimizing the curve's energy.

These different approaches enable to have comparable runtime when evaluating different frameworks.

F ABLATIONS

In this section we investigate several choice of parameter or parametrization done in the main body of the paper; namely, the polar decomposition of the *1-form* ω, the robustness of the framework to noisy labels, and the hyper-parameters of the extrapolation.

F.1 RANDERS METRIC PARAMETRIZATION

Here we verify that the use of the polar decomposition of ω from Eq.7 is useful. To check this we compare the setting where ρ_θ is learned or fixed to 1. We display the results in Table 3. The polar decomposition doesn't appear to significantly increase performances. This is actually to be expected as the accuracy of the learned hyperplane is already really close to 1 across the latent space for all base metric. For we have on the test set that $\rho_\theta(z) = 0.873 \pm 0.009$ for the *LAND* metric or even 0.9999 ± 0.0004 for the *LAND RBF* metric. We nonetheless expect that this decomposition could prove useful in some settings.

Table 3: Ablation study on the amplitude parametrization. Here, $d = 64, \alpha = 5.0, \eta = 0.999$, and $\zeta = 0.9$.

		Interpolation		Extrapolation	
Base metric	Learned ρ_θ	Kendall Tau (↑)	FWD (↓)	Kendall Tau (↑)	Relative Variation (%)
Euclidean	✗	1.0 ± 0.02	0.41	1.0 ± 0.02	85.59 ± 33.52
	✓	1.0 ± 0.02	0.41	1.0 ± 0.02	85.59 ± 33.52
LAND [1]	✗	0.46 ± 0.29	0.19	1.0 ± 0.0	90.00 ± 44.26
	✓	0.46 ± 0.29	0.19	1.0 ± 0.0	90.00 ± 44.26
LAND RBF [29]	✗	0.46 ± 0.3	0.22	1.0 ± 0.02	83.47 ± 30.76
	✓	0.46 ± 0.3	0.22	1.0 ± 0.02	83.47 ± 30.76
CovCentroids [9]	✗	0.45 ± 0.29	0.2	1.0 ± 0.03	74.53 ± 74.71
	✓	0.45 ± 0.29	0.21	1.0 ± 0.04	83.33 ± 80.28
PBCentroids	✗	0.77 ± 0.31	0.23	0.59 ± 0.55	28.75 ± 70.48
	✓	0.63 ± 0.39	0.24	0.37 ± 0.66	15.32 ± 65.99

F.2 NOISE ROBUSTNESS

We investigate how robust our introduced framework is robust to noise. Consequently we consider a corrupted dataset $\mathcal{D} = \{x_i, z_i, \tilde{s}_i\}$ where the sequential factors are corrupted. More specifically we have that $\tilde{s}_i = s_i + c \times \varepsilon$ with c the label noise parameter, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, and σ^2 the empirical variance of the training sequential factors.

The results in Table 4 demonstrate that the method is robust to small noise perturbation. For the interpolation task, the monotonicity score maintains an approximately constant level for $c < 1$. It degrades for higher values. In the extrapolation setting, albeit the Kendall Tau remains high, the latent space actually traversed drops quickly as demonstrated by the sharp decrease in relative variation.

Table 4: Ablation study on the amplitude parametrization. Here, $d = 64, \alpha = 5.0, \eta = 0.999$, and $\zeta = 0.9$.

		Interpolation		Extrapolation	
Base metric	Label Noise	Kendall Tau (↑)	FWD (↓)	Kendall Tau (↑)	Relative Variation (%)
Euclidean	0.0	0.95 ± 0.24	0.4	1.0 ± 0.02	85.81 ± 32.90
	0.1	0.98 ± 0.14	0.35	1.0 ± 0.01	91.93 ± 44.27
	0.5	0.89 ± 0.38	0.38	0.99 ± 0.08	55.37 ± 25.70
	1.0	0.67 ± 0.69	0.41	1.0 ± 0.03	30.65 ± 20.89
	2.0	0.44 ± 0.86	0.41	0.94 ± 0.27	13.43 ± 7.88
	5.0	0.08 ± 0.96	0.35	0.65 ± 0.64	5.84 ± 5.72
LAND [1]	0.0	0.44 ± 0.3	0.21	1.0 ± 0.03	88.98 ± 42.75
	0.1	0.47 ± 0.3	0.23	1.0 ± 0.03	94.08 ± 49.42
	0.5	0.42 ± 0.34	0.18	1.0 ± 0.04	76.00 ± 28.60
	1.0	0.31 ± 0.41	0.21	1.0 ± 0.05	82.99 ± 53.65
	2.0	0.21 ± 0.45	0.2	0.99 ± 0.11	47.10 ± 26.35
	5.0	0.05 ± 0.49	0.19	0.98 ± 0.16	23.01 ± 9.94
LAND RBF [29]	0.0	0.49 ± 0.3	0.25	1.0 ± 0.02	85.89 ± 33.61
	0.1	0.42 ± 0.31	0.19	1.0 ± 0.02	85.49 ± 34.21
	0.5	0.41 ± 0.34	0.18	1.0 ± 0.01	59.06 ± 21.57
	1.0	0.31 ± 0.4	0.22	0.99 ± 0.11	34.98 ± 12.43
	2.0	0.21 ± 0.45	0.22	0.92 ± 0.36	16.54 ± 7.85
	5.0	0.05 ± 0.5	0.2	0.75 ± 0.59	9.39 ± 8.15
CovCentroids [9]	0.0	0.43 ± 0.32	0.17	0.48 ± 0.56	21.30 ± 82.48
	0.1	0.45 ± 0.32	0.22	0.33 ± 0.59	9.75 ± 87.14
	0.5	0.41 ± 0.33	0.16	0.83 ± 0.39	81.04 ± 115.09
	1.0	0.32 ± 0.41	0.15	0.79 ± 0.43	74.41 ± 109.27
	2.0	0.21 ± 0.45	0.18	0.67 ± 0.53	48.50 ± 77.56
	5.0	0.04 ± 0.49	0.19	0.6 ± 0.55	36.92 ± 84.42
PBCentroids	0.0	0.65 ± 0.33	0.16	0.95 ± 0.25	64.57 ± 32.54
	0.1	0.63 ± 0.34	0.17	1.0 ± 0.05	65.43 ± 25.77
	0.5	0.68 ± 0.37	0.19	0.98 ± 0.16	57.61 ± 27.85
	1.0	0.52 ± 0.52	0.17	0.93 ± 0.31	32.47 ± 30.76
	2.0	0.34 ± 0.63	0.16	0.94 ± 0.22	13.75 ± 13.48
	5.0	0.11 ± 0.69	0.14	0.91 ± 0.28	12.50 ± 15.93

