Finsler Multi-Dimensional Scaling: Manifold Learning for Asymmetric Dimensionality Reduction and Embedding

Supplementary Material

This supplemental material is organized as follows:

- **Appendix A** contains the proofs of Theorems 1 and 2 and Propositions 1 and 2 and the details of the derivations for the Finsler stress function (Eq. (7)).
- **Appendices B and C** contain additional theoretical discussions. The former is dedicated to the link between current fields and Randers metrics, while the latter focuses on a generalisation of the Wormhole criterion to Finsler MDS to handle manifolds with missing parts.
- **Appendix D** contains implementation details and additional experiments complementing the visualisation experiments in Sec. 7.1 and the digraph representation learning experiments in Sec. 7.2.

A. Proofs and Derivations

A.1. Proof of Theorem 1

We here provide two proofs of this result. The first uses the Euler-Lagrange equation, a powerful and general tool in the calculus of variations. It can give some insights for generalisation to other metrics. However, given the simplicity of the canonical Randers space, a quick and direct proof is also given.

Euler-Lagrange. In calculus of variations, the Euler-Lagrange equation provides first order optimality necessary conditions on the solution of functionals involving functions $x(t)$ and their derivative $x'(t)$.

Theorem 3 (Euler-Lagrange equation)**.** *If a functional of a smooth scalar function $x(t)$ is given by $\mathcal{L}(x) = \int_0^1 L(t, x(t), x'(t))dt$, where L is a positive smooth function, then the solution minimising the functional $\mathcal{L}$ satisfies the equation*

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial x'} = 0.$$

Many generalisations of the Euler-Lagrange equations exist. In our case, when $x(t) = (x_1(t), \cdots, x_m(t))^\top \in \mathbb{R}^m$ is multi-dimensional, the Euler-Lagrange equation is duplicated for each output dimension. In other words, the minimum solution satisfies the set of equations

$$\frac{\partial \mathcal{L}}{\partial x_i} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial x_i'} = 0 \quad \forall i \in \{1, \cdots, m\}. \tag{11}$$

The Euler-Lagrange equations can be used to derive shortest geodesic paths in our canonical Randers space. The length is a functional (Eq. (1)), that can be rewritten from a Lagrangian perspective as

$$\mathcal{L}_{F^C}(\gamma) = \int_0^1 L\big(t, \gamma(t), \gamma'(t)\big)dt, \tag{12}$$

where

$$L\big(t, \gamma(t), \gamma'(t)\big) = F^C_{\gamma(t)}\big(\gamma'(t)\big) = \|\gamma'(t)\|_2 + \omega^\top \gamma'(t). \tag{13}$$

Denoting $\gamma = (\gamma_1, \cdots, \gamma_m)$ and $\gamma' = (\gamma_1', \cdots, \gamma_m')$, the Euler-Lagrange equations for this functional are given for all $i \in \{1, \cdots, m\}$ by

$$\frac{\partial L}{\partial \gamma_i} - \frac{d}{dt}\frac{\partial L}{\partial \gamma_i'} = 0. \tag{14}$$

Since L does not explicitly depend on γ, but only its derivative, we have that the Euler-Lagrange equations simplify to

$$0 = \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \gamma_i'} \tag{15}$$

$$= \frac{d}{dt}\left(\frac{\gamma_i'(t)}{\|\gamma'(t)\|_2} + \omega_i\right). \tag{16}$$

In the canonical space, ω is a uniform vector field, as such its coordinates ω_i do not depend on t. Thus, $\frac{d}{dt}\omega_i = 0$. We then have, stacking the Euler-Lagrange equations into vector form, that

$$\frac{d}{dt}\left(\frac{\gamma'(t)}{\|\gamma'(t)\|_2}\right) = 0. \tag{17}$$

Equation (17) is the same as the one we would obtain if $\omega \equiv 0$, i.e. if the metric was Riemannian. It is well-known to describe the equation of a straight line. To see this, if we take $t = s$ to be the Euclidean arclength parametrisation, then $\|\gamma'(s)\|_2 = 1$ and then the Euler-Lagrange equation becomes $\frac{d}{ds}\gamma'(s) = 0$, meaning that $\gamma'(s)$ is constant and thus $\gamma(s)$ is a straight Euclidean line. Shortest paths in the canonical Randers space are thus the straight segments as in the Euclidean space, making it a flat space.

Calculation. To better understand the particular structure of the canonical Randers space, we provide an alternative simple proof. Assume without loss of generality that $\omega = \alpha(0, \cdots, 0, 1)^\top$, and denote $\gamma(t) = (x_1(t), \cdots, x_m(t))^\top$.

Then

$$\begin{aligned}\mathcal{L}_{F^C}(\gamma) &= \int_0^1 F_{\gamma(t)}(\gamma'(t))dt \\ &= \int_0^1 \left(\|\gamma'(t)\|_2 + \omega^\top \gamma'(t)\right) dt \\ &= \int_0^1 \|\gamma'(t)\|_2 dt + \alpha \int_0^1 x'_m dt \\ &= \int_0^1 \|\gamma'(t)\|_2 dt + \alpha \int_{x_0}^{x_1} dx \\ &= \int_0^1 \|\gamma'(t)\|_2 dt + \alpha(x_1 - x_0). \end{aligned} \tag{18}$$

The right term is a constant not depending on the curve γ, whereas the left term is the usual functional giving the Euclidean length of the curve γ. Thus, the shortest path in the canonical Randers space is also the shortest path in the Euclidean space, which is given by the Euclidean segment $\gamma(t) = (1-t)x + ty$.

