FAST NON-METRIC MDS FOR NOISY HIGH-DIMENSIONAL DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Multidimensional scaling (MDS) constructs low-dimensional embeddings that approximate pairwise distances in a high-dimensional dataset. MDS exists in several flavors, with metric MDS approximating distances directly and non-metric MDS additionally optimizing for an arbitrary monotonic transformation of the high-dimensional distances. Empirically, metric MDS is known to suffer from the curse of dimensionality. Here we extend prior theoretical results for metric MDS on fully uninformative data to a more general case of finite, high-dimensional data with concentrated distances. We show that the metric MDS embeddings of such datasets stay distributionally close to the data-independent limit case. We then show that non-metric MDS is more robust to high-dimensional noise and can yield informative embeddings of such data. While existing implementations of non-metric MDS are slow, we develop a fast implementation using stochastic gradient descent in PyTorch. Using our implementation, we conduct an empirical study using multiple simulated and real-world datasets, including population-genomic and scRNA-seq data, and show that non-metric MDS can strongly outperform other methods in terms of global structure preservation.

1 INTRODUCTION

Multidimensional scaling (MDS) (Borg & Groenen, 2005) is one of the classic dimensionality-reduction and embedding methods developed in the 1960s (Shepard, 1962a;b; Kruskal, 1964a;b). Its goal is to find low-dimensional embedding vectors $\mathbf{z}_i \in \mathbb{R}^p$ such that pairwise distances $\|\mathbf{z}_i - \mathbf{z}_j\|$ between them approximate some given pairwise distances D_{ij}. MDS is well-known in the field of dimensionality reduction as the algorithm aiming to preserve global structure of the data (Lee & Verleysen, 2007), as opposed to various neighbor-embedding algorithms such as t-SNE (van der Maaten & Hinton, 2008) and UMAP (McInnes et al., 2018) that aim to preserve local structure.

Empirically, MDS often struggles to produce a meaningful embedding of high-dimensional data (de Bodt et al., 2025). Here we formalize this observation and study the behavior of MDS under distance concentration. We show that the so-called *metric MDS*, where $\|\mathbf{z}_i - \mathbf{z}_j\|$ should directly approximate given D_{ij}, suffers from the curse of dimensionality and remains close to a data-independent embedding (Mattner, 1992; Buja et al., 1994). In contrast, we demonstrate that *non-metric MDS*, a more flexible but less widely used flavor of MDS, where $\|\mathbf{z}_i - \mathbf{z}_j\|$ should approximate D_{ij} only up to an arbitrary monotonic transformation (i.e. approximating the order of distances and not the distances themselves), is more robust and can better tolerate high-dimensional noise, while still producing informative embeddings.

Most existing implementations of non-metric MDS, such as the one in scikit-learn (Pedregosa et al., 2011) in Python or the `smacof` package in R (de Leeuw & Mair, 2009; Mair et al., 2022), have quadratic memory and cubic runtime complexity, precluding its use for datasets with sample sizes $n \gg 10\,000$. Indeed, simply storing the $n \times n$ matrix of pairwise distances $\mathbf{D}$ in memory can already be prohibitive for large n. While several stochastic versions of metric MDS have been developed in various contexts (Agrafiotis, 2003; Zheng et al., 2018; Lambert et al., 2022; Fang et al., 2025; Hangan et al., 2026), fast implementations of non-metric MDS do not exist. Here we fill this gap.

In summary, our contributions are:

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.

1. we establish a perturbation bound for the limiting solution of metric MDS under distance concentration in the presence of high-dimensional noise;
2. we show that non-metric MDS is more robust and can tolerate higher-dimensional noise;
3. we develop a stochastic implementation of non-metric MDS in PyTorch;
4. we show that non-metric MDS can strongly outperform other methods in terms of global embedding quality.

We argue that our results are relevant for modern embedding methods, as many of them include MDS-based loss terms to ensure global structure preservation (see Discussion).

2 Related Work

On the theoretical side, so-called classical MDS (see below) has recently received attention from mathematicians and was shown to exhibit surprising properties in the continuous $n \to \infty$ limit (Kroshnin et al., 2022; Lim & Mémoli, 2024). Buja et al. (1994) characterized the solution of metric MDS in the continuous limit for fully uninformative inputs.

On the practical side, multiple papers independently re-introduced stochastic *metric* MDS in various contexts. Agrafiotis (2003) developed a stochastic metric MDS with batch size $b = 2$, calling it stochastic proximity embedding (SPE). This method was developed in the context of computational chemistry, to visualize large collections of chemical compounds (Agrafiotis et al., 2010).

Zheng et al. (2018) developed stochastic implementations of weighted MDS in the context of graph drawing. In that setting, the input matrix of distances is the matrix of shortest-path graph distances between pairs of graph nodes. For optimization, the authors sampled pairs of points, essentially using mini-batches of size $b = 2$. Hangan et al. (2026) adopted this approach for dimensionality reduction and Fang et al. (2025) used a similar approach with $b = n/10$ for their FastMDS.

Lambert et al. (2022) developed SQuadMDS (stochastic quartet MDS) that optimizes metric MDS using mini-batches of size $b = 4$ (quartets). The authors developed a C++ implementation and used it to obtain metric MDS embeddings of MNIST and the Tasic datasets, also analyzed here. Furthermore, the authors suggested a hybrid between SQuadMDS and t-SNE.

At the same time, we are not aware of any scalable implementations of *non-metric* MDS. All examples in Borg & Groenen (2005) and Mair et al. (2022) have $n < 100$.

3 Background

Notation Let the high-dimensional dataset consist of n vectors $\mathbf{x}_i \in \mathbb{R}^d$, forming an $n \times d$ data matrix $\mathbf{X}$. Let $\mathbf{D}$ be an $n \times n$ pairwise distance matrix with D_{ij} being some distance (e.g. Euclidean, cosine, etc.) between $\mathbf{x}_i$ and $\mathbf{x}_j$. Alternatively, we may be given (e.g. experimentally obtain) some distance matrix $\mathbf{D}$ directly, without access to any underlying $\mathbf{X}$, and without knowing the value of d.

The goal of any MDS algorithm is to construct low-dimensional embedding vectors $\mathbf{z}_i \in \mathbb{R}^p$ such that their pairwise distances approximate pairwise distances D_{ij}. Typically $p = 2$ for visualization purposes. We refer the reader to Borg & Groenen (2005) for a detailed systematic treatment of MDS.

Classical MDS In classical MDS, also known as Torgerson's scaling (Torgerson, 1952) and as principal coordinates analysis (PCoA) (Gower, 1966), the assumption is that $\mathbf{D}$ is a matrix of Euclidean distances. If so, it can be converted into the Gram matrix of centered vectors $G_{ij} = (\mathbf{x}_i - \bar{\mathbf{x}})^\top (\mathbf{x}_j - \bar{\mathbf{x}})$ by double-centering the matrix of squared distances:

$$\mathbf{G} = -(\mathbf{I} - \mathbf{1}_n/n)(\mathbf{D} \odot \mathbf{D}/2)(\mathbf{I} - \mathbf{1}_n/n), \tag{1}$$

where $\odot$ denotes elementwise multiplication, and $\mathbf{1}_n$ denotes the $n \times n$ matrix of all ones. If $\mathbf{G} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ is the eigendecomposition of $\mathbf{G}$, then $\mathbf{Z} = \mathbf{U}\mathbf{\Lambda}^{1/2}$ gives the principal components of $\mathbf{X}$ and the classical MDS of $\mathbf{D}$.

Note that if $\mathbf{D}$ is not a matrix of Euclidean distances, then classical MDS is not equivalent to principal component analysis (PCA) of $\mathbf{X}$. If $\mathbf{\Lambda}$ contains negative eigenvalues, then $\mathbf{D}$ cannot be obtained as

Euclidean distances between vectors of any dimensionality (Schoenberg, 1935). Recent work studied properties and extensions of classical MDS in this regime (Sonthalia et al., 2021; Deng et al., 2024).

Metric MDS The loss function of metric MDS is the squared error between D_{ij} and $\|\mathbf{z}_i - \mathbf{z}_j\|$:

$$\mathcal{L}_{\text{mMDS}} = \sum_{i<j} \big(\|\mathbf{z}_i - \mathbf{z}_j\| - D_{ij}\big)^2. \tag{2}$$

It is called (raw) stress. It is often more convenient to work with the raw stress divided by $n(n-1)$, which we will call the average stress. The common approach for minimizing the stress loss function relies on the iterative SMACOF algorithm (scaling by majorizing a complicated function; de Leeuw, 1977). This algorithm is used in most common MDS implementations, e.g. in R (de Leeuw & Mair, 2009; Mair et al., 2022) and in the scikit-learn library (Pedregosa et al., 2011) in Python. SMACOF converges to a local minimum of $\mathcal{L}_{\text{mMDS}}$, but note that $\mathcal{L}_{\text{mMDS}}$ is not convex and has many local minima, so the final embedding depends on the initialization.

Non-metric MDS Non-metric MDS (Shepard, 1962a;b; Kruskal, 1964a;b) additionally allows an arbitrary monotonic transformation $f(\cdot)$ of the input distances:

$$\mathcal{L}_{\text{nMDS}} = \sum_{i<j} \big(\|\mathbf{z}_i - \mathbf{z}_j\| - f(D_{ij})\big)^2 \quad \text{s.t.} \quad \frac{1}{n(n-1)} \sum_{i<j} f(D_{ij})^2 = 1. \tag{3}$$

Here, to avoid a trivial solution with $f(\cdot) = 0$, the transformed values $f(D_{ij})$ are usually constrained to have mean squared off-diagonal value equal to 1 (de Leeuw & Mair, 2009; Mair et al., 2022). Fixing $f(\cdot)$ to be proportional to the identity recovers metric MDS up to the scaling factor. The standard approach for minimizing this loss function is alternating minimization: after each step of the SMACOF algorithm, an optimal $f(\cdot)$ is fit by isotonic (monotonic) regression of the current $\|\mathbf{z}_i - \mathbf{z}_j\|$ values on D_{ij} values. The resulting $f(D_{ij})$ values are then scaled to satisfy the constraint.

Stress-1 It is often more convenient to use some normalized version of stress, for example so-called stress-1 (Kruskal, 1964a;b):

$$\text{Stress-1} = \sqrt{\frac{\sum_{i<j} \big(\|\mathbf{z}_i - \mathbf{z}_j\| - f(D_{ij})\big)^2}{\sum_{i<j} \big(\|\mathbf{z}_i - \mathbf{z}_j\|\big)^2}}. \tag{4}$$

It can be shown that any minimum of raw (or average) stress corresponds to a minimum of stress-1, if $\mathbf{Z}$ is appropriately rescaled. The optimal scaling factor can be computed analytically, and the stress-1 of the rescaled embedding ends up being equal to $\sqrt{s^2/(1+s^2)}$ where s is the stress-1 value of $\mathbf{Z}$ obtained by minimizing Eq. 2 or Eq. 3 (Borg & Groenen, 2005, Chapter 11.1). In the following, we always report this rescaled value (which one would also get by directly minimizing stress-1). Note that scikit-learn returns stress-1 value without this adjustment. If $f(\cdot)$ is normalized as in Eq. 3, then, at the minimum, the rescaled stress-1 coincides with the square root of the average stress (ibid.).