F.3 EXTRAPOLATION HYPER-PARAMETERS

The extrapolation under Finsler Geometry relies on a correct tuning of the target concentration parameter η, the norm of the speed vector α and the exponential moving average coefficient ζ. We display in Figure 3 and 4 the variation of the monotonicity score and the relative variation when varying the parameters. We only include the results on asymmetric frameworks. The results demonstrate that the momentum sampling parameter ζ has little influence on the results. It also

demonstrates that even low values of η yield sequentially consistent trajectories, but to obtain a better traversal of the latent space, a higher value of α is needed.

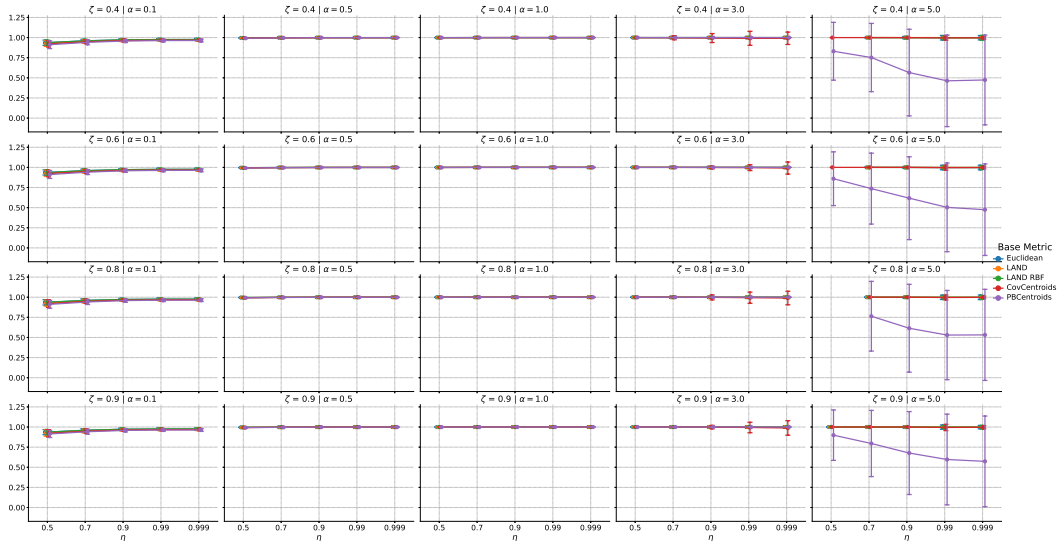


Figure 3: **Monotonicity of extrapolated trajectories**: Evaluation of the extrapolation framework across different hyper-parameters. Here the latent dimension is 64. The error bar correspond to the standard deviation across the computed trajectories with 25 times points.