A.2. Proof of Proposition 1

Although the shortest paths are the same in the canonical Randers space and in the Euclidean space, i.e. $\gamma^{F^C}_{x\to y}(t) = (1-t)x + ty$, their lengths are not the same as they depend on the direction of traversal. Since the metric is canonical, it does not depend on the position $\gamma^{F^C}_{x\to y}(t)$. Noticing that $\left(\gamma^{F^C}_{x\to y}\right)'(t) = y - x$, a direct calculation gives

$$\begin{aligned} d_{F^C}(x,y) &= \mathcal{L}_{F^C}\left(\gamma^{F^C}_{x\to y}\right) \\ &= \int_0^1 F^C_{\gamma^{F^C}_{x\to y}(t)}\left(\left(\gamma^{F^C}_{x\to y}\right)'(t)\right)dt \\ &= \int_0^1 \left(\left\|\left(\gamma^{F^C}_{x\to y}\right)'(t)\right\|_2 + \omega^\top \left(\gamma^{F^C}_{x\to y}\right)'(t)\right)dt \\ &= \int_0^1 \left(\|y-x\|_2 + \omega^\top (y-x)\right)dt \\ &= \|y-x\|_2 + \omega^\top(y-x). \quad \square \end{aligned} \tag{19}$$

A.3. Proof of Theorem 2

By assumption, the data can be accurately embedded in the Euclidean space $\mathbb{R}^m$. Denote $\boldsymbol{X} \in \mathbb{R}^m$ this solution, with $d(x_i, x_j) = \boldsymbol{D}_{i,j}$ for all pairs (i,j). Consider now the Finsler MDS problem into the canonical Randers space of dimension $\mathbb{R}^{m+1}$. Without loss of generality, we can assume that ω is along the last coordinate axis. The embedding $\boldsymbol{Y} = [\boldsymbol{X}, 0] \in \mathbb{R}^{N\times(m+1)}$, which is the concatenation of the m-dimensional Euclidean embedding with a last 0 coordinate is the minimal solution. Indeed, since the embedding lies in a hyperplane orthogonal to ω, we have $d_F(x_i, x_j) = d_E(x_i, x_j)$ for all pairs (i,j). Since the Euclidean embedding is accurate, we have $d_{F^C}(x_i, x_j) = \boldsymbol{D}_{i,j}$ for all pairs (i,j). $\square$

A.4. Derivation of Eq. (7)

Plugging into the Finsler stress (Eq. (4)) the canonical Randers distances between embedded points (Eq. (6)), we have

$$\begin{aligned}\sigma^2(\boldsymbol{X}) = &\sum_{i,j} w_{ij}\|x_j - x_i\|_2^2 \\ &+ 2\sum_{i,j} w_{ij}\|x_j - x_i\|_2 \omega^\top (x_j - x_i)^\top \\ &+ \sum_{i,j} w_{ij}(x_j - x_i)\omega\omega^\top (x_j - x_i)^\top \\ &- 2\sum_{i,j} w_{ij}\boldsymbol{D}_{ij}\|x_j - x_i\|_2 \\ &- 2\sum_{i,j} w_{ij}\boldsymbol{D}_{ij}\omega^\top (x_j - x_i)^\top + \sum_{i,j} w_{ij}\boldsymbol{D}_{ij}^2. \end{aligned} \tag{20}$$

As $w_{ij} = w_{ji}$, the second summation term vanishes

$$\begin{aligned}\sigma^2(\boldsymbol{X}) = &\sum_{i,j} w_{ij}\|x_j - x_i\|_2^2 \\ &+ \sum_{i,j} w_{ij}(x_j - x_i)\omega\omega^\top (x_j - x_i)^\top \\ &- 2\sum_{i,j} w_{ij}\boldsymbol{D}_{ij}\|x_j - x_i\|_2 \\ &- 2\sum_{i,j} w_{ij}\boldsymbol{D}_{ij}\omega^\top (x_j - x_i)^\top + \sum_{i,j} w_{ij}\boldsymbol{D}_{ij}^2. \end{aligned} \tag{21}$$

The terms $\sum_{i,j} w_{ij}\|x_j - x_i\|_2^2$ and $\sum_{i,j} w_{ij}\boldsymbol{D_{ij}}\|x_j - x_i\|_2$ are the ones we would obtain in the traditional SMACOF algorithm [44], and can be written, respectively, $\mathrm{tr}(\boldsymbol{X}^\top V \boldsymbol{X})$ and $\mathrm{tr}(\boldsymbol{X}^\top B(\boldsymbol{X})\boldsymbol{X})$, with V and B given by Eq. (8) and Eq. (9).

The terms $\sum_{i,j} w_{ij}(x_j - x_i)\omega\omega^\top(x_j - x_i)^\top$ and $\sum_{i,j} w_{ij}\boldsymbol{D}_{ij}\omega^\top(x_j - x_i)^\top$ are specific to the Randers metric, and can be simply written as $\mathrm{tr}(\boldsymbol{X}^\top V \boldsymbol{X}\omega\omega^\top)$ and $\mathrm{tr}((W^\top \odot \boldsymbol{D}^\top - W \odot \boldsymbol{D})\mathbb{1}_m\omega^\top \boldsymbol{X}^\top)$.

A.5. Proof of Proposition 2

Our proof is based on the *majorisation* approach [32, 44]. Inspired by the traditional SMACOF algorithm, we aim to find a function $g(\cdot,\cdot)$ that satisfies all the following conditions for any points $\boldsymbol{X}$ and $\boldsymbol{Y}$:

(i) $\sigma^2(\boldsymbol{X}) = g(\boldsymbol{X}, \boldsymbol{X})$,
(ii) $\sigma^2(\boldsymbol{X}) \le g(\boldsymbol{X}, \boldsymbol{Y})$ for any $\boldsymbol{Y}$,
(iii) $g(\boldsymbol{X}, \boldsymbol{Y})$ can be easily minimised with respect to $\boldsymbol{X}$ for any $\boldsymbol{Y}$.

For such a function g, the algorithm

$$\boldsymbol{X}^{(k+1)} = \operatorname*{argmin}_{\boldsymbol{X}} g(\boldsymbol{X}, \boldsymbol{X}^{(k)}) \tag{22}$$

decreases the stress at each iteration as

$$g(\boldsymbol{X}^{(k)}, \boldsymbol{X}^{(k)}) \geq g(\boldsymbol{X}^{(k+1)}, \boldsymbol{X}^{(k)}) \geq g(\boldsymbol{X}^{(k+1)}, \boldsymbol{X}^{(k+1)}). \tag{23}$$

Since the stress $\sigma^2(\boldsymbol{X}^{(k)}) = g(\boldsymbol{X}^{(k)}, \boldsymbol{X}^{(k)})$ decreases at each iteration, the algorithm converges to a local minimum (sandwich theorem).