Scikit-learn implementation and our improvements The scikit-learn library (Pedregosa et al., 2011) in Python contains implementations of metric and non-metric MDS using the SMACOF algorithm. While working on this paper, we identified a number of issues with that implementation and fixed them in a series of four PRs (PR numbers hidden to preserve anonymity), which have been merged into the version 1.8 (released in December 2025). Apart from fixing some implementation errors in non-metric MDS, we implemented the classical MDS algorithm (as `manifold.ClassicalMDS` class) and set it as default initialization for metric and non-metric MDS. All scikit-learn experiments reported below used version 1.8 and default hyperparameters (default from version 1.10), and the algorithm always reached the default convergence criterion.

4 The curse of dimensionality for metric MDS

Here we will extend and combine prior results by Mattner (1992), Buja et al. (1994), and Aggarwal et al. (2001) to show that finite, noisy, high-dimensional data yields an uninformative metric MDS embedding. The motivates the use of non-metric MDS for high-dimensional data.

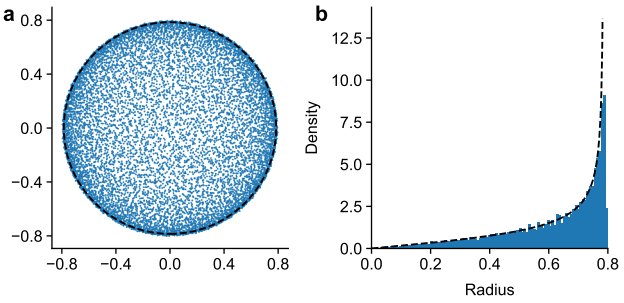

Figure 1: **(a)** Metric MDS of a $10\,000 \times 10\,000$ distance matrix with all off-diagonal elements equal to $1 + \varepsilon_{ij}$ with $\varepsilon_{ij} \sim \mathcal{N}(0, 0.01)$. The dashed circle has radius $\pi/4$. **(b)** Radial density of points in panel (a). Linear increase corresponds to the uniform 2D density. The dashed line shows theoretical prediction.

The standard result on distance concentration is that in the presence of high-dimensional noise, all pairwise distances become approximately the same (Aggarwal et al., 2001). For instance, consider an arbitrary 2D dataset with n points (e.g. sampled from a mixture of well-separated Gaussians, as we will do in Section 6). To mimic high-dimensional noise, we append d additional non-discriminative dimensions with i.i.d. values with mean 0, variance σ^2, and finite 4th moment, so that the resulting dataset has dimensionality $d + 2$. As d increases, all pairwise distances D_{ij} will concentrate around $\sqrt{2d\sigma^2}$ as all vectors become close to orthogonal and each of them has squared norm close to $d\sigma^2$ (Hall et al., 2005; Hiraoka et al., 2026).

Dividing all distances by $\sqrt{2d\sigma^2}$, we see that the $n \times n$ distance matrix approaches $D_{ij} = 1 - \delta_{ij}$ with all off-diagonal elements equal to 1. The continuous limit of metric MDS, corresponding to $n \to \infty$, on this fully uninformative input was analyzed by Mattner (1992) and Buja et al. (1994). Let $\mathcal{P}(\mathbb{R}^2)$ denote the set of probability measures on $\mathbb{R}^2$. For $\rho \in \mathcal{P}(\mathbb{R}^2)$ and $X, Y \overset{\text{i.i.d.}}{\sim} \rho$ the average stress functional is

$$S[\rho] = \mathbb{E}_\rho\big[(\|X - Y\| - 1)^2\big] = \iint \big(\|x - y\| - 1\big)^2 \rho(x)\rho(y)\, dx\, dy. \tag{5}$$

Theorem 1 (Limiting metric MDS embedding, (Mattner, 1992; Buja et al., 1994))**.** *The unique minimizer of $S[\rho]$ over $\mathcal{P}(\mathbb{R}^2)$ with $\mathbb{E}[\rho] = 0$ is the Riesz $s = 1$ equilibrium measure with density*

$$\rho^*(x) = \frac{2}{\pi^2 \sqrt{(\pi/4)^2 - \|x\|^2}} \text{ with support on } \|x\| < \frac{\pi}{4}. \tag{6}$$

It achieves a normalized stress value of $S^* := S[\rho^*] = 1 - \frac{\pi^2}{12} \approx 0.1775$.

We include a detailed proof of this theorem in Appendix C. The main observation is that the Laplacian of the functional derivative of S yields the Riesz $s = 1$ potential. This is solved by the Riesz $s = 1$ equilibrium measure (Riesz, 1938; Landkof, 1972; Chafai et al., 2022). Different from the idealized setting, real data is finite and pairwise distances are only concentrated, not exactly equal. Using the energy distance D_E^2 (Székely & Rizzo, 2013), defined for two (possibly discrete) distributions μ and ν as

$$D_E^2(\mu, \nu) = 2 \iint \|x - y\|\, d\mu(x)\, d\nu(y) - \iint \|x - x'\|\, d\mu(x)\, d\mu(x') - \iint \|y - y'\|\, d\nu(y)\, d\nu(y'), \tag{7}$$

we establish the following perturbation bound for this setting:

Proposition 2. *Let $D'_{ij} = 1 - \delta_{ij} - \varepsilon_{ij}$ be an $n \times n$, zero-diagonal, symmetric dissimilarity matrix, so that its root-mean-squared deviation from the fully uninformative D is $\Delta = \sqrt{\frac{2}{n(n-1)} \sum_{i<j} \varepsilon_{ij}^2}$. Then for any solution X' of metric MDS for dissimilarities D', we have*

$$D_E^2(X, \rho^*) \;\le\; \frac{\pi^2}{24n} \;+\; 2\sqrt{2S^*}\,\Delta \;+\; 2\,\Delta^2 \;\approx\; \frac{0.411}{n} + 0.843\,\Delta + 2\,\Delta^2. \tag{8}$$

Appendix D contains the proof, which relies on bounding the energy distance by the stress functional, taking finite samples from ρ^*, and using a reverse triangle inequality.

The marginal distribution of the norm $\|X\|$ for $X \sim \rho^*$ is a rescaled Kumaraswamy $(a = 2,\ b = 1/2)$ distribution on $[0, \pi/4)$ with density $p(r) = 4r/\big(\pi\sqrt{(\pi/4)^2 - r^2}\big)$ (Corollary 6). Our simulation

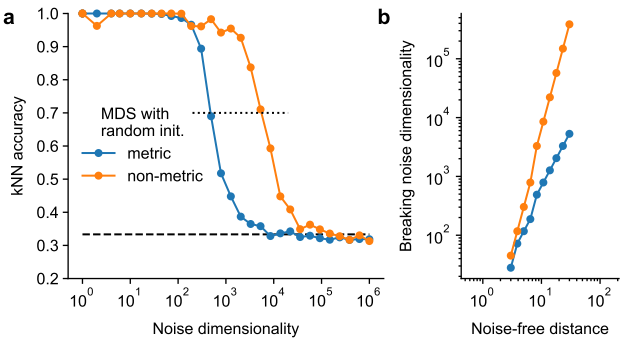

Figure 2: **(a)** Leave-one-out kNN classification accuracy ($k = 10$) in MDS embeddings of noisy three-class dataset. There are $n = 99$ points from 3 equisized classes, equidistant from each other with distance $D = 10$ in $\mathbb{R}^2$. We add $d \in [1, 10^6]$ noise dimensions with i.i.d. $\mathcal{N}(0, 1)$ values. MDS is run with random initialization. Average over 20 runs. **(b)** Minimum d that yields kNN accuracy below 0.7 for $D \in [3, 30]$.

with $n = 10\,000$ (Figure 1) nearly equidistant points confirms that the optimum of metric MDS stays close to the continuous limit.

Curse of dimensionality: Metric MDS of noisy high-dimensional data with large sample size will look like a disk with points concentrated on the boundary, with radius $\bar{D}\pi/4$, where $\bar{D}$ is the average high-dimensional pairwise distance in the data.

5 Non-metric MDS is more robust to high-dimensional noise

We experimentally measured the robustness of metric and non-metric MDS to high-dimensional noise using the following simulation. We simulated a noisy dataset by generating $n = 99$ points from 3 equisized classes, equidistant from each other with distance $D = 10$ in $\mathbb{R}^2$. We then added $d \in [1, 10^6]$ noise dimensions with i.i.d. $\mathcal{N}(0, \sigma^2)$ values with $\sigma = 1$. Then we ran both versions of MDS using random initialization (we only used random initialization in this section) and measured the kNN classification accuracy in the embedding. We found that non-metric MDS could tolerate much higher noise dimensionality d before kNN accuracy collapsed (Figure 2a).

We then asked what value of d leads to the accuracy collapse. The pairwise distances between noise vectors concentrate around $\sqrt{2d\sigma^2}$. Metric MDS considers absolute distances and gets strongly distorted when these distances become comparable to D. This happens when $\boxed{d^*_{\text{mMDS}} = \Theta(D^2/\sigma^2)}$.

For large d, the pairwise distances between noisy observations can be approximated by $\sqrt{2d\sigma^2 + D^2} \approx \sqrt{2d\sigma^2} + D^2/(2\sqrt{2d\sigma^2})$. Non-metric MDS gets strongly distorted when the last term ('signal') becomes comparable to the fluctuations of the first term ('noise'), so that the noise affects the *ranking* of distances. The distances between noise vectors follow a χ-distribution and its standard deviation asymptotically equals σ. Equating the second term above to σ yields $\boxed{d^*_{\text{nMDS}} = \Theta(D^4/\sigma^4)}$.

To verify this experimentally, we ran our simulation for various values of $D \in [3, 30]$ and found the minimal $d^*(D)$ value with kNN accuracy below 0.7 ('collapse'). In log-log coordinates, $d^*(D)$ was close to a linear function (Figure 2b) with regression slopes 2.2 for metric MDS and 4.0 for non-metric MDS, in an excellent agreement with our prediction. In a more varied setup, with varying clusters sizes, noise types, and only approximate equidistance, the regression slopes were still within $[2.08, 2.20]$ and $[4.00, 4.05]$, for metric and non-metric MDS, respectively (Appendix B).

6 Stochastic non-metric MDS

SGD implementation in PyTorch To be able to run non-metric MDS on datasets with large sample size, we implemented a simple stochastic approximation of both metric and non-metric MDS (Listing S1). We first describe the algorithmic steps used in both setups. In each training epoch, we randomly split the dataset into $\lfloor n/b \rfloor$ mini-batches of size b. If the input data are given as $\mathbf{X}$ matrix, then for each mini-batch, we compute $b \times b$ values of D_{ij} (in order to avoid storing the entire $n \times n$ matrix $\mathbf{D}$ in memory, as this can be prohibitive for large n). Then we use the raw stress (Eq. 4) as the loss, with the sum only running over the b^2 terms in the mini-batch.