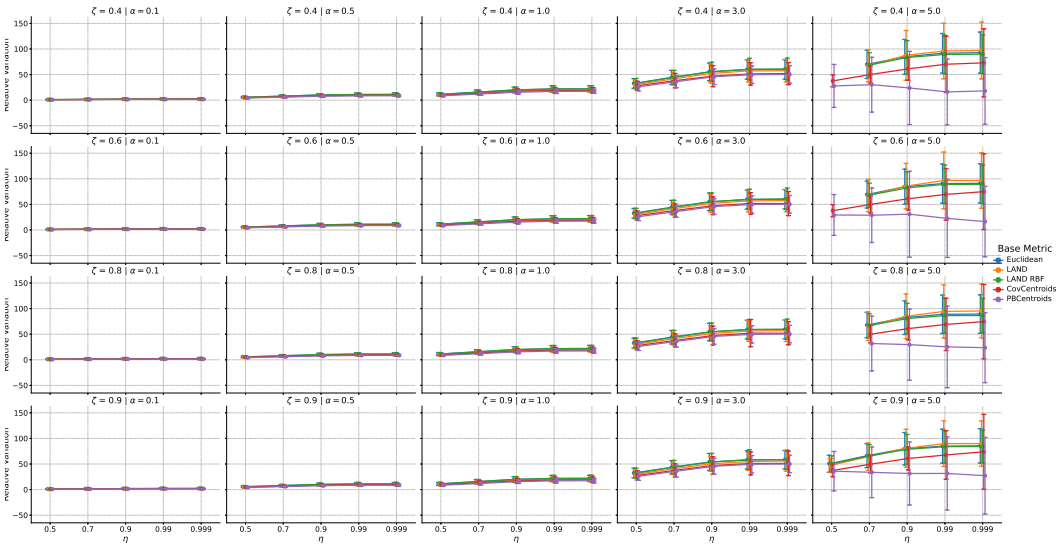


Figure 4: **Relative variation of extrapolated trajectories**: Evaluation of the extrapolation framework across different hyper-parameters. Here the latent dimension is 64. The error bar correspond to the standard deviation across 500 computed trajectories with 25 times points.

G Concentration parameter tuning

In the context of extrapolation, we need to carefully tune the choice of the initial speed. As we designed the metric so that the expected direction of evolution is in the direction of $-\omega$, we sample accordingly using a von Mises Fisher distribution. To calibrate its concentration parameter, we aim to choose it so that we control the average cosine similarity between the sampled speed and the direction of the vector field ω. Let $x \sim \text{vMF}(\mu, \kappa)$ for some unit vector $\mu \in \mathbb{S}^{d-1}$ the mean

direction and concentration parameter $\kappa \in \mathbb{R}^+$. For some target value η, we aim to find κ such that $\eta = \mathbb{E}_x[\text{cosim}(\mu, x)]$. We denote f the density of the distribution such that :

$$f(x) = C_d(\kappa) \exp(\kappa \mu^\top x) \tag{49}$$

$$\text{with } C_d(k) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} I_{\frac{d}{2}-1}(\kappa)} \tag{50}$$

Here I is the modified Bessel function of the first kind at order $\frac{d}{2} - 1$. As both, $\|x\|_2 = 1$ and $\|\mu\|_2 = 1$, the cosine similarity between x and μ is given by $\text{cosim}(\mu, x) = \mu^\top x$.

We make use of the log partition trick. Note:

$$Z(\kappa) = \frac{1}{C_d(\kappa)} = \int_{\mathbb{S}^{d-1}} \exp(\kappa \mu^\top x) \mathrm{d}\sigma(x) \tag{51}$$

Then because :

$$\frac{\mathrm{d}Z}{\mathrm{d}\kappa} = \int_{S^{d-1}} \mu^\top x \exp(\kappa \mu^\top x) \mathrm{d}\sigma(x) \tag{52}$$

And $\log(C_d) = \frac{d-2}{2} \log(\kappa) - \log I_{\frac{d-2}{2}}(\kappa) - \frac{d}{2} \log(2\pi)$, we have the expected cosine similarity:

$$\mathbb{E}[\mu^\top x] = Z(\kappa) \frac{\mathrm{d}Z}{\mathrm{d}\kappa} \tag{53}$$

$$= \frac{\mathrm{d} \log Z}{\mathrm{d}\kappa} = -\frac{\mathrm{d} \log C_d}{\mathrm{d}\kappa} \tag{54}$$

$$= \frac{2-d}{2\kappa} + \frac{I'_{\frac{d}{2}-1}(\kappa)}{I_{\frac{d}{2}-1}(\kappa)} \tag{55}$$

$$= \frac{2-d}{2\kappa} + \frac{I_{\frac{d}{2}}(\kappa) + \frac{d-2}{2\kappa} I_{\frac{d}{2}-1}}{I_{\frac{d}{2}-1}(\kappa)} \tag{56}$$

$$= \frac{I_{\frac{d}{2}}(\kappa)}{I_{\frac{d}{2}-1}(\kappa)} \tag{57}$$

Where we used the fact that $I'_p(v) = \frac{p}{v} I_p(v) + I_{p+1}(v)$.

Consequently for a target cosine similarity $\eta \in [0, 1]$, we solve, for κ, $\eta = \frac{I_{\frac{d}{2}}(\kappa)}{I_{\frac{d}{2}-1}(\kappa)}$. Following the work of Banerjee et al. (2005), we use the approximation of the ratio of Bessel functions so that

$$\kappa = \frac{\eta d - \eta^3}{1 - \eta^2} \tag{58}$$

H EXPERIMENTS DETAILS

Unless mentioned otherwise we use the same set of hyper-parameters for each experiments. For more details, consult the available source code. The Shepp-Logan images are 256×256 black and white images in the range $[0, 1]$.

Encoder The encoder downsamples the input through five convolution blocks: Conv(1→16), Conv(16→32), Conv(32→64), Conv(64→128), and Conv(128→256). The first four blocks use stride-2 convolutions with 3×3 kernels, reducing spatial dimensions from 256×256 to 16×16. The final block uses stride-1 to preserve spatial resolution. Each block applies instance normalization with affine parameters and PReLU activation. The resulting 256×16×16 feature map is flattened and projected through separate linear layers to produce the mean μ and log-variance $\log \sigma^2$ of the latent distribution.