We now look for a suitable function g. From the derivation of the stress function in Eq. (21), we have

$$\sigma^2(\boldsymbol{X}) = \operatorname{tr}(\boldsymbol{X}^\top V \boldsymbol{X}) + \operatorname{tr}(\boldsymbol{X}^\top V \boldsymbol{X} \omega\omega^\top) + 2\operatorname{tr}(C\boldsymbol{X}^\top) - 2\operatorname{tr}(\boldsymbol{X}^\top B(\boldsymbol{X})\boldsymbol{X}), \tag{24}$$

As in the traditional SMACOF [10], the Cauchy-Schwarz inequality implies that

$$\operatorname{tr}(\boldsymbol{X}^\top B(\boldsymbol{X})\boldsymbol{X}) \geq \operatorname{tr}(\boldsymbol{X}^\top B(\boldsymbol{Y})\boldsymbol{Y}) = \mu(\boldsymbol{X}, \boldsymbol{Y}). \tag{25}$$

We thus have

$$\sigma^2(\boldsymbol{X}) \leq g(\boldsymbol{X}, \boldsymbol{Y}), \tag{26}$$

where

$$g(\boldsymbol{X}, \boldsymbol{Y}) = \operatorname{tr}(\boldsymbol{X}^\top V \boldsymbol{X}) + \operatorname{tr}(\boldsymbol{X}^\top V \boldsymbol{X} \omega\omega^\top) + 2\operatorname{tr}(C\boldsymbol{X}^\top) - 2\mu(\boldsymbol{X}, \boldsymbol{Y}). \tag{27}$$

To implement the majorisation update rule (Eq. (22)), we need to compute the gradient of $g(\cdot, \boldsymbol{X}^{(k)})$ (Eq. (27)). We have

$$\nabla_{\boldsymbol{X}} g(\boldsymbol{X}^{(k+1)}, \boldsymbol{X}^{(k)}) = 2V\boldsymbol{X}^{(k+1)} + 2V\boldsymbol{X}^{(k+1)}\omega\omega^\top + 2C - 2B(\boldsymbol{X}^{(k)})\boldsymbol{X}^{(k)}. \tag{28}$$

Since $\boldsymbol{X}^{(k+1)}$ minimises g (Eq. (22)), the first-order optimality conditions lead to

$$2V\boldsymbol{X}^{(k+1)} + 2V\boldsymbol{X}^{(k+1)}\omega\omega^\top + 2C = 2B(\boldsymbol{X}^{(k)})\boldsymbol{X}^{(k)}. \tag{29}$$

The terms in $\boldsymbol{X}^{(k+1)}$ are linear, we can thus pseudo-invert this system of equations to get the update rule. Recall how to rewrite a linear system of equations to only have the unknowns to the right of the coefficient matrix.

Lemma 1. *For any matrices A, X, and C, we have*

$$AXB = C \iff (B^\top \otimes A)\operatorname{vec}(X) = \operatorname{vec}(C).$$

Applying this rewrite to the linear system of equations Eq. (29), we get the desired update rule

$$\operatorname{vec}(\boldsymbol{X}^{(k+1)}) = K^\dagger \operatorname{vec}(B(\boldsymbol{X}^{(k)})\boldsymbol{X}^{(k)} - C), \tag{30}$$

where $K = (I_m + \omega\omega^\top) \otimes V$ is a Kronecker matrix.

B. The Relationship between Current Fields and Randers Metrics

The search of shortest-time trajectories in a medium with time-independent wind is an old problem first studied by Ernst Zermelo [111] and is called the *Zermelo navigation problem*. In fact, it has turned out to be such an important question that it can be used to explain causality in space-time [17]. In the presence of wind $v(x)$, unit balls of the Finsler metric F_x are offset by $v(x)$. To remain in a Finsler space, where 0 is inside unit balls, the wind must have a small magnitude $F_x(-v(x)) < 1$. Note that in the presence of large winds, the wind implies irreversible displacements, explaining the irreversibility of time and causality in the world. However, the obtained metric in large winds is no longer a Finsler metric.

Consider the traditional case of a Riemannian manifold $\mathcal{X}$. For notational simplicity, we will drop the explicit dependence on x. The Riemannian metric is written as $R(u) = \|u\|_M$. Consider a wind with small magnitude $\|v\|_{M^{-1}} < 1$. The Zermelo metric F, which provides the Finsler metric measuring the traversal time of agents along curves on $\mathcal{X}$ with wind v is given by the equation [92]

$$R\left(\frac{u}{F(u)} - v\right) = 1. \tag{31}$$

Solving this equation with respect to $F(u)$ yields the Zermelo metric given by

$$F(u) = \|u\|_{M_v} + \omega_v^\top u, \tag{32}$$

where

$$M_v = \frac{1}{(1 - \|u\|_M^2)^2}\Big(Mvv^\top M + (1 - \|v\|_M^2)M\Big), \tag{33}$$

$$\omega_v = -\frac{1}{1 - \|v\|_M^2} Mv. \tag{34}$$

The Zermelo metric is thus a Randers Finsler metric. In particular, note that for the traditional isotropic Riemannian metric with $M = I$, and a small current $\|v\|_2^2 \ll 1$, then $M_v \approx M$ and the Randers drift component becomes $\omega_v \approx -v$. As we work on synthetic current data with $M = I$, we make the simplifying approximation when computing the Zermelo-Randers metric that it is given by $F(u) = \|u\|_2 - v^\top u$. Thus our Randers linear drift component is given by the opposite of the current field.

C. Wormhole Finsler MDS

Our Finsler MDS formulation allows to use non-uniform weights $w_{i,j}$ in the Finsler stress function, similar to regular MDS approaches. Here, we focus on generalising the recent state-of-the-art method WHCIE [12] for computing theoretically guaranteed consistent pairs of points on manifolds sampled with missing parts. It was originally motivated for improving unsupervised shape matching to handle partial shapes by filtering out inconsistent pairs from the Gromov-Wassertein loss [11]. We first present the existing approaches in Riemannian manifolds and then focus on our generalisation to Finsler manifolds.

Riemannian wormhole criterion. Let $\tilde{\mathcal{X}}$ be a Riemannian data manifold (without missing parts) and $\tilde{\mathcal{Y}} \subset \tilde{\mathcal{X}}$ be a version of the data manifold that is missing some parts $\tilde{\mathcal{Y}} \neq \tilde{\mathcal{X}}$. Let $\tilde{\boldsymbol{X}}$ be sampled data on $\tilde{\mathcal{Y}}$ (and thus also on $\tilde{\mathcal{X}}$). Data dis-

similarities are computed as shortest path distances. However, depending on whether we are given the full manifold $\tilde{\mathcal{X}}$ or the partial one $\tilde{\mathcal{Y}}$, the computed data dissimilarities $D_{\tilde{\mathcal{X}}}$ and $D_{\tilde{\mathcal{Y}}}$, computed respectively on $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$, might differ. This is due to the fact that geodesic trajectories in the full manifold $\tilde{\mathcal{X}}$ might pass through missing parts of $\tilde{\mathcal{Y}}$ making shortest paths on $\tilde{\mathcal{Y}}$ longer for some pairs of points. As the data dissimilarities differ, optimising the stress function with each of them and using the same uniform weight scheme $w_{i,j} = 1$ for all pairs will lead to different embeddings $\boldsymbol{X} \neq \boldsymbol{Y}$. The objective here is to design a different strategy on the weights $w_{i,j}$ such that the resulting embeddings are as close as possible $\boldsymbol{X} \approx \boldsymbol{Y}$, meaning that the scheme is robust to missing parts.