The entire model is an instance of `torch.nn.Embedding`, initialized with PCA (in case $\mathbf{X}$ is used as input) or with classical MDS (in case $\mathbf{D}$ is used). We scale the initialization to have a pre-specified standard deviation σ_{init} of the first component. For optimization we use Adam with learning rate λ and cosine learning-rate decay over n_{epochs}. This is in agreement with Agrafiotis (2003) and Zheng et al. (2018) who argued that starting with a large learning rate and decaying it down to zero helps optimizing stochastic metric MDS.

For the stochastic version of non-metric MDS, the only modification to the above is that D_{ij} values are transformed with isotonic regression (we use `IsotonicRegression` from scikit-learn) and then normalized to have mean squared off-diagonal value equal to 1. As isotonic regression can be slow, we opted to fit it only on the first batch in each epoch; for subsequent batches in the same epoch, we use the fitted regression model to transform the D_{ij} values. The normalization coefficient was also computed on the first batch and used throughout the epoch. Fitting isotonic regression more often led to only very minor improvements in the final loss (Table S1).

Default hyperparameters As the default hyperparameters, we use $\lambda = 0.02$ and $\sigma_{\text{init}} = 0.5$ (which we found to work better than 1 on some of the datasets). For small datasets with $n \le 5\,000$, we use batch size $b = 100$, and for larger datasets we use $b = 1\,000$. The default number of epochs is set to $n_{\text{epochs}} = 200$ if $n \le 50\,000$ and to $n_{\text{epochs}} = 20$ otherwise. We found these parameters to work reasonably well across all considered datasets. We increased the number of epochs to $n_{\text{epochs}} = 500$ for non-metric MDS of the Tasic dataset, see below. Note that on some of our datasets a much smaller number of training epochs would suffice to reach a similar final loss.

Implementation details We observed that the optimal learning rate grew with the batch size (Figure S1) and that larger batch sizes were beneficial and allowed to achieve lower loss. However, the runtime of our implementation scales as $b^2 \cdot \lfloor n/b \rfloor \approx nb$, and in the non-metric case, the isotonic regression also becomes slower with larger b. We found $b = 1000$ to be a reasonable compromise.

All experiments were run on a laptop with Intel Core i5-8265U CPU at 1.6 GHz and 16 Gb RAM.

We ran all experiments for three random seeds and report means. The standard deviations were so small that we omitted them (rounded 0.00 for Table 1 and no more than 0.01 for Table 2).

7 Experimental results

7.1 Non-metric MDS mitigates the curse of dimensionality

We started with analyzing a synthetic noisy blob dataset, with 5 000 points sampled from a mixture of three well-separated spherical 2D Gaussians (with means sampled from $[-10, 10]$ and variances 1) with additional 1000 non-discriminative standard Gaussian dimensions (Figure 3). Classical MDS (i.e., PCA) produced an embedding with well-separated classes (Figure 3a), essentially recovering the original 2D data. At the same time, metric MDS could barely separate the classes (Figure 3b) and in fact typically failed to yield separated classes when using random initialization (by default we used classical MDS as initialization for metric MDS).

To evaluate the embeddings in a loss-independent way, we used triplet accuracy (for each sample A we randomly selected samples B and C from the upper/lower halves of pairwise distance distribution from A; we then computed the fraction of such triplets where A–C embedding distance was smaller than A–B). We also used Pearson and Spearman correlations between pairwise distances in the embedding space and in the original space (here we randomly sampled 500 points and computed all distances within this subset). These are commonly used metrics to assess global embedding quality (e.g. Becht et al., 2019; Kobak & Berens, 2019; Huang et al., 2022; de Bodt et al., 2025).

On the noisy blob dataset, the triplet accuracy was higher for classical MDS (0.89) than for metric MDS (0.83). The metric MDS had the shape predicted above in Section 4: it was mostly contained within the circle with radius $\bar{D}\pi/4$ (Figure 3b, dashed line), with higher density along the circle and lower density inside it. Indeed, the high-dimensional distances in this dataset were tightly concentrated around $\bar{D} \approx \sqrt{2 \cdot 1000}$ (Figure 3d), making it impossible to reproduce them in 2D.

Using the scikit-learn implementation of non-metric MDS with our improvements (see Section 3), we found that non-metric MDS resulted in a clearer class separation (Figure 3c) and the highest triplet

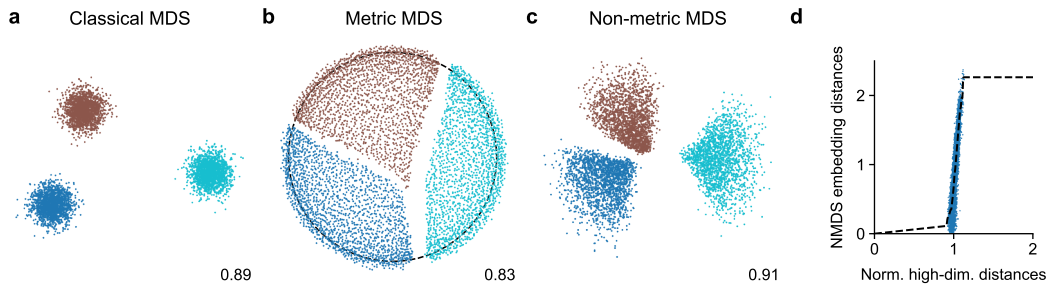

Figure 3: Classical, metric, and non-metric MDS embeddings of the noisy blobs dataset ($n = 5\,000$ in 1002 dimensions). Values in the lower-right are triplet accuracies (see text). **(b–c)** Embeddings obtained using the SMACOF implementation in scikit-learn with our improvements. The dashed circle in (b) shows our theoretical prediction and has radius $\bar{D}\pi/4$, where $\bar{D}$ is the average high-dimensional pairwise distance. **(d)** Shepard diagram for the non-metric MDS: embedding distances plotted against normalized high-dimensional distances. Dashed line shows monotonic regression fit.

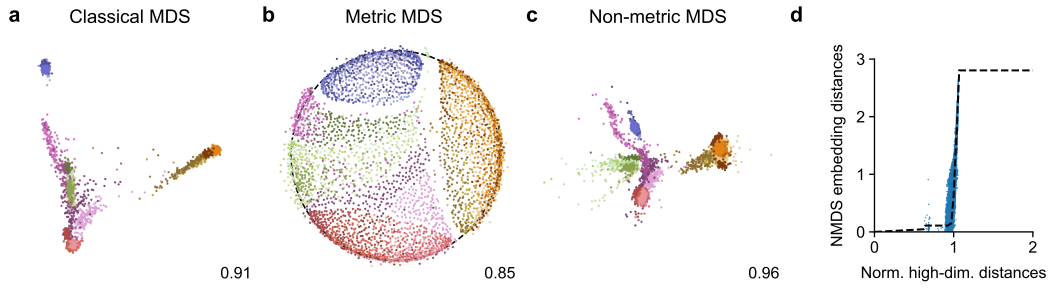

Figure 4: Classical, metric, and non-metric MDS embeddings of the 1000 Genomes dataset ($n = 3450$ in 53 999 dimensions). See caption of Figure 3 for explanations. Colors correspond to the sampling population (26 global populations), see The 1000 Genomes Project Consortium (2015). The dashed circle in (b) shows our theoretical prediction and has radius $\bar{D}\pi/4$.

accuracy (0.91). Isotonic regression transformed tightly concentrated high-dimensional distances to a range between 0 and 2 (Figure 3d; in the MDS literature this kind of plot is known as *Shepard diagram*). Pearson and Spearman evaluations yielded the same conclusions (Tables S2).

We observed the same behavior in the 1000 Genomes dataset (The 1000 Genomes Project Consortium, 2015), which we preprocessed following de Bodt et al. (2025), resulting in 53 999 feature dimensions (Figure 4). Here input distances were also highly concentrated (Figure 4d). As a result, metric MDS struggled to represent the data and produced a nearly spherical embedding with triplet accuracy (0.85) lower than that of classical MDS (0.91). The radius of the metric MDS embedding was in good agreement with our theoretical prediction (Figure 4b). In contrast, non-metric MDS produced a more informative embedding with better class separation and much higher triplet accuracy (0.96). In genomics it is common to pre-process the data with PCA. Doing it prior to metric MDS improved the triplet accuracy of the resulting embedding (50 PCA dimensions: 0.88; 10 dimensions: 0.91) but did not reach the level of non-metric MDS.

On the other hand, on the Digits dataset (available in scikit-learn) there was little difference between the performance of metric and non-metric MDS (Figure S3). This makes sense because the pairwise distances in this dataset were less tightly concentrated (Figure S3d).

7.2 Stochastic non-metric MDS is accurate and fast

Using all three datasets presented above (noisy blobs, 1000 Genomes, and Digits), we compared our stochastic implementation with the scikit-learn implementation with our modifications. We found that both implementations achieved the same stress-1 values on all three datasets and for both metric/non-metric MDS (Table 1, left). We confirmed that stress-1 computed on the entire dataset

Table 1: Stress-1 values and runtimes using the SMACOF and SGD implementations. On all datasets, our stochastic implementation achieves the same embedding quality as the reference non-stochastic implementation while being much faster. Tasic and MNIST datasets are too large for the non-stochastic implementation. The runtimes correspond to setup described in the text. Stress-1 values for SGD are the average over batches in the last epoch.

	Stress-1					**Runtime**				
	Blobs	Gen.	Digits	Tasic	MNIST	Blobs	Gen.	Digits	Tasic	MNIST
n						5 000	3 450	1 797	23 822	70 000
Metric	0.40	0.41	0.33	–	–	99 s	72 s	9 s	–	–
with SGD	0.40	0.41	0.33	0.25	0.36	73 s	15 s	4 s	4 m	14 m
Non-metric	0.25	0.13	0.29	–	–	892 s	204 s	131 s	–	–
with SGD	0.25	0.13	0.29	0.07	0.29	73 s	20 s	6 s	37 m	17 m

Table 2: Triplet accuracies. The highest value in each column is highlighted in bold, together with all values within 0.02. Non-metric MDS always achieves the best (or one of the best) result(s). See Tables S2 for Pearson and Spearman correlations between pairwise distances. Scikit-learn implementation for the first three datasets, our stochastic SGD for the last two.

	Blobs	Gen.	Digits	Tasic	MNIST
Classical	0.89	0.91	0.80	**0.99**	0.72
Metric	0.83	0.85	**0.84**	0.94	0.80
Non-metric	**0.91**	**0.96**	**0.86**	**0.99**	**0.83**
t-SNE	0.86	0.79	0.74	0.78	0.66
UMAP	0.86	0.68	0.68	0.79	0.64

after SGD was very close (differed at most by 0.01) to the within-batch estimates of stress-1 in the last epoch. At the same time, the stochastic implementation was much faster in all cases (Table 1, right). The speed-up was particularly noticeable for non-metric MDS.