Decoder The decoder maps the latent code back to image space through a symmetric upsampling path. A linear layer projects the latent vector to 256×16×16 dimensions, followed by five upsampling blocks

that restore the spatial resolution to 256×256. Each upsampling block consists of nearest-neighbor 2× upsampling, a 3×3 convolution, instance normalization (without affine parameters), and PReLU activation. The final layer applies a 3×3 convolution followed by sigmoid activation to produce the reconstruction.

Randers metric Both the direction and amplitude of the Randers metric ω are MLP with two layers and SiLU activations. The amplitude regressor part is fed through a sigmoid layer to ensure that the results stay in $[0, 1]$, hence keeping the positivity of the metric.

Regressor In the SheppLogan experiments, for the regressor we use a two stage approach. First we directly segment the image using morphological operations to retrieve the top left ellipsis. Once segmented, we can fit an ellipsis and then compute its area. However when the geodesics cross regions of the latent space where the decoder yields less well reconstructed images, the morphological segmentation fails. In such cases we train a *resnet18* networks to directly predict the value of interest. The morphological segmentation yields more precise results on correctly decoded images, hence this pipeline.

Training recipe The VAE is trained using a combination of the regular ELBO-loss and the MS-SSIM loss from Zhao et al. (2015) for improved reconstruction. We train for 3000 epochs with a cosine learning rate schedule. The training and evaluation are done using a single Nvidia Ada 6000.

I PSEUDO CODE

In this section we summarize the algorithms needed to learn the Randers metric and to evaluate the interpolated and extrapolated trajectories. Note that the user is free to choose the initial encoder/decoder as well as the base metric G. We advise to refer to the code and the paper for the GEORCE geodesic solver Rygaard & Hauberg (2025). For completeness we reproduce the pseudo code for the Riemannian solver.

Algorithm 1 Learning $\tilde{\phi}_\theta$

Require: $\{(z_i, s_i)\}_{i=1}^N$: latent points and their associated sequential factors. G : the Riemannian metric
  Initialize $\tilde{\phi}_\theta$ randomly
  while not converged **do**
    for Batch of points $\{z_i, s_i\}_{i=1}^B$ **do**
      for each z_i in the batch **do**
        $y_i^j = \text{sign}(s_j - s_i)$ for $z_j \in \text{KNN}(z_i)$ ▷ Local pseudo-labels
        $\tilde{l}_i^j = \langle \tilde{\phi}_\theta(z_i), (z_j - z_i) \rangle_{G(z_i)}$
        $l_\phi(z_i) = \frac{1}{K_\phi} \sum_j \max(0, 1 - y_i^j \times \tilde{l}_i^j)$ ▷ Local hinge loss
      end for
      Update $\tilde{\phi}_\theta$ using $\frac{1}{B} \sum_{i=1}^B l_\phi(z_i)$
    end for
  end while

Algorithm 2 Learning ρ_θ

Require: $\{(z_i, s_i)\}_{i=1}^N$: latent points and their associated sequential factors, $\tilde{\phi}_\theta$: learned direction field, G : the Riemannian metric
Initialize ρ_θ randomly
while not converged **do**
  for Batch of points $\{z_i, s_i\}_{i=1}^B$ **do**
    for each z_i in the batch **do**
      $y_i^j = \text{sign}(s_j - s_i)$ for $z_j \in \text{KNN}(z_i)$ ▷ Local pseudo-labels
      $\tilde{l}_i^j = \langle \tilde{\phi}_\theta(z_i), (z_j - z_i) \rangle_{G(z_i)}$
      $\text{acc}_\phi(z_i) = \frac{1}{K} \sum_j 1[y_i^j \times \tilde{l}_i^j > 0]$ ▷ Hyperplane accuracy
      $l_\rho(z_i) = \|\rho_\theta(z_i) - 2 \times \max(\text{acc}_\phi(z_i) - 0.5)\|^2$ ▷ Regression loss
    end for
    Update ρ_θ using $\frac{1}{B} \sum_{i=1}^B l_\rho(z_i)$
  end for
end while

Algorithm 3 Randers metric construction

Require: $(z, v) \in T\mathcal{Z}$: point and tangent vector, $\tilde{\phi}_\theta$: learned direction field, ρ_θ : learned amplitude regressor, G : base metric, ϵ : small positive constant to ensure positivity
$\phi_\theta(z) = \frac{\tilde{\phi}_\theta(z)}{\|\tilde{\phi}_\theta(z)\|_{G^{-1}(z)} + \epsilon}$ ▷ Normalize direction field
$\omega_\theta(z) = \rho_\theta(z) \times \phi_\theta(z)$ ▷ Polar form of ω
$\mathcal{F}(z, v) = \sqrt{v^T G(z) v} + \langle \omega_\theta(z), v \rangle$ ▷ Randers metric definition

Algorithm 4 Geodesic evaluation

Require: $\{(z_i, s_i)\}_{i=1}^N$: latent points and their associated sequential factors, $\mathcal{F}$: learned Finsler-Randers metric, g : decoder, r : regressor to retrieve sequential factors from images
for each pairs (z_0, z_1) such that $s_0 < s_1$ **do**
  $\gamma_t = \text{Geodesic}(\mathcal{F}, z_0, z_1)$ ▷ Compute geodesic between z_0 and z_1 under metric $\mathcal{F}$
  $\hat{x}(t) = g(\gamma_t)$ ▷ Decode the geodesic to get images
  $\hat{s}(t) = r(\hat{x}(t))$ ▷ Retrieve sequential factors from the decoded images
  Compute $m(\hat{s}(t))$ ▷ Compute monotonicity score of the retrieved sequential factors
end for