Pairs of points x_i and x_j are said to be consistent if their shortest path distances $(D_{\tilde{\mathcal{X}}})_{i,j} = (D_{\tilde{\mathcal{Y}}})_{i,j}$ are consistent on the full $\tilde{\mathcal{X}}$ and partial $\tilde{\mathcal{Y}}$ shapes. In practice, a majority of pairs is consistent [12], but a significant amount of pairs are inconsistent, leading to incorrect geodesic dissimilarity estimates in the partial case affecting the embedding. To mitigate this effect, a natural approach is to filter out inconsistent pairs my masking out their contribution to the stress function. This translates to choosing a weight scheme $w_{i,j} \in \{0, 1\}$, with $w_{i,j} = 1$ only for consistent pairs. One way of proceeding is to use heuristics for short distance computations and focus only on local pairs [87]. More recently, another paradigm has shown impressive results [12, 79]. Rather than focusing on local pairs, the idea is to design a criterion that can guarantee whether a pair is consistent. Guaranteeing means that there is theoretically no false alarm possible by the criterion: only consistent pairs are found. More general criteria find more consistent pairs, allowing the method to use more non-perturbed information to find the embedding.

A common misconception is to believe that shortest paths not intersecting the boundary $\tilde{\mathcal{B}} = \delta\tilde{\mathcal{Y}}$ provide consistent pairs, as was debunked in [12]. Rather than focusing on the intersection with the boundary of the partial manifold, the distances to the boundary were used to define the criteria. Let $\tilde{x}_{i_{\tilde{\mathcal{B}}}}$ and $\tilde{x}_{j_{\tilde{\mathcal{B}}}}$ be the closest boundary points to $\tilde{x}_i$ and $\tilde{x}_j$ on the partial shape $\tilde{\mathcal{Y}}$,

$$\tilde{x}_{i_{\tilde{\mathcal{B}}}} = \operatorname*{argmin}_{\tilde{x}_b \in \tilde{\mathcal{B}}} (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,b} \quad \text{and} \quad \tilde{x}_{j_{\tilde{\mathcal{B}}}} = \operatorname*{argmin}_{\tilde{x}_b \in \tilde{\mathcal{B}}} (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{j,b}. \tag{35}$$

In [14, 79], a first criterion $\mathcal{C}_{\mathcal{T}} : \tilde{\mathcal{Y}} \times \tilde{\mathcal{Y}} \to \{0, 1\}$ was proposed

$$\mathcal{C}_{\mathcal{T}}(\tilde{x}_i, \tilde{x}_j) = \mathbb{1}_{(\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,j} \leq (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,i_{\tilde{\mathcal{B}}}} + (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{j,j_{\tilde{\mathcal{B}}}}}, \tag{36}$$

where $\mathbb{1}$ is the indicator function. The idea behind this criterion is that if geodesic paths on the full manifold $\tilde{\mathcal{X}}$ between points $\tilde{x}_i$ and $\tilde{x}_j$ should pass through missing parts in $\tilde{\mathcal{Y}}$, then their length is at least the sum of the distances to the boundary $\tilde{\mathcal{B}}$. However, this intrinsic criterion is particularly conservative as it discards the length of this trajectory between boundary points, since information on the manifold is lost inside missing parts. Recently, [12] lifted extrinsic information to provide a worst case bound on the length of paths between boundary points. If the Riemannian metric on the manifold is the standard one given by the identity matrix, then trajectories between boundary points on the manifold are at least longer than the length of the straight segment in the original Euclidean embedding space $\mathbb{R}^n$

$$(\boldsymbol{D}_{\tilde{\mathcal{X}}})_{b_1,b_2} \geq d_E(\tilde{x}_{b_1}, \tilde{x}_{b_2}) \tag{37}$$

for any boundary points $\tilde{x}_{b_1}$ and $\tilde{x}_{b_2}$. From this simple observation, [12] generalised the $\mathcal{C}_{\mathcal{T}}$ criterion to the *wormhole criterion* $\mathcal{C}_{\mathcal{W}} : \tilde{\mathcal{Y}} \times \tilde{\mathcal{Y}} \to \{0, 1\}$ defined as

$$\mathcal{C}_{\mathcal{W}}(\tilde{x}_i, \tilde{x}_j) = \mathbb{1}_{(\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,j} \leq \boldsymbol{K}^E_{i,j}}, \tag{38}$$

where the threshold matrix $\boldsymbol{K}^E_{i,j}$ is computed as

$$\boldsymbol{K}^E_{i,j} = \min_{\tilde{x}_{b_1}, \tilde{x}_{b_2} \in B} (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,b_1} + (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{j,b_2} + d_E(\tilde{x}_{b_1}, \tilde{x}_{b_2}). \tag{39}$$

For more general Riemannian metrics on the manifold, [12] showed how to generalise the wormhole criterion. The idea is to provide a worst case bound on the distance of each infinitesimally small Euclidean arclength step along the straight Euclidean segment between boundary points. Denote $\lambda_{\tilde{M}} > 0$ to be the minimum eigenvalue of the Riemannian metric $\tilde{M}$ over the full manifold $\tilde{\mathcal{X}}$, and can be assumed to be given. By bounding the Riemannian length of Euclidean arclength steps along curves, Eq. (37) becomes

$$(\boldsymbol{D}_{\tilde{\mathcal{X}}})_{b_1,b_2} \geq \sqrt{\lambda_{\tilde{M}}} d_E(\tilde{x}_{b_1}, \tilde{x}_{b_2}) \tag{40}$$

for any boundary points $\tilde{x}_{b_1}, \tilde{x}_{b_2} \in \tilde{\mathcal{B}}$. The wormhole criterion then becomes

$$\mathcal{C}_{\mathcal{W}}(\tilde{x}_i, \tilde{x}_j) = \mathbb{1}_{(\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,j} \leq \boldsymbol{K}^R_{i,j}} \tag{41}$$

where the generalised Riemannian threshold matrix $\boldsymbol{K}^R$ is now

$$\boldsymbol{K}^R_{i,j} = \min_{\tilde{x}_{b_1}, \tilde{x}_{b_2} \in B} (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,b_1} + (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{j,b_2} + \sqrt{\lambda_{\tilde{M}}} d_E(\tilde{x}_{b_1}, \tilde{x}_{b_2}). \tag{42}$$