The scikit-learn implementation constructs the full $\mathbf{D}$ matrix in memory, which becomes prohibitive for $n \gg 10\,000$. Due to memory overflow, we were unable to run it on our two further datasets: the Tasic dataset of single-cell transcriptomic data from mouse cortex (Tasic et al., 2018), analyzed previously in Kobak & Berens (2019) and de Bodt et al. (2025) (we followed the preprocessing from these two papers), and the MNIST dataset. Therefore, for these two datasets, all following results were obtained using our stochastic implementation.

On the Tasic dataset, non-metric MDS produced an embedding with the shared highest triplet accuracy (0.99, as for PCA, vs. 0.93 for metric MDS) and the highest Spearman correlation (0.96, vs. 0.91–0.92 for PCA and metric MDS) (Figure 5). One salient feature of this dataset is that it contains a small amount of various non-neural cells that transcriptomically are (a) far away from all neural cells; and (b) far away from each other. Non-metric MDS emphasized this structure by locating non-neural cells along an arc going around the cluster of neural cells. This happened at a price of squeezing all neural cells together, compared to the metric MDS embedding. Note that here we increased the n_{epochs} value for non-metric MDS to 500; our default value of 200 produced a similar-looking embedding but with a less pronounced arc structure.

On the MNIST dataset, the difference between metric and non-metric MDS was visually not very apparent (Figure S4), but non-metric MDS did show higher global structure preservation, especially for the distance correlations (Tables S2). Similar to the Digits dataset, the MNIST dataset arguably lacks global structure that can be represented by any MDS. One aspect of the data visible in the non-metric MDS embedding is the higher density of digits '1' (orange color in Figure S4c).

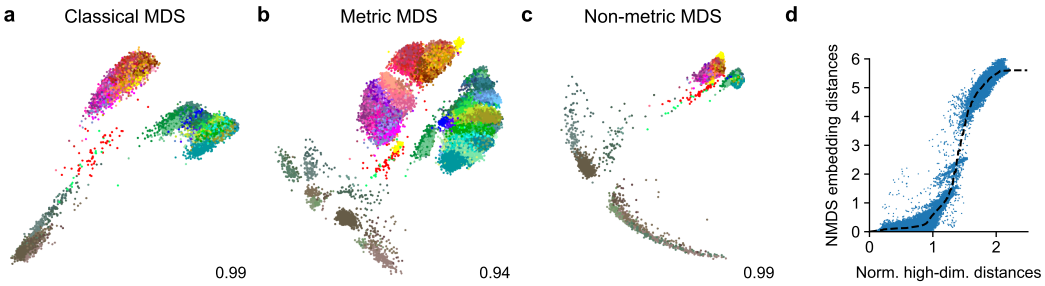

Figure 5: Classical, metric, and non-metric MDS embeddings of the Tasic dataset ($n = 23\,822$ in 50 dimensions). See caption of Figure 3 for explanations. Colors correspond to cell types (warm colors: inhibitory neurons; cold colors: excitatory neurons; brown/grey colors: non-neural cells). See Figure S5 for the same Shepard diagram as in (d), but colored by cell identity of cell pairs.

7.3 Across datasets, non-metric MDS achieves the highest global structure preservation

Across all five datasets, non-metric MDS achieved the highest performance in 15 out of 15 cases (five datasets and three evaluation metrics), including a shared highest in 5 out of 15 cases (Tables 2, S2).

For completeness, we also evaluated the t-SNE and the UMAP embeddings of all five datasets. These two algorithms are currently by far the most popular methods of representing high-dimensional data in 2D (de Bodt et al., 2025). Both t-SNE and UMAP are designed to preserve the nearest neighbors, which makes them excel at local structure preservation. These two algorithms would strongly outperform all flavors of MDS in terms of *local* quality metrics, such as kNN recall and kNN accuracy. In contrast, their capacity to preserve *global* structure of the data is known to be very poor (de Bodt et al., 2025) and indeed, this is what we confirmed on our datasets (Tables 2, S2).

8 Discussion

Our contribution in this paper is two-fold. Conceptually, we showed that metric MDS suffers from the curse of dimensionality. We derived the a perturbation bound stating that metric MDS remains close to a meaningless circular embedding under high-dimensional noise. We confirmed this effect in simulations and real datasets.

Practically, we developed a fast implementation of non-metric MDS, showed that non-metric MDS is more robust and can strongly outperform other embedding methods in terms of global structure preservation, in particular on datasets with prominent high-dimensional noise. Non-metric MDS was introduced over 60 years ago but has received almost no attention in modern manifold-learning and dimensionality-reduction literature, compared to metric MDS. We are not aware of any modern benchmark studies of dimensionality-reduction methods that include non-metric MDS or of any prior scalable implementations.

Future work and possible applications Several recent studies have developed hybrid methods combining neighbor-embedding methods such as t-SNE (van der Maaten & Hinton, 2008) or UMAP (McInnes et al., 2018) that exhibit high local structure preservation with global methods such as PCA or MDS. For example, SQuadMDS-hybrid (Lambert et al., 2022) includes both t-SNE and metric MDS losses in its composite loss term. Similarly, DREAMS (Kury et al., 2026) regularized t-SNE towards a pre-computed PCA or MDS embedding. Our results suggest that non-metric MDS can be a good choice for a global counterpart in such hybrid methods. Other examples of modern methods relying on metric MDS are PHATE (Moon et al., 2019) and HeatGeo (Huguet et al., 2023) which approximate diffusion-based graph distances in 2D. Here also, non-metric MDS could provide additional flexibility to embed the diffusion-based distances.

A recent work by Li et al. (2025) suggested a deflation procedure that can remove the contribution of high-dimensional noise from a distance matrix. It would be interesting to study if this can be combined with metric MDS to achieve similar results as with non-metric MDS.

AI USE STATEMENT

In this work, we used generative AI tools (Claude and ChatGPT) to formulate mathematical claims, to provide critical ingredients for the proofs, and to writing down the final proofs. We used interactively sessions with Claude 4.7 for Theorem 1 and Appendix C and ChatGPT 5.5 for Proposition 2 and Appendix D. We manually and in detail checked the correctness of all generated proofs. For instance, Claude's initial proof of Theorem 1 only showed that the KKT conditions lead to the Riesz equilibrium measure, but did not verify that it actually satisfies the KKT conditions. Moreover, there was initially no convexity argument and Claude only claimed that ρ^* is optimal among radially symmetric distributions. When prompted to verify the off-support KKT condition, Claude hallucinated a *minimum* principle for subharmonic functions. But when we pushed back, it came up with the correct solution based on the *maximum* principle for subharmonic functions, which indeed exists and yields a valid proof.

We did not use generative AI for other purposes. All code and experiments were implemented manually, and the manuscript text was written by hand.

ETHICS STATEMENT

We do not see any ethical issues with this work.

REPRODUCIBILITY STATEMENT

We aid reproducibility by fully describing our stochastic MDS algorithms in Section 6, including default hyperparameters and our compute setup. The full PyTorch implementation is also included as Listing S1. We submit the code required for running the experiments and for creating figures as part of the supplementary material and will release it on GitHub .

We included detailed proofs of all our theoretical claims in Appendices C and D.

REFERENCES

Charu C Aggarwal, Alexander Hinneburg, and Daniel A Keim. On the surprising behavior of distance metrics in high dimensional space. In *International conference on database theory*, pp. 420–434. Springer, 2001.

Dimitris K Agrafiotis. Stochastic proximity embedding. *Journal of Computational Chemistry*, 24 (10):1215–1221, 2003.

Dimitris K Agrafiotis, Huafeng Xu, Fangqiang Zhu, Deepak Bandyopadhyay, and Pu Liu. Stochastic proximity embedding: methods and applications. *Molecular Informatics*, 29(11):758–770, 2010.

Ery Arias-Castro and Siddharth Vishwanath. Stability of sequential lateration and of stress minimization in the presence of noise. *arXiv preprint arXiv:2310.10900*, 2023.

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, 37(1):38–44, 2019.

Ingwer Borg and Patrick JF Groenen. *Modern multidimensional scaling: Theory and applications*. Springer, 2005.

Andreas Buja, BF Logan, JA Reeds, and Larry A Shepp. Inequalities and positive-definite functions arising from a problem in multidimensional scaling. *The Annals of Statistics*, pp. 406–438, 1994.

Djalil Chafai, Edward B Saff, and Robert S Womersley. Threshold condensation to singular support for a riesz equilibrium problem. *arXiv preprint arXiv:2206.04956*, 2022.

Cyril de Bodt, Alex Diaz-Papkovich, Michael Bleher, Kerstin Bunte, Corinna Coupette, Sebastian Damrich, Enrique Fita Sanmartin, Fred A Hamprecht, Emőke-Ágnes Horvát, Dhruv Kohli, Smita Krishnaswamy, John A. Lee, Boudewijn P. F. Lelieveldt, Leland McInnes, Ian T. Nabney, Maximilian Noichl, Pavlin G. Poličar, Bastian Rieck, Guy Wolf, Gal Mishne, and Dmitry Kobak. Low-dimensional embeddings of high-dimensional data. *arXiv preprint arXiv:2508.15929*, 2025.

Jan de Leeuw. Application of convex analysis to multidimensional scaling. In *Recent Developments in Statistics*, pp. 133–145. North-Holland, 1977.

Jan de Leeuw and Patrick Mair. Multidimensional scaling using majorization: SMACOF in R. *Journal of Statistical Software*, 31:1–30, 2009.

Chengyuan Deng, Jie Gao, Kevin Lu, Feng Luo, Hongbin Sun, and Cheng Xin. Neuc-MDS: Non-Euclidean multidimensional scaling through bilinear forms. *Advances in Neural Information Processing Systems*, 37:121539–121569, 2024.

Zhongxi Fang, Xun Su, Tomohisa Tabuchi, Jianming Huang, and Hiroyuki Kasai. StableMDS: A novel gradient descent-based method for stabilizing and accelerating weighted multidimensional scaling. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

John C Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53(3-4):325–338, 1966.

Peter Hall, James Stephen Marron, and Amnon Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67 (3):427–444, 2005.

Daniel Hangan, Stephen Kobourov, and Jacob Miller. Bridging graph drawing and dimensionality reduction with stochastic stress optimization, 2026. URL `https://arxiv.org/abs/2605.00641`.

Yasuaki Hiraoka, Yusuke Imoto, Shu Kanazawa, and Enhao Liu. Curse of dimensionality on persistence diagrams. *Foundations of Data Science*, 8:236–286, 2026.

Haiyang Huang, Yingfan Wang, Cynthia Rudin, and Edward P Browne. Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. *Communications biology*, 5(1):719, 2022.

Guillaume Huguet, Alexander Tong, Edward De Brouwer, Yanlei Zhang, Guy Wolf, Ian Adelstein, and Smita Krishnaswamy. A heat diffusion perspective on geodesic preserving dimensionality reduction. *Advances in Neural Information Processing Systems*, 36:6986–7016, 2023.

Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications*, 10(1):5416, 2019.

Alexey Kroshnin, Eugene Stepanov, and Dario Trevisan. Infinite multidimensional scaling for metric measure spaces. *ESAIM: Control, Optimisation and Calculus of Variations*, 28(58), 2022.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964a.