Algorithm 5 GeneralizedImplicitLeapfrog

Require: H : hamiltonian function to integrate, (q_0, p_0) : initial position and momentum, L: number of integration step, γ : step size.
$k = 0$
$q_k, p_k = q_0, p_0$
while $k < L$ **do**
  $p_{k+\frac{1}{2}} = p_k - \gamma \nabla_q H(q_k, p_{k+\frac{1}{2}})$ ▷ Half update momentum (implicit equation)
  $q_{k+1} = q_k + \frac{\gamma}{2} \left(\nabla_p H(q_k, p_{k+\frac{1}{2}}) + \nabla_p H(q_{k+1}, p_{k+\frac{1}{2}}) \right)$ ▷ Update position (implicit equation)
  $p_{k+1} = p_{k+\frac{1}{2}} - \gamma \nabla_q H(q_{k+1}, p_{k+\frac{1}{2}})$ ▷ Update momentum
  $k = k + 1$
end while

Algorithm 6 Extrapolation computation

Require: $z_0 \in \mathcal{Z}$: start latent point, $\mathcal{F}^*$: learned dual Finsler-Randers metric, ω: learned vector field, η : target cosine similarity, α : norm of the initial speed, ζ : EMA parameter of the momentum, T : number of points.

$\kappa = \frac{\eta d - \eta^3}{1-\eta^2}$ ▷ Eq. 58
$q_t = z_0$
$p_t = 0$
for $t \in [\![1\ ;\ T]\!]$ **do**
  $v_t = \alpha \times \text{vMF}(-\frac{\omega(q_t)}{\|\omega(q_t)\|_2}, \kappa)$ ▷ Sample initial speed
  $\tilde{p}_t = g(q_t, v_t) v_t$ ▷ Legendre transform
  $p_t = (1-\zeta)\tilde{p}_t + \zeta p_t$ ▷ Exponential Moving Average of the momentum
  $q_t, p_t = \text{GeneralizedImplicitLeapfrog}(\frac{1}{2}\mathcal{F}_*^2, q_t, p_t, L = 1, \gamma = \frac{1}{T})$
end for

Algorithm 7 GEORCE for Riemannian manifolds

Input: tol, T
Output: Geodesic estimate $x_{0:T}$
Set $x_t^{(0)} \leftarrow a + \frac{b-a}{T}t$, $u_t^{(0)} \leftarrow \frac{b-a}{T}$ for $t = 0., \ldots, T-1$ and $i \leftarrow 0$
while $\|\nabla_y E(y) \mid_{y=x_t^{(i)}} \|_2 > \text{tol}$ **do**
  $G_t \leftarrow G\left(x_t^{(i)}\right)$ for $t = 0, \ldots, T-1$
  $\nu_t \leftarrow \nabla_y \left(u_t^{(i)} G\left(y\right) u_t^{(i)}\right) \mid_{y=x_t^{(i)}}$ for $t =, 1 \ldots, T-1$
  $\mu_{T-1} \leftarrow \left(\sum_{t=0}^{T-1} G_t^{-1}\right)^{-1} \left(2(a-b) - \sum_{t=0}^{T-1} G_t^{-1} \sum_{t>j}^{T-1} \nu_j\right)$
  $\tilde{u}_t \leftarrow -\frac{1}{2} G_t^{-1} \left(\mu_{T-1} + \sum_{j>t}^{T-1} \nu_j\right)$ for $t = 0, \ldots, T-1$
  Using line search find α^* for the following optimization problem with the discrete energy functional E

$$\alpha^* = \underset{\alpha}{\text{argmin}} \quad E\left(\tilde{x}_{0:T}\right) \quad \text{(exact line search)}$$
$$\text{s.t.} \quad \tilde{x}_{t+1} = \tilde{x}_t + \alpha \tilde{u}_t + (1-\alpha) u_t^{(i)}, \quad t = 0, \ldots, T-1,$$
$$\tilde{x}_0 = a.$$

  Set $u_t^{(i+1)} \leftarrow \alpha^* \tilde{u}_t + (1-\alpha^*) u_t^{(i)}$ for $t = 0, \ldots, T-1$
  Set $x_{t+1}^{(i+1)} \leftarrow x_t^{(i+1)} + u_t^{(i+1)}$ for $t = 0, \ldots, T-1$
  $i \leftarrow i + 1$
end while
return x_t for $t = 0, \ldots, T-1$