The criteria $\mathcal{C}_{\mathcal{T}}(x_i, x_j)$ and $\mathcal{C}_{\mathcal{W}}(x_i, x_j)$ are chosen to be the weights $w_{i,j}$ for the TCIE [79] and WHCIE [12] methods respectively. In particular, WHCIE demonstrates impressive robustness and forms the current state-of-the-art in finding consistent pairs on Riemannian manifolds.

The core idea behind the wormhole criterion is in Eqs. (37) and (40), that find how to lower bound the manifold's metric length of Euclidean arclength infinitesimal steps. We propose to take this idea and apply it to Finsler manifolds.

Finsler wormhole criterion. Assume now that the data manifold $\tilde{\mathcal{X}}$ is equipped with a Finsler metric $\tilde{F}$ and that there exists $C_{\tilde{F}} > 0$ such that the Finsler length of infinitesimal Euclidean arclength steps $d\tilde{s}$ is bounded by

$\tilde{F}_{\tilde{x}}(d\tilde{s}) \geq C_{\tilde{F}}\|d\tilde{s}\|_2$. Then the Finsler length of curves between boundary points can be lower bounded using the Euclidean embedding distance.

Proposition 3. *The Finsler distance on the Finsler manifold $\tilde{\mathcal{X}}$ between any points x_i and x_j is lower bounded by*

$$(\boldsymbol{D}_{\tilde{\mathcal{X}}})_{i,j} \geq C_{\tilde{F}} d_E(\tilde{x}_i, \tilde{x}_j)$$

Proof. The proof is an immediate generalisation of the arguments in the Riemannian case. By integrating the lower bound on Euclidean arclength steps $\tilde{F}_{\tilde{x}}(d\tilde{s}) \geq C_{\tilde{F}}\|d\tilde{s}\|_2$, and since the euclidean length of any curve between x_i and x_j is at least that of the Euclidean straight segment between them, we get the desired lower bound. □

Denote $\boldsymbol{K}^F$ the generalised Finsler threshold matrix

$$\boldsymbol{K}^F_{i,j} = \min_{\tilde{x}_{b_1}, \tilde{x}_{b_2} \in B} (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,b_1} + (\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{b_2,j} + C_{\tilde{F}} d_E(\tilde{x}_{b_1}, \tilde{x}_{b_2}). \tag{43}$$

We can then define the Finsler wormhole criterion $\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}$.

Definition 4 (Finsler wormhole criterion)**.** *The Finsler wormhole criterion $\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}$ is defined as*

$$\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_i, \tilde{x}_j) = \mathbb{1}_{(\boldsymbol{D}_{\tilde{\mathcal{Y}}})_{i,j} \leq \boldsymbol{K}^F_{i,j}}.$$

By construction, the Finsler wormhole criterion only finds consistent pairs.

Theorem 4 ($\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}$ guarantees consistent pairs)**.** *The Finsler wormhole criterion guarantees found pairs to be consistent.*

Proof. The proof follows the exact same arguments as in the Riemannian case, where now Eqs. (37) and (40) are replaced with Proposition 3. □

We thus propose the weight scheme $w_{i,j} = \mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_i, \tilde{x}_j)$ for Finsler MDS to provide robust embeddings to missing components. For optimisation algorithms requiring a symmetric weight scheme, such as our Finsler Smacof algorithm, we symmetrise it by taking the intersection $w_{i,j} = \sqrt{\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_i, \tilde{x}_j)\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_j, \tilde{x}_i)}$. Note that the square root is superfluous for binary criteria, but is not so when considering soft masks. In [12], the criterion is sometimes softened by considering the ratio between the computed shortest path lengths and the criterion matrix, and cutting it off to 1. This allows to take into account almost consistent pairs where there is only a small perturbation of the true geodesic distance, providing a reasonable compromise between accuracy and amount of data to rely on. We can soften our criterion in the same fashion by taking: $\min\left\{\frac{\boldsymbol{K}^F}{\boldsymbol{D}_{\tilde{\mathcal{Y}}}}, 1\right\}$.

We now show in a useful example how to derive the Finsler constant $C_{\tilde{F}}$ when the Finsler metric is a Randers metric with isotropic uniform Riemannian component $\tilde{F}_{\tilde{x}}(u) = \|u\|_2 + \tilde{\omega}(\tilde{x})^\top u$. Taking $u = d\tilde{s}$ to be an infinitesimal Euclidean arclength tangent vector, its Finsler length becomes minimal when $d\tilde{s}$ is oppositely aligned with the Randers drift component $\tilde{\omega}(\tilde{x})$. This leads to $\tilde{F}_{\tilde{x}}(d\tilde{s}) \geq (1 - \|\tilde{\omega}(\tilde{x})\|_2)\|d\tilde{s}\|_2$. Assuming the knowledge of $\tilde{\alpha}_{\max} = \max_{\tilde{x}} \|\tilde{\omega}(\tilde{x})\|_2 < 1$, for instance if we are provided with the maximum possible norm of the current on the manifold, we get $\tilde{F}_{\tilde{x}}(d\tilde{s}) \geq (1 - \tilde{\alpha}_{\max})\|d\tilde{s}\|_2$, meaning that $C_{\tilde{F}} = 1 - \tilde{\alpha}_{\max}$.

D. Implementation Details and Additional Experiments

D.1. Data Visualisation Experiments

We describe the implementation details in Appendix D.1.1 of experiments in Sec. 7.1 and present additional visualisation results in Sec. 7.1. These simple experiments do not require any advanced hardware, e.g. a commercial CPU suffices.