Joseph B Kruskal. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2): 115–129, 1964b.

Noël Kury, Dmitry Kobak, and Sebastian Damrich. DREAMS: Preserving both local and global structure in dimensionality reduction. *Transactions on Machine Learning Research*, 2026.

Pierre Lambert, Cyril de Bodt, Michel Verleysen, and John A Lee. SQuadMDS: a lean stochastic quartet MDS improving global structure preservation in neighbor embedding like t-SNE and UMAP. *Neurocomputing*, 503:17–27, 2022.

Naum S Landkof. *Foundations of modern potential theory*, volume 180. Springer, 1972.

John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer, 2007.

Keyi Li, Yuval Kluger, and Boris Landa. Euclidean distance deflation under high-dimensional heteroskedastic noise. *arXiv*, 2025.

Sunhyuk Lim and Facundo Mémoli. Classical multidimensional scaling on metric measure spaces. *Information and Inference: A Journal of the IMA*, 13(2):iaae007, 2024.

Patrick Mair, Patrick JF Groenen, and Jan de Leeuw. More on multidimensional scaling and unfolding in R: smacof version 2. *Journal of Statistical Software*, 102:1–47, 2022.

L Mattner. Extremal problems for probability distributions: a general method and some examples. *Lecture Notes-Monograph Series*, pp. 274–283, 1992.

Lutz Mattner. Behandlung einiger extremalprobleme für wahrscheinlichkeitsverteilungen. *(No Title)*, 1990.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Kevin R Moon, David Van Dijk, Zheng Wang, Scott Gigante, Daniel B Burkhardt, William S Chen, Kristina Yim, Antonia van den Elzen, Matthew J Hirn, Ronald R Coifman, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12): 1482–1492, 2019.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Marcel Riesz. Intégrales de Riemann–Liouville et potentiels. In *Collected Papers*, pp. 482–526. Springer, 1938.

Isaac J Schoenberg. Remarks to Maurice Frechet's article "Sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de Hhilbert". *Annals of Mathematics*, 36(3):724–732, 1935.

Isaac J Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44(3):522–536, 1938.

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *The Annals of Statistics*, 41(5): 2263–2291, 2013.

Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, 1962a.

Roger N Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika*, 27(3):219–246, 1962b.

Rishi Sonthalia, Greg Van Buskirk, Benjamin Raichel, and Anna Gilbert. How can classical multidimensional scaling go wrong? *Advances in Neural Information Processing Systems*, 34: 12304–12315, 2021.

Gábor J Székely and Maria L Rizzo. Energy statistics: A class of statistics based on distances. *Journal of Statistical Planning and Inference*, 143(8):1249–1272, 2013.

Bosiljka Tasic, Zizhen Yao, Lucas T. Graybuck, Kimberly A. Smith, Thuc Nghi Nguyen, Darren Bertagnolli, Jeff Goldy, Emma Garren, Michael N. Economo, Sarada Viswanathan, Osnat Penn, Trygve Bakken, Vilas Menon, Jeremy Miller, Olivia Fong, Karla E. Hirokawa, Kanan Lathia, Christine Rimorin, Michael Tieu, Rachael Larsen, Tamara Casper, Eliza Barkan, Matthew Kroll, Sheana Parry, Nadiya V. Shapovalova, Daniel Hirschstein, Julie Pendergraft, Heather A. Sullivan, Tae Kyung Kim, Aaron Szafer, Nick Dee, Peter Groblewski, Ian Wickersham, Ali Cetin, Julie A. Harris, Boaz P. Levi, Susan M. Sunkin, Linda Madisen, Tanya L. Daigle, Loren Looger, Amy Bernard, John Phillips, Ed Lein, Michael Hawrylycz, Karel Svoboda, Allan R. Jones, Christof Koch, and Hongkui Zeng. Shared and distinct transcriptomic cell types across neocortical areas. *Nature*, 563(7729):72–78, 2018.

The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526 (7571):68–74, 2015.

Warren S Torgerson. Multidimensional scaling: I. Theory and method. *Psychometrika*, 17(4): 401–419, 1952.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

Jonathan X Zheng, Samraat Pawar, and Dan FM Goodman. Graph drawing by stochastic gradient descent. *IEEE Transactions on Visualization and Computer Graphics*, 25(9):2738–2748, 2018.

A SUPPLEMENTARY FIGURES, TABLES, AND LISTINGS

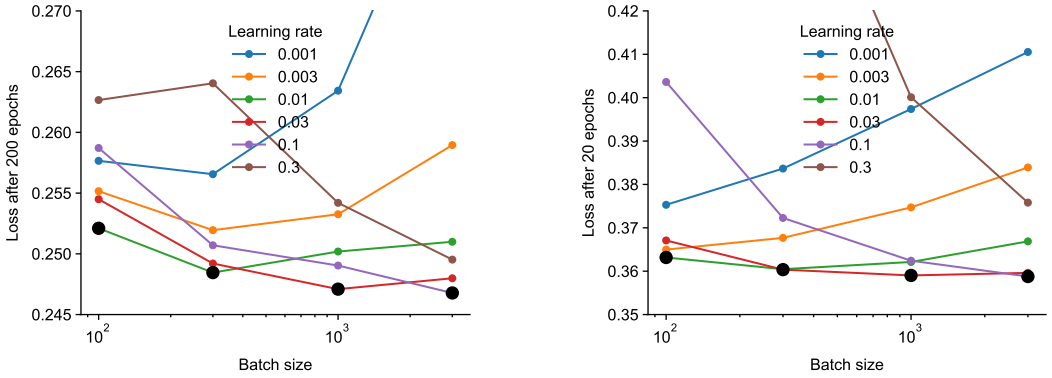

Figure S1: Stress-1 obtained using the SGD implementation of metric MDS using different batch sizes and learning rates for the Tasic (left; 200 epochs) and MNIST (right; 20 epochs) datasets. It is beneficial to use larger batch sizes together with higher learning rates. The runtime scales linearly with the batch size, see text.

Table S1: Runtime and final loss of our stochastic non-metric MDS implementation depending on how often the isotonic regression is run. Results are shown for the Blobs and the MNIST datasets. All hyperparameters (batch size, number of epochs, etc.) are kept constant at the values described in the main text (Blobs: batch size 100, sample size 5000, 200 epochs; MNIST: batch size 1000, sample size 70 000, 20 epochs).

	Blobs		MNIST	
	Time	Loss	Time	Loss
Once every 100 epochs	1min 16s	0.249	—	—
Once every 10 epochs	1min 20s	0.247	11min 32s	0.297
Once every epoch (default)	1min 20s	0.249	10min 22s	0.293
Once every 10 batches	1min 23s	0.247	10min 57s	0.293
Every batch	1min 58s	0.245	17min 38s	0.292

Table S2: Pearson and Spearman correlation coefficients between pairwise distances in the high-dimensional space and in the embedding. The highest value in each column is highlighted in bold, together with all values within 0.02. Non-metric MDS always achieves the best (or one of the best) result(s).

	Pearson					**Spearman**				
	Blobs	Gen.	Digits	Tasic	MNIST	Blobs	Gen.	Digits	Tasic	MNIST
Classical	0.76	0.82	0.57	**0.92**	0.53	0.78	0.84	0.56	0.91	0.51
Metric	0.67	0.69	**0.72**	**0.90**	0.65	0.67	0.71	**0.73**	0.92	0.65
Non-metric	**0.84**	**0.90**	**0.74**	**0.91**	**0.78**	**0.85**	**0.96**	**0.75**	**0.96**	**0.80**
t-SNE	0.68	0.65	0.54	0.46	0.41	0.70	0.62	0.50	0.55	0.39
UMAP	0.72	0.45	0.39	0.54	0.36	0.74	0.30	0.37	0.55	0.34

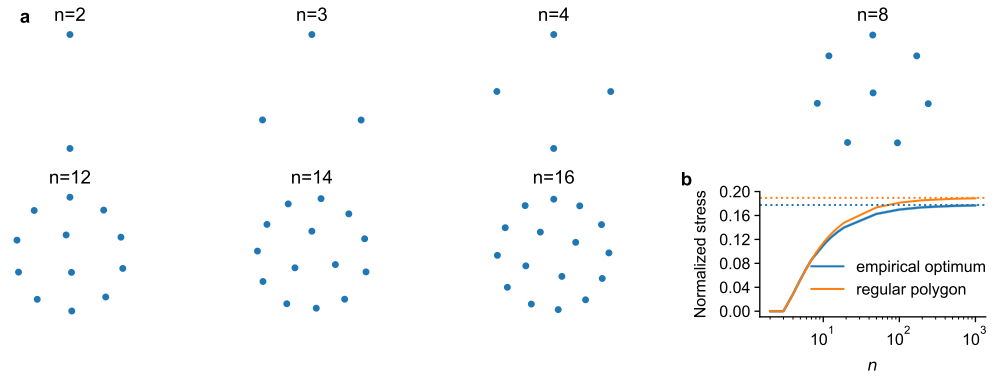

Figure S2: Empirically optimal 2D MDS layouts of n equidistant points, obtained using the scikit-learn implementation. **(a)** For $n = 2$ and 3 exact equidistance can be achieved. Starting with $n = 4$ pairwise equidistance needs to be approximated. Empirically, for $n = 4, \ldots, 7$ a regular polygon is optimal. At $n \geq 8$ our experiments showed that adding additional points inside the polygon achieves lower loss. **(b)** Normalized stress values for the regular polygons and MDS embeddings (averaged over 3 seeds, standard deviations were negligible), as a function of sample size n. Dotted lines show the optimal stress for the continuum limits from Lemma 4 and Theorem 1.

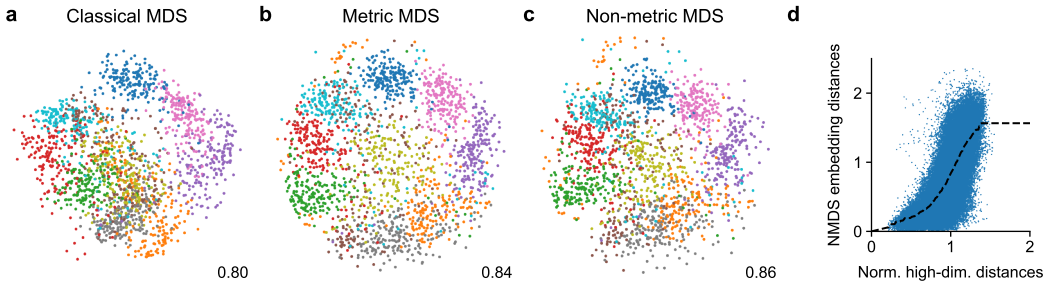

Figure S3: Classical, metric, and non-metric MDS embeddings of the Digits dataset ($n = 1797$ in 64 dimensions). See caption of Figure 3 for explanations.