J Interpolation and Extrapolation example

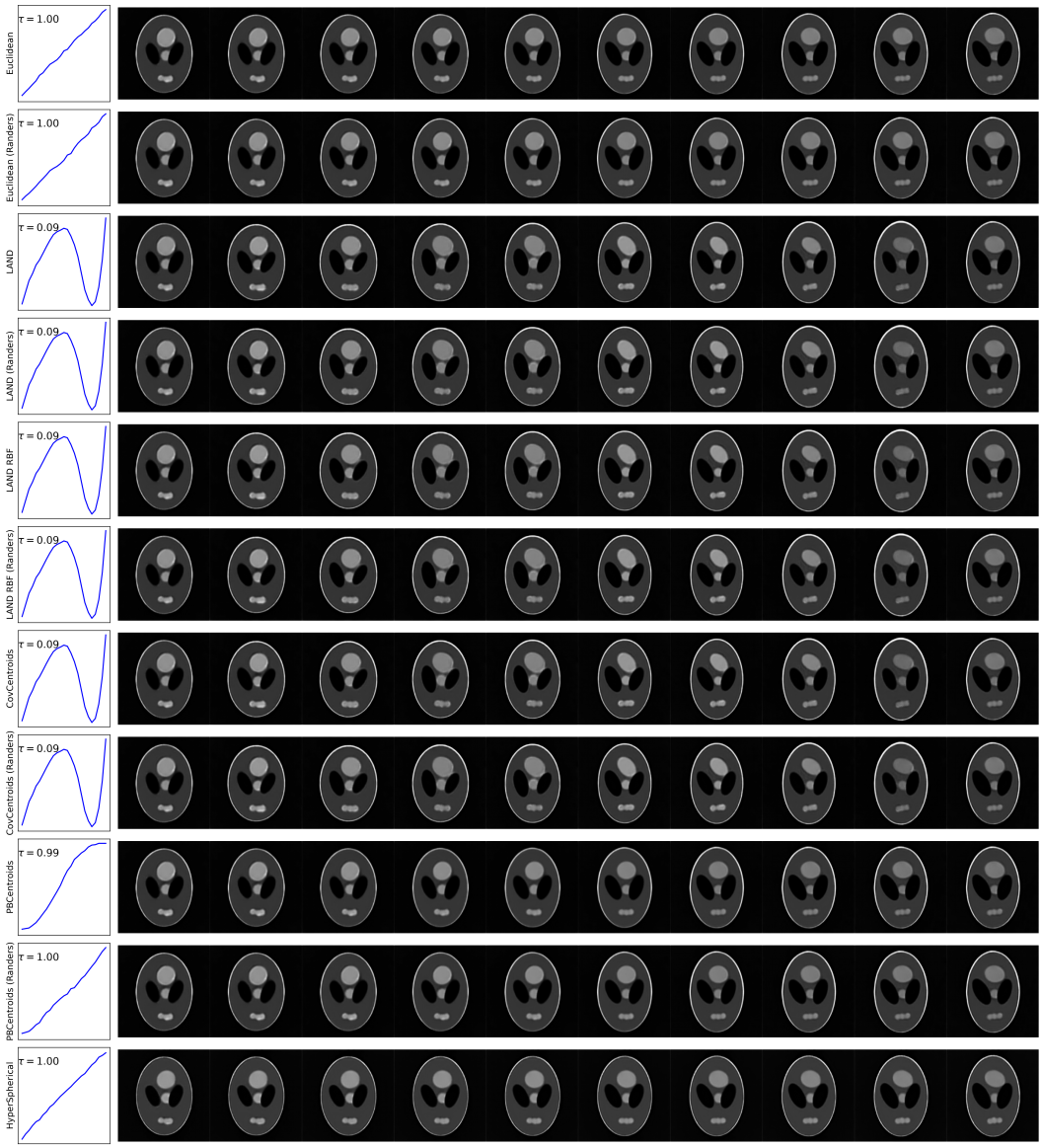


Figure 5: Example of interpolation with different framework with fixed endpoints on the Shepp-Logan dataset . Left columns show the predicted sequential factors along the trajectory.