D.1.1 Implementation Considerations

In the visualisation experiments, we embed data with Finsler MDS into the canonical Randers space $\mathcal{X} = \mathbb{R}^m$, with $m \in \{2, 3\}$. The canonical Randers metric is chosen to have the fixed asymmetry level $\alpha = 0.5$. All Finsler MDS embeddings for visualisation are computed with the Finsler SMACOF algorithm. Unless specified otherwise, they use uniform weights $w_{i,j}$. Recall that the traditional SMACOF algorithm is well-known to be sensitive to initialisation. To avoid getting stuck in bad local minima, it is considered standard practice to initialise it with the Isomap [88, 96] embedding, even if the weights $w_{i,j}$ are not uniform. Following this idea, we initialise the SMACOF algorithm with the Isomap embedding to $\mathbb{R}^m$ applied to the symmetrised dissimilarity matrix $\boldsymbol{D}^S = \frac{\boldsymbol{D}+\boldsymbol{D}^\top}{2}$.

In practice, we found that pseudo-inverting the K matrix for the Finsler SMACOF update (see Proposition 2) was slow and unstable when there are many data points. To overcome this issue, we first multiplied Eq. (29) by $V^\top$ leading to a more stable update rule requiring the pseudo-inversion of a symmetric matrix

$$\mathrm{vec}(\boldsymbol{X}^{k+1}) = (K')^\dagger \,\mathrm{vec}(B'(\boldsymbol{X}^k)\boldsymbol{X}^k - C'), \tag{44}$$

where the matrices K', $B'(\boldsymbol{X}^k)$, and C' are the modified matrices $K' = (I_m + \omega\omega^\top) \otimes (V^\top V)$, $B'(\boldsymbol{X}^k) = V^\top B(\boldsymbol{X}^k)$, and $C' = V^\top C$. In addition, we resorted to the Generalized Minimal Residual method (GMRES) [83], which is a fast alternative solver of linear systems, bypassing the need to compute the Moore-Penrose pseudo-inverse of the large matrix $(K')^\dagger$ when the number of points N is large. We share the seeded code to reproduce our data and results.

Asymmetric Manifold Flattening. In this experiment from

the main paper and the additional one in the supplementary, we sample $N = 3000$ i.i.d. random vertices from the Swiss roll. The unit Euclidean vector $\tilde{\omega}$, giving the direction of the Randers metric equipping the Swiss roll, is chosen to be intrinsically uniform along the length of the Swiss roll. Note that although they are intrinsically uniform in the tangent planes $\mathcal{T}_{\tilde{x}}\tilde{\mathcal{X}} = \mathbb{R}^2$, they are not uniform extrinsically when rotating these planes to be tangent to the original embedding of the Swiss roll, as shown for instance in Fig. 2. Denote $\hat{\omega}(\tilde{x}) \in \mathbb{R}^3$ the extrinsic embedding of $\tilde{\omega}$ in the original embedding space $\mathbb{R}^3$ of the Swiss roll manifold $\tilde{\mathcal{X}}$. To compute the asymmetric geodesic distances, we compute the symmetric k-Nearest Neighbour (kNN) graph, with $k = 10$, based on the Euclidean distances in $\mathbb{R}^3$. Once the logical graph is computed, we compute the distances on these edges using a first order approximation. If points $\tilde{x}_i$ and $\tilde{x}_j$ are neighbours, we approximate $d_{\tilde{F}^{\tilde{\alpha}}}(x_i, x_j) \approx \|x_j - x_i\|_2 + \tilde{\alpha}\hat{\omega}(x_i)^\top(x_j - x_i)$, and assign this distance to the directed edge from node i to node j, and vice versa for the directed edge from node j to node i. This procedure, which generalises the standard Isomap [88, 96] approach, constructs an asymmetric weighted kNN directed graph. We can now apply Dijkstra's algorithm [34] to compute the approximate geodesic distances between all pairs of points. The results form the dissimilarity matrix $\boldsymbol{D}$, which is the input for the embedding algorithm. The result in Fig. 2 corresponds to $\tilde{\alpha} = 0.3$.

Robustness to Holes. In this experiment, 2000 i.i.d. points are sampled on the full Swiss roll, but points falling within a rectangular region encoding the hole are removed. The Randers metric equipping the manifold $\tilde{\mathcal{X}}$ is the same as in the *Asymmetric Manifold Flatenning* experiments on the Swiss roll with $\tilde{\alpha} = 0.5$. We apply the same algorithm to compute the Randers distance between points, with $k = 15$ in the kNN graph construction. To create an embedding that is robust to the missing part, the weights are given by the binary Finsler wormhole criterion, logically symmetrised: $w_{i,j} = \sqrt{\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_i, \tilde{x}_j)\mathcal{C}_{\mathcal{W}_{\mathcal{F}}}(\tilde{x}_j, \tilde{x}_i)}$ (see Appendix C). To compute the wormhole criterion, we assume that the metric is behaved in the missing parts similarly to the rest of the data, leading to a choice of $\tilde{\alpha}_{\max} = \tilde{\alpha}$ to compute the constant $C_{\tilde{F}}$ in Proposition 3.

Unflattening Current Maps. In this experiment from the main paper and the additional one in the supplementary (corresponding to Fig. 8), we sample N i.i.d. random points in a rectangular region $\tilde{\Omega}$ of the plane $\mathbb{R}^2$. Given an unconstrained current $\breve{v}(\tilde{x}_i) \in \mathbb{R}^2$ at any point $\tilde{x}_i \in \tilde{\Omega}$, the current is then chosen to be $\tilde{v}(\tilde{x}_i) = \tilde{\alpha}\frac{\breve{v}(\tilde{x}_i)}{\max_j \|\breve{v}(\tilde{x}_j)\|_2}$. The Randers metric at the sampled point $\tilde{x}_i$ is then chosen to be $\tilde{F}(\tilde{x}_i) = \|u\|_2 - \tilde{v}(\tilde{x}_i)^\top u$ (see Appendix B). We then apply the same algorithm as in the *Asymmetric Manifold Flattening* experiment on the Swiss roll to compute Randers geodesic distances, using a kNN graph with $k = 10$ neighbours. Note that since the original space and its tangent space coincide $\tilde{\mathcal{X}} = \mathcal{T}_{\tilde{x}}\tilde{\mathcal{X}}$ at all points $\tilde{x}$, the extrinsic embedding of the drift component $\tilde{\omega}(\tilde{x}) = -\tilde{v}(\tilde{x})$ is the same as its intrinsic version $\hat{\omega}(\tilde{x}) = \tilde{\omega}(\tilde{x})$.