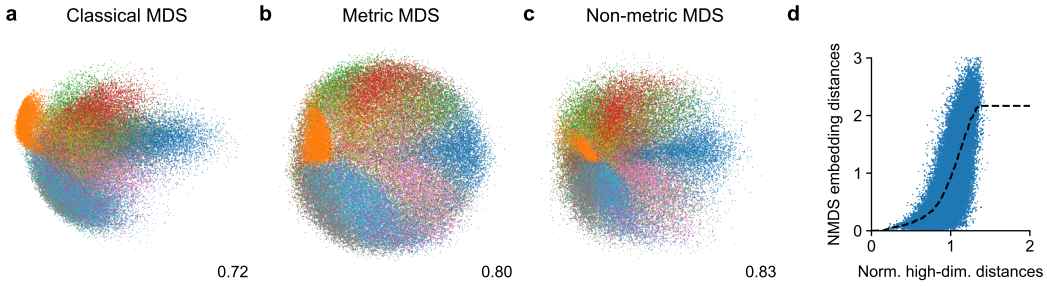

Figure S4: Classical, metric, and non-metric MDS embeddings of the MNIST dataset ($n = 70\,000$ in 784 dimensions). See caption of Figure 3 for explanations.

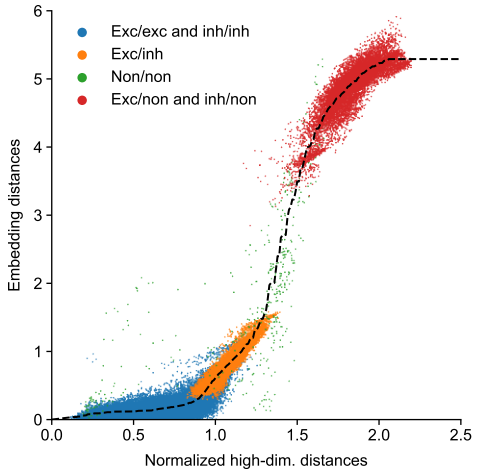

Figure S5: Shepard diagram for the non-metric MDS embedding of the Tasic dataset. Reproduced from Figure 5d but here each pair of cells is colored based on the cells' identities: excitatory/excitatory and inhibitory/inhibitory neuron pairs are shown in blue; excitatory/inhibitory neuron pairs — in orange; non-neural/non-neural pairs — in green; neural/non-neural pairs — in red.

```python
import numpy as np
import time
import torch
from sklearn.manifold import ClassicalMDS
from sklearn.decomposition import PCA
from sklearn.isotonic import IsotonicRegression
from scipy.spatial.distance import cdist

def stochasticMDS(
    X,
    n_components=2,
    precomputed_distances=False,
    non_metric=False,
    verbose=True,
    n_epochs="auto",
    batch_size="auto",
    print_every_epoch="auto",
    lr=0.02,
    start_nonmetric_from_epoch=0,
    init_scale=0.5,
    init=None,
    seed=42,
):

    ### Default hyperparameter values
    sample_size = X.shape[0]

    if batch_size == "auto":
        if sample_size < 5000:
            batch_size = 100
        else:
            batch_size = 1000

    if n_epochs == "auto":
        if sample_size < 50_000:
            n_epochs = 200
        else:
            n_epochs = 20

    if print_every_epoch == "auto":
        if n_epochs > 20:
            print_every_epoch = 10
        else:
            print_every_epoch = 1

    ### Initialization
    if init is not None:
        Z = init
    else:
        if precomputed_distances:
            Z = ClassicalMDS(
                n_components=n_components
            ).fit_transform(X)
        else:
            Z = PCA(
                n_components=n_components
            ).fit_transform(X)
        Z /= np.std(Z[:, 0])
        Z *= init_scale

    ### Set up the model
    torch.manual_seed(seed)
    model = torch.nn.Embedding(
        sample_size, n_components
    ).from_pretrained(
        torch.tensor(Z.astype(np.float32)),
        freeze=False
    )

    optimizer = torch.optim.Adam(
        model.parameters(),
        lr=lr
    )
    scheduler = (
        torch.optim.lr_scheduler.CosineAnnealingLR
    )(
        optimizer, T_max=n_epochs
    )

    if non_metric:
        ir = IsotonicRegression(out_of_bounds="clip")
```

```python

    ### Training loop
    for epoch in range(n_epochs):
        shuffle = torch.randperm(sample_size)

        epoch_loss = 0
        t_start = time.time()

        for batch in range(sample_size // batch_size):
            ind = shuffle[
                batch * batch_size :
                (batch + 1) * batch_size
            ]
            ind_np = ind.numpy()

            if precomputed_distances:
                dists = X[ind_np, :][:, ind_np]
            else:
                dists = cdist(X[ind_np], X[ind_np])

            optimizer.zero_grad()
            model_dists = torch.squeeze(
                torch.cdist(
                    model(ind)[None, :, :],
                    model(ind)[None, :, :]
                )
            )

            ### Non-metric part
            if non_metric:
                if epoch >= start_nonmetric_from_epoch:
                    if batch == 0:
                        ir.fit(
                            dists.ravel(),
                            (
                                model_dists.detach().
                                numpy().ravel()
                            )
                        )

                    dists = ir.transform(
                        dists.ravel()
                    ).reshape(
                        dists.shape
                    )

                    if batch == 0:
                        dists_scaling = np.sqrt(
                            np.sum(dists ** 2) / (batch_size ** 2 - batch_size)
                        )

                    dists /= dists_scaling

            dists = torch.tensor(
                dists.astype(np.float32)
            )

            ### Loss function = average stress
            loss = torch.mean(
                (model_dists - dists) ** 2
            )

            loss.backward()
            optimizer.step()
            epoch_loss += loss.item()

        scheduler.step()
        epoch_loss /= (sample_size // batch_size)
        t_end = time.time()

    Z = model.weight.detach().numpy()
    return Z, epoch_loss
```

Listing S1: Stochastic MDS implementation.

Table S3: Extended robustness analysis settings. The first line summarizes the setup for Fig. 2b, the other lines show the settings for the extended analysis.

Noise	σ	Jitter	Cluster sizes	
			Size	Sampling
Gaussian	1.0	No	$= 99$	Fixed
Gaussian	1.0	Yes	~ 99	Random
Uniform	1.0	Yes	~ 99	Random
Exponential	1.0	Yes	~ 99	Random
Gaussian	2.0	Yes	~ 99	Random
Gaussian	1.0	Yes	~ 297	Random

B Extended robustness analysis

In addition to the setup in Fig. 2, we also ran a larger suite of robustness experiments. In particular, we

- explored two dataset sizes, $n^* = 99$ and $n^* = 297$
- made the total number of points random by sampling $n \sim \text{Poisson}(n^*)$
- made the sizes of the three clusters random, by sampling their sizes from $\text{Multinomial}(n; 1/3, 1/3, 1/3)$
- varied the pairwise distances between clusters in 10 geometric steps between $D = 8$ and $D = 64$
- broke the noise-free equidistance of the clusters by jittering the points away from their cluster centers by standard Gaussians in the first two dimensions, scaled by $D/10$
- used different variance per noise dimension of $\sigma = 1$ and $\sigma = 2$
- varied the type of noise in the additional d noise dimension between Gaussian, uniform, and exponential
- increased the range of noise dimensions to $d \in [20, 15\,000\,000]$

All configurations are summarized in Tab. S3. We ran each (d, D) configuration 20 times and fitted power laws to the 10 points $\big(D, d^*(D/\sigma)\big)$, where d^* is the number of noise dimensions at which the kNN accuracy of the embedding dipped below 0.7. Finally, we averaged the slopes over the 20 of each configuration. For metric MDS, these mean slopes were within $[2.09, 2.20]$ and for non-metric MDS within $[4.00, 4.05]$ in close alignment with our theoretical predictions of 2 and 4, respectively.

C Optima of metric MDS for equidistant inputs

Lemma 3 (Optimal regular polygons)**.** *For $n \geq 2$ points placed on a regular n-gon of circumradius r, the stress-minimizing radius is*

$$r_n^* = \frac{1}{n} \cot \frac{\pi}{2n} \xrightarrow{n\to\infty} 2/\pi.$$

The corresponding optimal normalized stress is

$$S_n^* = 1 - \frac{2}{n(n-1)} \cot^2\left(\frac{\pi}{2n}\right) \xrightarrow{n\to\infty} 1 - 8/\pi^2.$$

Proof. Place the vertices at $x_k = r\, e^{2\pi i k/n}$ for $k = 0, \ldots, n-1$. The chord length between vertices k and $k+j$ is $\|x_k - x_{k+j}\| = 2r \sin(\pi j/n)$. For each gap $j = 1, \ldots, n-1$ there are exactly n pairs $(i,\, i+j)$, so the average per-pair stress is

$$S_n = \frac{n}{n(n-1)} \sum_{j=1}^{n-1} \big(2r \sin(\tfrac{\pi j}{n}) - 1\big)^2 = \frac{1}{n-1} \sum_{j=1}^{n-1} \big(2r \sin(\tfrac{\pi j}{n}) - 1\big)^2.$$

Expanding and applying the identities $\sum_{j=1}^{n-1}\sin^2(\pi j/n) = n/2$ and $\sum_{j=1}^{n-1}\sin(\pi j/n) = \cot(\pi/(2n))$, we get

$$S_n = \frac{2n}{n-1}\,r^2 - \frac{4\cot(\pi/(2n))}{n-1}\,r + 1.$$

This convex quadratic in r is minimized at $r_n^* = \cot(\pi/(2n))/n$. The corresponding minimal value of S_n is

$$S_n^* = 1 - \frac{2}{n(n-1)}\cot^2\left(\frac{\pi}{2n}\right).$$

As $n \to \infty$, $\cot(\pi/(2n)) \sim 2n/\pi$, so $r_n^* \to 2/\pi$ and $S_n^* \to 1 - 8/\pi^2$. □

Recall that $\mathcal{P}(\mathbb{R}^2)$ denotes the set of probability measures on $\mathbb{R}^2$. For $\rho \in \mathcal{P}(\mathbb{R}^2)$ and $X, Y \overset{\text{i.i.d.}}{\sim} \rho$ the normalized stress functional (corresponding to the raw stress divided by $n(n-1)/2$, not to the stress-1) was defined in Eq. 5 as

$$S[\rho] = \mathbb{E}_\rho\big[(\|X - Y\| - 1)^2\big] = \iint \big(\|x - y\| - 1\big)^2 \rho(x)\rho(y)\,dx\,dy. \tag{9}$$

Lemma 4 (Optimal circle)**.** *For the uniform measure μ_r on a circle of radius r, the stress-minimizing radius is $r^* = 2/\pi$, with optimal stress*

$$S[\mu_{2/\pi}] = 1 - 8/\pi^2 \approx 0.1894$$

Proof. This follows by a limit argument from Lemma 3, but for the convenience of the reader, we include a direct proof as well. For $X, Y \overset{\text{i.i.d.}}{\sim} \mu_r$, the angle between them Φ is distributed according to $\mathrm{Uniform}[0, 2\pi)$ and $\|X - Y\| = 2r\sin(\Phi/2)$. Using that $\int_0^{2\pi}\sin(\phi/2)\,d\phi = 4$ and $\int_0^{2\pi}\sin^2(\phi/2)\,d\phi = \pi$, we get

$$\mathbb{E}\|X - Y\| = \frac{4r}{\pi}, \qquad \mathbb{E}\|X - Y\|^2 = 2r^2.$$

Hence $S[\mu_r] = 2r^2 - 8r/\pi + 1$, which is minimized at $r^* = 2/\pi$ with optimal stress $1 - 8/\pi^2$. This confirms the continuous limit. □

The following theorem was shown independently by Mattner (1990; 1992) and Buja et al. (1994).