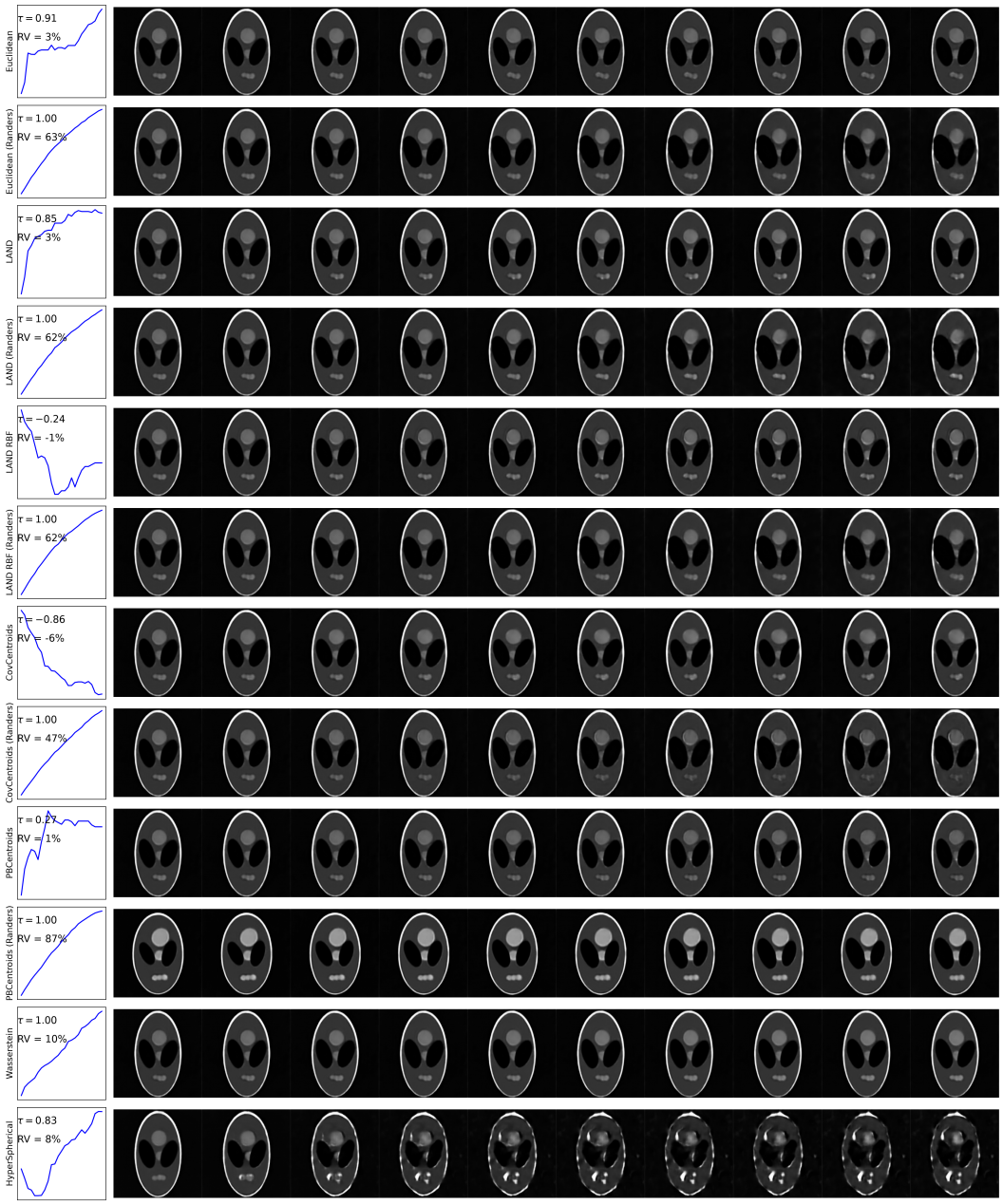


Figure 6: Example of extrapolation with different framework on the Shepp-Logan dataset. Left columns show the predicted sequential factors along the trajectory. Here we have $\alpha = 5.0$, $\zeta = 0.9$, and $\eta = 0.999$