For the experiment in the main paper, we sample $N = 2000$ points from the domain $\tilde{\Omega} = [0, 10]^2$. At any point $\tilde{x}_i = (\tilde{x}_i^{(1)}, \tilde{x}_i^{(2)})^\top \in \Omega$, we define the unconstrained current field $\breve{v}(\tilde{x}_i) = \big(\sin(\nu\tilde{x}_i^{(1)}) + \cos(\nu\tilde{x}_i^{(2)}), \cos(\nu\tilde{x}_i^{(1)}) - \sin(\nu\tilde{x}_i^{(2)})\big)^\top$, with $\nu = 2$. The current field is constructed with $\tilde{\alpha} = 0.5$. For the river experiment in the supplementary, we sample $N = 1000$ points from the domain $\tilde{\Omega} = [0, 10] \times [0, 1]$. The unconstrained current at point $\tilde{x}_i$ is given by $\breve{v}(\tilde{x}_i) = (1 - |2\tilde{x}_i^{(2)} - 1|, 0)^\top$. The current is then constructed with $\tilde{\alpha} = 0.2$.

Revealing Graph Hierarchies In this experiment, we construct a full and complete binary tree of depth $h = 7$, having thus $N = 2^{h+1} - 1 = 255$ nodes. The edge from a parent to it child is given the weight of 0.5, whereas the edge from a child to its parent has a weight of 1.5. Additionally, we add undirected edges between all nodes at the same height, with a weight of 0.1. Given two nodes connected by an edge, their distances is given by the edge weight. Asymmetric geodesic distances between any two nodes are then computed using Dijkstra's algorithm, which constitute the dissimilarity matrix $\boldsymbol{D}$.

D.1.2 Additional Results

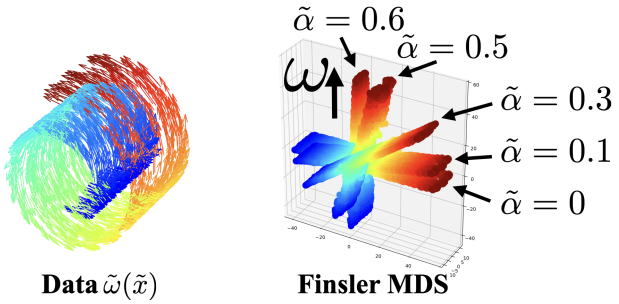

Figure 6. Flattening the Swiss roll equipped with a Randers metric $\tilde{F}^{\tilde{\alpha}}$ with various asymmetry levels $\tilde{\alpha}$ given by $\tilde{F}^{\tilde{\alpha}}_{\tilde{x}}(u) = \|u\|_2 + \tilde{\alpha}\tilde{\omega}^\top u$ and $\|\tilde{\omega}\|_2 = 1$. We superimpose on the right the resulting Finsler MDS embeddings in the 3D canonical Randers space with fixed asymmetry $\alpha = \|\omega\|_2$.

Asymmetric Manifold Flattening. By changing the value of $\tilde{\alpha}$, we vary the amount of asymmetry on the Swiss roll. However, in this experiment, we do not change the asymmetry measure of the canonical Randers space of the embed-

dings: α is fixed. We superimpose in Fig. 6 the resulting Finsler MDS embeddings for various asymmetry levels of the data $\tilde{\alpha} \in \{0, 0.1, 0.3, 0.5, 0.6\}$. In all cases, the embedded Swiss roll resembles a flat 2D band in 3D, albeit with varying vertical orientation. As expected, the higher the value of $\tilde{\alpha}$, the more the embedded Swiss roll becomes vertical along the axis $\vec{z}$ of asymmetry. When the data is (close to) symmetric, i.e. $\tilde{\alpha}$ is (close to) 0, the embedding is (close to) aligned with the xy plane. Finsler MDS thus not only provides embeddings preserving the manifold structure, but its verticality also provides an intuitive visual cue encoding the asymmetry of the data.

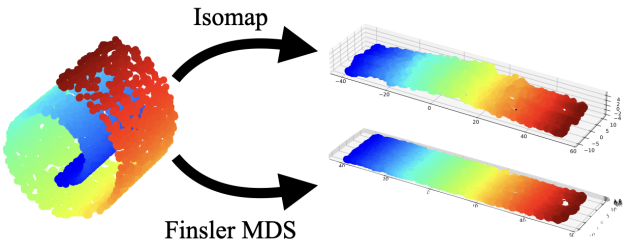

Figure 7. Flattening the original symmetric Swiss roll. The embedding is either into the Euclidean space $\mathbb{R}^3$ with Isomap or into the canonical Randers space $\mathbb{R}^3$ with our Finsler MDS. Finsler MDS provides robust embeddings that generalises traditional symmetric embedding methods on symmetric data while revealing the additional information that the data is symmetric.

Symmetric Manifold Flattening. We focus on the embedding of the vanilla symmetric Swiss roll, i.e. $\tilde{\alpha} = 0$, to $\mathbb{R}^3$ using either the traditional MDS, with Isomap, or Finsler MDS, with our Finsler SMACOF algorithm. These results are presented without other values of $\tilde{\alpha}$ in Fig. 7. For the Isomap embedding to $\mathbb{R}^3$, the Swiss roll is not perfectly flattened in the xy hyperplane. This incorrectly suggests that the Swiss roll is not a flat Riemannian structure, i.e. with effectively 0 Gaussian curvature. This error is due to small noise in the estimate of the distance matrix $\boldsymbol{D}$ as Dijkstra's algorithm only provides an approximation to geodesic distances as geodesic paths are constrained to live on the neighbourhood graph constructed from the data. To avoid this issue, the Swiss roll is usually embedded to $\mathbb{R}^2$, yielding the desired 2D flattened Swiss roll rectangle. In contrast, our Finsler MDS embedding to the canonical Randers space $\mathbb{R}^3$ is flattened to the xy plane and is similar to the ideal 2D Isomap embedding, as predicted by Theorem 2. Additionally, our embedding also provides the information that the original Swiss roll is a symmetric structure as all embedded points have the same height. As such, Finsler MDS not only robustly provides superior embeddings for symmetric data that generalise the traditional methods, it also yields additional information compared to them.

Dataset	Cora	Citeseer	Gr-QC	Chameleon	Squirrel	Arxiv-Year
$\lvert\mathcal{V}\rvert$	2,708	3,327	5,242	2,277	5,201	169,343
$\lvert\mathcal{E}\rvert$	5,429	4,552	14,496	31,371	198,353	1,166,243

Table 3. Summary of dataset statistics for link prediction tasks. We note $\lvert\mathcal{V}\rvert$ and $\lvert\mathcal{E}\rvert$ the numbers of nodes and edges, respectively.