Theorem 5 (Limiting MDS embedding, detailed version of Thm. 1)**.** *Up to translation, the unique minimizer of $S[\rho]$ over $\mathcal{P}(\mathbb{R}^2)$ is the Riesz $s = 1$ equilibrium measure with density*

$$\rho^*(x) = \frac{2}{\pi^2\sqrt{(\pi/4)^2 - \|x\|^2}}\,\mathbf{1}\left(\|x\| < \frac{\pi}{4}\right). \tag{10}$$

It achieves optimal stress

$$S^* = 1 - \frac{\pi^2}{12} \approx 0.1775. \tag{11}$$

Proof. We proceed in eight steps.

Since $S[\rho]$ depends only on pairwise distances $\|X - Y\|$, translating ρ does not affect $S[\rho]$, so we can assume $\mathbb{E}[X] = 0$ without loss of generality. We show in Step 1 that S is strictly convex on the set of zero-centered probability distributions ρ. As a result, any such ρ satisfying the Karush–Kuhn–Tucker (KKT) conditions is the unique global minimizer; Steps 2–7 then confirm ρ^* is that point. Note that we need KKT conditions because our optimization problem has both equality and inequality constraints: we optimize Eq. 9 over ρ that has to be non-negative and integrate to one.

Step 1: Strict Convexity of S. To show that S is convex, we consider two zero-centered probability measures $\rho \neq \rho'$. Denoting $\eta = \rho' - \rho$, we have

$$\begin{aligned}
S[\rho'] = S[\rho + \eta] &= \iint \big(\|x - y\| - 1\big)^2 \big(\rho(x) + \eta(x)\big)\big(\rho(y) + \eta(y)\big)\,dx\,dy \\
&= S[\rho] + 2\int\left(\int \big(\|x - y\| - 1\big)^2 \rho(y)\,dy\right)\eta(x)\,dx + \iint \big(\|x - y\| - 1\big)^2 \eta(x)\eta(y)\,dx\,dy \\
&= S[\rho] + 2\int \frac{\delta S}{\delta \rho}(x)\eta(x)\,dx + S[\eta].
\end{aligned}$$

Strict convexity means that the functional increases faster than a local linear approximation. The second term is linear in η (and the functional $\eta \mapsto \int \frac{\delta S}{\delta \rho}\eta$ is the local linear approximation at ρ), so to demonstrate strict convexity we need to show that $S[\eta] > 0$. We have

$$\begin{aligned}
S[\eta] &= \iint \big(\|x-y\|-1\big)^2 \,\eta(x)\eta(y)\,dx\,dy \\
&= \iint \big(\|x\|^2 - 2x^\top y + \|y\|^2 - 2\|x-y\| + 1\big)\,\eta(x)\eta(y)\,dx\,dy \\
&\overset{(a)}{=} \iint \big(-2x^\top y - 2\|x-y\|\big)\,\eta(x)\eta(y)\,dx\,dy \\
&= -2\int x^\top \eta(x)\,dx \int y\eta(y)\,dy - 2\iint \|x-y\|\eta(x)\eta(y)\,dx\,dy \\
&\overset{(b)}{=} -2\iint \|x-y\|\eta(x)\eta(y)\,dx\,dy \\
&\overset{(c)}{>} 0,
\end{aligned}$$

where we dropped terms not depending on both x and y at (a) because $\int \eta = 0$ and used $\mathbb{E}_\eta(X) = 0$ at (b). Inequality (c) follows because $\eta \neq 0$ with $\int \eta = 0$ and $\|x-y\|$ is a strictly conditionally negative definite kernel (Schoenberg, 1938).

Thus, S is strictly convex on the set of zero-centered probability distributions.

Step 2: Expand $S[\rho]$. Expanding the square in equation 9, we get

$$S[\rho] = \mathbb{E}\|X-Y\|^2 - 2\mathbb{E}\|X-Y\| + 1. \tag{12}$$

Note that this identity only holds if all terms on the right are finite. We will now show that this is true for any ρ for which $S[\rho] < \infty$, and we are only interested in this case as we are looking for the minimum of S. The pointwise bound $a^2 \leq 2(a-1)^2 + 2$, valid since $(a-2)^2 \geq 0$, gives $\mathbb{E}\|X-Y\|^2 \leq 2S[\rho] + 2 < \infty$. We denote $\mathbb{E}\|X\|^2$ as σ^2. Conditioning on X yields

$$\mathbb{E}_Y\big[\|X-Y\|^2\big] = \mathbb{E}_Y\big[\|X\|^2 - 2X\cdot Y + \|Y\|^2\big] = \|X\|^2 + \mathbb{E}\|Y\|^2 = \|X\|^2 + \sigma^2,$$

where we used $\mathbb{E}[Y] = 0$ and $\mathbb{E}\|Y\|^2 = \sigma^2$. Taking the outer expectation, we get $\mathbb{E}\|X-Y\|^2 = 2\sigma^2$. The left part is finite, hence $\sigma^2 < \infty$.

Thus,

$$S[\rho] = 2\sigma^2 - 2\mathbb{E}\|X-Y\| + 1.$$

Step 3: KKT conditions. We can now compute the functional derivative of $S[\rho]$ with respect to ρ. For

$$\sigma^2[\rho] = \int \|x\|^2 \rho(x)\,dx,$$

the functional is linear in ρ (the integrand $\|x\|^2$ does not depend on ρ), so its functional derivative is simply the integrand:

$$\frac{\delta \sigma^2}{\delta \rho}(x) = \|x\|^2.$$

For the interaction term

$$\mathbb{E}\|X-Y\| = \iint \|x-y\|\,\rho(x)\rho(y)\,dx\,dy,$$

the functional is quadratic in ρ with symmetric kernel $k(x,y) = \|x-y\|$. Thus by the product rule we get

$$\frac{\delta}{\delta\rho(x)}\big(\mathbb{E}\|X-Y\|\big) = 2\int \|x-y\|\,\rho(y)\,dy.$$

Together we obtain the functional derivative

$$\frac{\delta S}{\delta \rho}(x) = 2\|x\|^2 - 4\int \|x-y\|\,\rho(y)\,dy.$$

At the minimum, the local linear approximation to S must equal zero or some constraint must be active. Taking the expression for the local linear approximation from Step 1, we obtain the condition for the minimum as

$$\int \frac{\delta S}{\delta \rho}(x)\eta(x)\,dx = 0,$$

for all η such that $\eta(x) = 0$ for $x \notin \operatorname{supp}(\rho)$. In words, at the optimum, moving an infinitesimal bit of mass from any occupied point $x \in \operatorname{supp}(\rho)$ to any other occupied point $x' \in \operatorname{supp}(\rho)$ must leave S unchanged to first order. This is of course satisfied when $\frac{\delta S}{\delta \rho} = 0$, but not only then. As we are optimizing over the space of probability distributions, the constraint of them integrating to 1 implies that $\int \eta = 0$, so the condition is also satisfied when $\frac{\delta S}{\delta \rho} = C$ for any constant C. This is effectively the Lagrange multiplier arising from the equality constraint.

Off the support, we cannot decrease ρ further due to the inequality constraint $\rho(x) \geq 0$. So η can only place mass there, not take it away. To ensure that placing mass off the support with some perturbation η that has $\eta(x) > 0$ for some $x \notin \operatorname{supp}(\rho)$ does not decrease the functional, $\frac{\delta S}{\delta \rho}(x)$ must be at least as large as on the support.

Taking this together, we arrive at the full set of KKT conditions at a minimizer:

$$\frac{\delta S}{\delta \rho}(x) = C \quad \text{for } x \in \operatorname{supp}(\rho),$$
$$\frac{\delta S}{\delta \rho}(x) \geq C \quad \text{for } x \notin \operatorname{supp}(\rho).$$

Rearranging the on-support condition gives

$$2 \int \|x - y\|\, \rho(y)\, dy = \|x\|^2 - \tfrac{1}{2}C \qquad \text{for } x \in \operatorname{supp}(\rho). \tag{$\star$}$$

Step 4: Laplacian trick. Apply Δ_x to both sides of equation $\star$. For the right-hand side, $\Delta_x\big(\|x\|^2\big) = 4$ in $\mathbb{R}^2$. For the left-hand side, we use

$$\Delta_x \|x - y\| = \frac{1}{\|x - y\|} \qquad \text{for } x \neq y. \tag{13}$$

To see this, note that we can assume $y = 0$ by translation. Then for $r = \|x\|$ we have $\Delta r = r'' + r'/r = 0 + 1/r$ by the radial Laplacian formula in $\mathbb{R}^2$. Since $1/r$ is locally integrable in $\mathbb{R}^2$, $\Delta_x \|x - y\| = 1/\|x - y\|$ distributionally with no delta correction at $x = y$. Commuting Δ_x with the integral (justified for ρ with $\sigma^2 < \infty$ by regularizing $\|x - y\|_\varepsilon = \sqrt{\|x - y\|^2 + \varepsilon^2}$, differentiating under the integral for fixed $\varepsilon > 0$, then passing $\varepsilon \to 0$ by dominated convergence) yields

$$\int \frac{\rho(y)}{\|x - y\|}\, dy = 2 \qquad \text{for } x \in \operatorname{supp}(\rho). \tag{$\star\star$}$$

Step 5: Riesz equilibrium measure. Equation equation $\star\star$ is the condition that the Riesz s=1 potential

$$V(x) = \int \frac{\rho(y)}{\|x - y\|}\, dy$$

is constant (and equal to 2) on $\operatorname{supp}(\rho)$. This is satisfied by the Riesz $s = 1$ equilibrium (Riesz, 1938; Landkof, 1972, § II.3.13, pp. 163–164, and Appendix, pp. 399–400; Chafai et al., 2022, Lem. 1.2)

$$\rho^*(x) = \frac{2}{\pi^2 \sqrt{(\pi/4)^2 - \|x\|^2}}\, \mathbf{1}\big(\|x\| < \pi/4\big). \tag{14}$$

Step 6: Verifying the on-support KKT condition. The minimizer ρ^* is radially symmetric on the disc $B(0, \pi/4)$, hence the functional derivative $\frac{\delta S}{\delta \rho}$ is also radially symmetric. The functional derivative is also harmonic because

$$\Delta \frac{\delta S}{\delta \rho}(x) = 8 - 4V(x) = 0.$$

Any radially symmetric harmonic function must be of the form $a \log \|x\| + b$. If we show that $\frac{\delta S}{\delta \rho}$ is finite at $x = 0$, it will follow that it must be constant. We denote $R = \pi/4$ and compute

$$\frac{\delta S}{\delta \rho}(0) = -4 \int \|y\| \, \rho^*(y) \, dy = -4 \int_0^R r \cdot \frac{4r}{\pi \sqrt{R^2 - r^2}} \, dr$$
$$\overset{(a)}{=} -\frac{16R^2}{\pi} \int_0^{\pi/2} \sin^2\theta \, d\theta = -\frac{16R^2}{\pi} \cdot \frac{\pi}{4} = -\frac{\pi^2}{4},$$

where at (a) we substituted $r = R\sin(\theta)$. Note that this implies $C = \frac{\delta S}{\delta \rho}(0) = -\pi^2/4$.