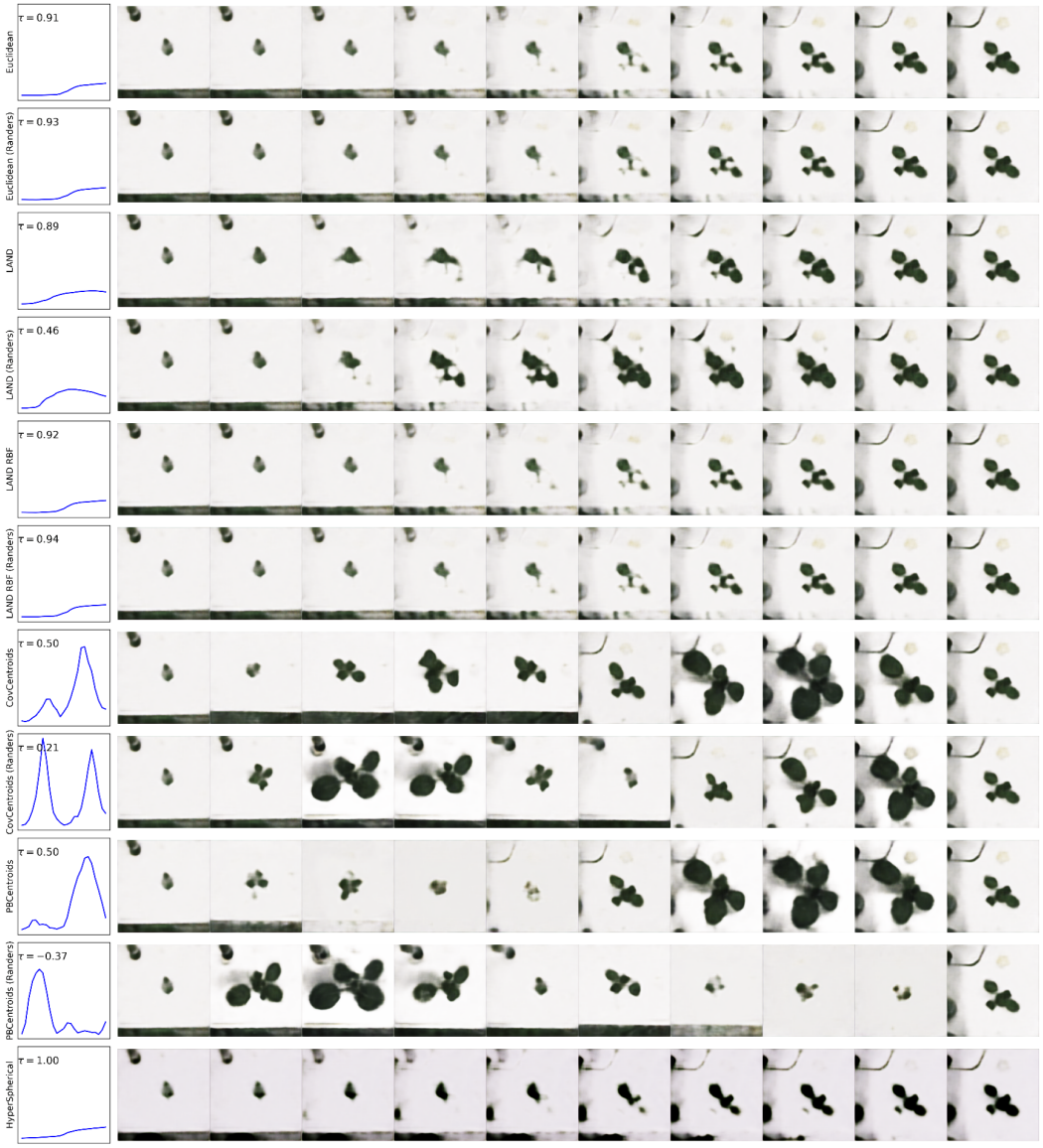


Figure 7: Example of interpolation with different framework with fixed endpoints on the KOMAT-SUNA dataset . Left columns show the predicted sequential factors along the trajectory.

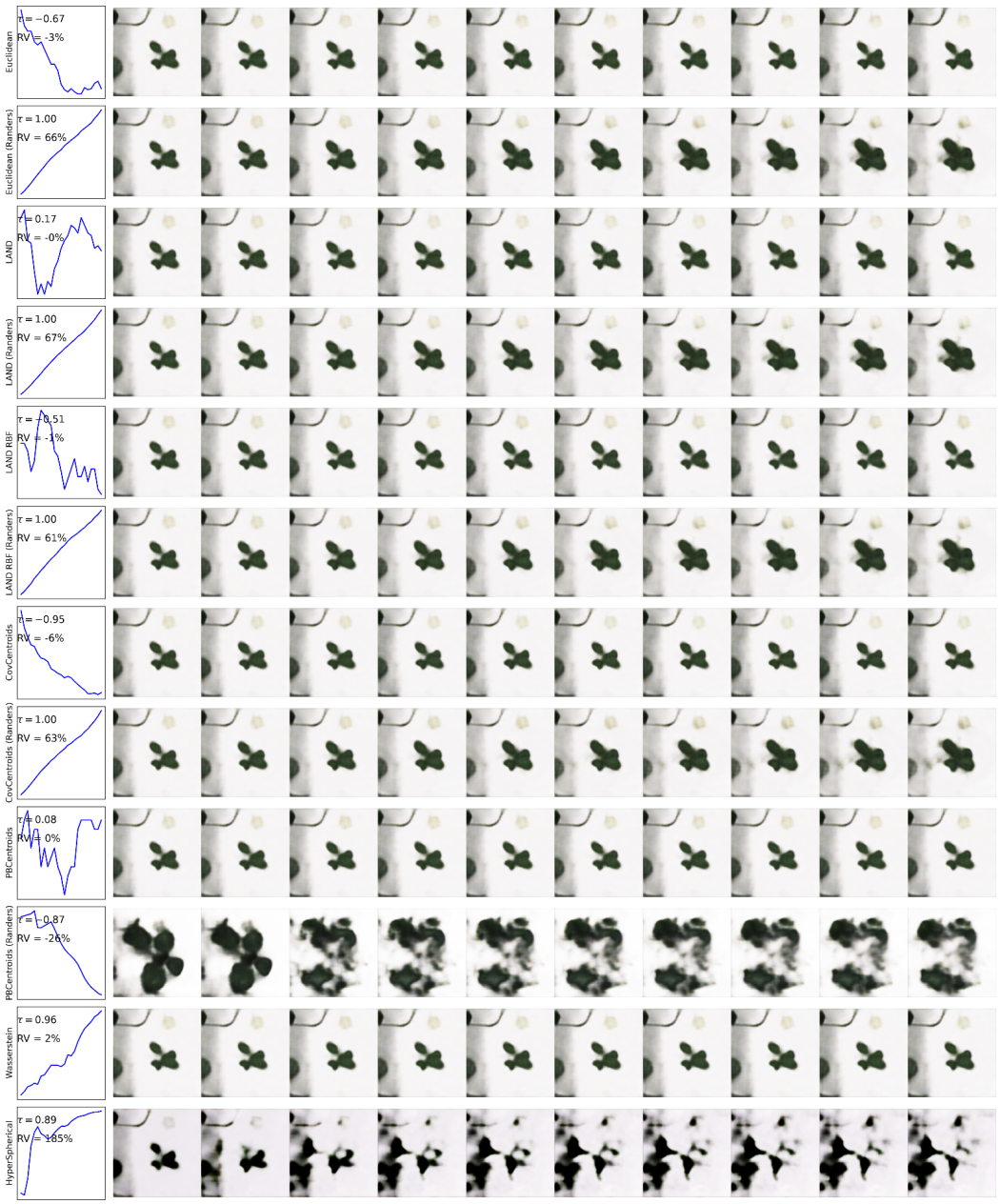


Figure 8: Example of extrapolation with different framework on the KOMATSUNA dataset. Left columns show that predicted sequential factors along the trajectory. Here we have $\alpha = 5.0$, $\zeta = 0.9$, and $\eta = 0.999$