Unflattening Current Maps. In addition to the unflattening of the current map in Fig. 4, we also embed using Finsler MDS the classic river manifold in Fig. 8. As explained in Appendix B, from a timewise perspective, we equip the river with a Randers field with $\tilde{\omega} = -\tilde{v}$, where $\tilde{v}$ is the current field. The Finsler MDS embedding of the river leads to an intuitive embedding clearly revealing the existence of asymmetry between points upstream and downstream. This contrasts with the original current map, even when enriched with arrows to artificially break the Euclidean symmetry, as they can be difficult to discern when numerous or with low magnitude.

D.2. Digraph Embedding and Link Prediction Experiments Implementation Details

We describe the setups and additional details of our experiments in Sec. 7.2. The experiments are performed on a NVIDIA DGX A100 GPU.

Datasets. For both the digraph embedding and link prediction tasks, we evaluate on six publicly available directed graph datasets: the citation networks Cora [89] and Citeseer [108], the arXiv collaboration network in general relativity and quantum cosmology (Gr-QC) [57], and three heterophilic graphs: Chameleon, Squirrel [82], and Arxiv-Year [47]. The detailed statistics of these benchmarks are summarized in Tab. 3.

Digraph Embedding Baseline. To utilize the directional property for learning efficient representations, we propose computing embeddings in Finsler space instead of Euclidean space. We note that while [60] explores a Finsler-Riemannian framework for graph embedding, their approach is not applicable to directed graphs. Therefore, we do not include comparisons between their framework and our Finsler representation for digraph embedding.

Link Prediction Baseline. For link prediction tasks, we compare our method with NERD [51], DiGCN [97], MagNet [112], DiGAE [53], ODIN [109], and DUPLEX [50]. NERD is a shallow method that uses node semantics based on a random walk strategy to sample node neighbourhoods from a directed graph. DiGCN introduces a spectral Graph Neural Network (GNN) model built on digraph convolution, utilizing Personalized PageRank as its foundation. MagNet

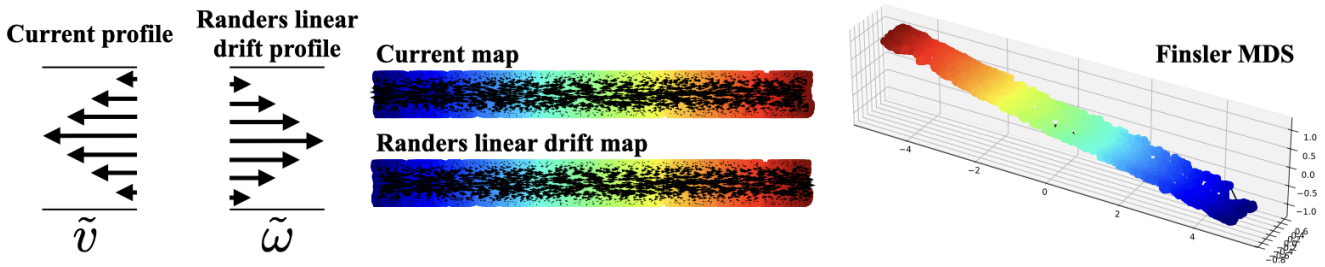

Figure 8. Embedding of the river map with a fixed current profile $\tilde{v}$. The associated Randers drift component $\tilde{\omega}$ is in the opposite direction. Plotting arrows on the map might lead to cluttered visualisations that make the asymmetry difficult to read even in this simple toy example. In contrast the Finsler MDS embedding clearly reveals the asymmetric nature of the river while preserving its spatial straight property.

proposes the magnetic Laplacian to define graph convolutions. Both DiGCN and MagNet are spectral-based methods. DiGAE is a digraph autoencoder model that employs a directed GCN as its encoder. ODIN is a recent shallow method that learns multiple embeddings per node to model directed edge formation factors while disentangling interest factors from in-degree and out-degree biases. DUPLEX employs dual graph attention network encoders that operate on a Hermitian adjacency matrix.

Digraph Embedding Setup. To assess the capacity of the Euclidean and Finsler representations, we embed the full data with a Multi-Layer Perceptron (MLP) and compute the pairwise distances in the embedding space. We implement our proposed method using PyTorch [72]. We train the Euclidean embedding with Euclidean stochastic gradient descent. Riemannian stochastic gradient descent [8] generalises classical stochastic gradient descent to optimization on Riemannian manifolds by replacing Euclidean updates with retractions that map stochastic gradients from tangent spaces back onto the manifold. For the Finsler embedding, we train the embedding by adapting the Riemannian stochastic gradient descent from the Riemannian metric with the canonical Randers metric defined in Sec. 5. We consider the embedding space $\mathbb{R}^m$ of various dimensions $m \in \{2, 5, 10, 50\}$. For each dimension m, we use the Optuna [2] hyperparameter optimiser to choose the learning rate, the number of hidden layers, the hidden dimension, and the dropout probability within candidate sets. These candidate sets are $\{5e^{-1}, 3e^{-1}, 2e^{-1}, 1e^{-1}, 5e^{-2}, 1e^{-2}, 5e^{-3}, 1e^{-3}\}$ for the learning rate, $\{1, 2, 3, \cdots, 10\}$ for the number of hidden layers, $\{64, 128, 256, 512\}$ for the hidden dimension, and $\{0, 0.1, 0.2, \cdots, 0.9\}$ for the dropout probability.

Link Prediction Setup. We evaluate on two types of link prediction tasks. The first task involves predicting the direction of edges between vertex pairs u and v, where it is known that there exists an edge between the two vertices but not its direction: $(u, v) \in \mathcal{E}$ or $(v, u) \in \mathcal{E}$. The second task focuses on existence prediction, where the goal is to determine whether $(u, v) \in \mathcal{E}$, considering vertex pairs (u, v). For link prediction tasks, we divide the graph datasets, by partitioning the edges randomly while preserving the graph connectivity, into 80% (of edges) for training, 15% (of edges) for testing, and 5% (of edges) for validation, following the work [112]. Performance is assessed by measuring the Area Under the ROC Curve (AUC). The link prediction quality is computed by the average performance and standard deviation over 10 random splits. We utilize the source code released by the authors for the baseline algorithms and optimize their hyperparameters using Optuna [2].