Step 7: Verifying the off-support KKT condition.

Part A: $V(x) = 2$ on $\partial B(0, \pi/4)$. By Step 5, we have $V(x) = 2$ for all $\|x\| < \pi/4$, and by continuity $V(x) = 2$ on the boundary.

Part B: $V(x) \leq 2$ for $\|x\| > \pi/4$. For $\|x\| > \pi/4$ (so $\|x - y\| > 0$ for all $y \in \text{supp}(\rho^*)$), differentiating under the integral and using $\Delta_x(1/\|x - y\|) = 1/\|x - y\|^3$ yields

$$\Delta_x V(x) = \int \frac{\rho^*(y)}{\|x - y\|^3} \, dy > 0.$$

So V is subharmonic on the unbounded domain $\mathbb{R}^2 \setminus \overline{B(0, \pi/4)}$, with $V = 2$ on $\partial B(0, \pi/4)$ and $V(x) \to 0$ as $\|x\| \to \infty$. By the maximum principle, $V \leq 2$ on the entire exterior.

Part C: Boundary data at $r = \pi/4$. Define

$$g(x) = \frac{\delta S}{\delta \rho}(x) - C = 2\|x\|^2 - 4 \int \|x - y\| \rho^*(y) \, dy - C.$$

so that the off-support KKT condition is $g(x) \geq 0$ for $\|x\| \geq \pi/4$. Since $\rho^* > 0$ on the open disk, the on-support equality equation $\star$ holds for all $\|x\| < \pi/4$, i.e. $g \equiv 0$ on $B(0, \pi/4)$. The function g is radial (since ρ^* is) and globally C^1: for

$$\nabla g(x) = 4x - 4 \int \frac{x - y}{\|x - y\|} \, \rho^*(y) \, dy,$$

the integrand is dominated by the L^1 function ρ^* (since $\left\|(x - y)/\|x - y\|\right\| = 1$), so dominated convergence yields continuity of ∇g on $\mathbb{R}^2$. Writing $g(r)$ for the radial profile, the interior identity $g \equiv 0$ on $B(0, \pi/4)$ together with C^1 matching across $r = \pi/4$ gives

$$g(\pi/4) = 0, \qquad g'(\pi/4) = 0. \tag{15}$$

Part D: Radial ODE outside the disk. For $\|x\| > \pi/4$, differentiating under the integral and using the 2D identity $\Delta_x \|x - y\| = 1/\|x - y\|$ (Step 4) gives

$$\Delta g(x) = 8 - 4\, V(x) \geq 0,$$

where the inequality is Part B. For radial g, $\Delta g = g'' + g'/r$, so

$$\big(r \, g'(r)\big)' = r \, \Delta g(r) \geq 0 \qquad (r > \pi/4). \tag{16}$$

Part E: Two integrations. By equation 16, the function $r \, g'(r)$ is non-decreasing on $(\pi/4, \infty)$. For any $r > \pi/4$, the fundamental theorem of calculus and the boundary data equation 15 give

$$r \, g'(r) \geq \pi/4 \cdot g'(\pi/4) = 0,$$

hence $g'(r) \geq 0$ for $r \geq \pi/4$. A second integration from $\pi/4$ to r yields

$$g(r) \geq g(\pi/4) = 0 \qquad (r \geq \pi/4),$$

which is the off-support KKT condition.

Step 8: Optimal stress. We compute the terms σ^2 and $\mathbb{E}\|X - Y\|$ in equation 12. First,

$$\sigma^2 = \int \|x\|^2 \rho(x) dx \overset{r=\|x\|}{=} \int_0^R r^2 \cdot \frac{2}{\pi^2 \sqrt{R^2 - r^2}} 2\pi r \, dr = \frac{4}{\pi} \int_0^R \frac{r^3}{\sqrt{R^2 - r^2}} \, dr$$
$$\overset{(i)}{=} \frac{4}{\pi} \int_0^{\pi/2} \frac{R^3 \sin^3(\theta)}{R\sqrt{1 - \sin^2(\theta)}} R\cos(\theta) \, d\theta = \frac{4R^3}{\pi} \int_0^{\pi/2} \sin^3(\theta) \, d\theta = \frac{4R^3}{\pi} \cdot \frac{2}{3} = \frac{8R^3}{3\pi} = \frac{\pi^2}{24},$$

where at (i) we substituted $r = R\sin(\theta)$. Integrating equation $\star$ against ρ^* gives

$$\mathbb{E}\|X - Y\| = \tfrac{1}{2}\sigma^2 - \tfrac{1}{4}C.$$

Substituting everything into equation 12 and using $C = -\pi^2/4$ from Step 6:

$$S^* = 2\sigma^2 - 2\mathbb{E}\|X - Y\| + 1 = 2\sigma^2 - \sigma^2 + \tfrac{1}{2}C + 1 = \frac{\pi^2}{12} - \frac{\pi^2}{24} - \frac{\pi^2}{8} + 1 = 1 - \frac{\pi^2}{12} \approx 0.1775.$$

This completes the proof. □

The final version of our proof is essentially a more verbose version of Mattner (1992)'s argument. We include it for the convenience of the reader. While proving this theorem, we were not aware of the work of Mattner (1992) and Buja et al. (1994). Both works were pointed out to us by a MathOverflow user (we will give further details in the deanonymized version).

Corollary 6 (Distribution of point norms under ρ^*)**.** *If $X \sim \rho^*$, the radius $\|X\|$ has as rescaled Kumaraswamy (a=2, b=1/2) density*

$$f_R(r) = \frac{4r}{\pi\sqrt{(\pi/4)^2 - r^2}} \mathbf{1}(r < \pi/4).$$

Proof. The density $f_R(r) = 2\pi r \cdot \rho^*(r)$ is obtained by integrating ρ^* over the circle of radius r. This is the Kumaraswamy density

$$abx^{a-1}(1 - x^a)^{b-1}$$

for $a = 2, b = 1/2$ rescaled from $[0, 1)$ to $[0, \pi/4)$. □

D From the Continuum Minimizer to Finite Configurations

Theorem 1 identifies the continuum minimizer ρ^*. We now quantify how close the empirical distribution of a finite stress-minimizing configuration is to ρ^*, both for exact unit targets and for noisy targets. The natural yardstick is the *energy distance*, whose square coincides with the second-order remainder already computed in Step 3 of the proof of Theorem 1.

Notation and standing assumptions. Throughout, $X = (x_1, \ldots, x_n) \in \mathbb{R}^{2\times n}$ with $n \geq 2$, we write $m = n(n-1)$ for the number of ordered off-diagonal pairs. The finite MDS stress of the Problem Formulation is

$$S_n(X) \;=\; \frac{1}{m} \sum_{i \neq j} \big(\|x_i - x_j\| - 1\big)^2, \tag{17}$$

a sum over ordered pairs $i \neq j$ only: there are no self-distances, equivalently the diagonal target distances are set to zero. Given a symmetric perturbation $\varepsilon_{ij} = \varepsilon_{ji}$ of the unit targets, $1 + \varepsilon_{ij}$, the perturbed stress and the root-mean-square perturbation are

$$S_n^\varepsilon(X) \;=\; \frac{1}{m} \sum_{i \neq j} \big(\|x_i - x_j\| - 1 - \varepsilon_{ij}\big)^2, \qquad \delta_n^2 \;=\; \frac{1}{m} \sum_{i \neq j} \varepsilon_{ij}^2. \tag{18}$$

Since S_n depends only on the pairwise differences $x_i - x_j$, it is invariant under a common translation of all points, so we restrict to *centered* configurations, i.e. the linear subspace $V = \{X : \sum_i x_i = 0\} \cong \mathbb{R}^{2(n-1)}$. On V both S_n and S_n^ε are *continuous* and *coercive*, meaning the value tends to $+\infty$ as $\|X\| \to \infty$ on V. A continuous coercive function on a finite-dimensional space attains its minimum,

since its sublevel sets are closed (by continuity) and bounded (by coercivity), hence compact, so the Weierstrass theorem applies. We fix global minimizers

$$X_n \in \operatorname*{argmin}_X S_n(X), \qquad X_n^\varepsilon \in \operatorname*{argmin}_X S_n^\varepsilon(X), \qquad \alpha_n \;=\; S_n(X_n) \;=\; \min_X S_n(X). \tag{19}$$

To a configuration X we associate the *centered empirical measure* $\mu_X = \frac{1}{n}\sum_{i=1}^n \delta_{x_i-\bar{x}}$ with $\bar{x} = \frac{1}{n}\sum_i x_i$; centering changes neither pairwise distances nor the stress.

D.1 The energy distance and the stability identity

Definition 7 (Energy distance)**.** For probability measures μ, ν on $\mathbb{R}^2$ with finite first moment, the *squared energy distance* is

$$D_E^2(\mu,\nu) \;=\; 2\iint \|x-y\|\, d\mu(x)\, d\nu(y) - \iint \|x-x'\|\, d\mu(x)\, d\mu(x') - \iint \|y-y'\|\, d\nu(y)\, d\nu(y'). \tag{20}$$

Because the Euclidean-distance kernel $\|x-y\|$ is strictly conditionally negative definite (Schoenberg, 1938), one has $D_E^2(\mu,\nu) \geq 0$ with equality if and only if $\mu = \nu$. Thus, D_E is a genuine metric on distributions (Székely & Rizzo, 2013; Sejdinovic et al., 2013). The first finite moment is no real restriction here. Both μ_X and ρ^* are compactly supported, and more generally any measure of finite stress has finite second moment (Step 1), hence finite first moment.

The next lemma is the quantitative counterpart of the strict convexity established in Step 3 of the proof of Theorem 1. It upgrades "$S[\eta] > 0$" to an exact identity with D_E^2 as the leading term.

Lemma 8 (Quadratic stability)**.** *Let*

$$g(x) = \|x\|^2 - 2\int \|x-y\|\, d\rho^*(y) + \frac{\pi^2}{8}$$

be the KKT slack from the proof of Theorem 1 (Steps 7–8), so that $g \equiv 0$ on $B(0,\pi/4)$ and $g \geq 0$ on $\mathbb{R}^2$. Then for every centered probability measure μ with finite second moment (equivalently $S[\mu] < \infty$, cf. Step 1),

$$S[\mu] - S^* \;=\; D_E^2(\mu,\rho^*) \;+\; \int g\, d\mu \;\geq\; D_E^2(\mu,\rho^*), \tag{21}$$

with equality whenever μ is supported in the closed disk $\overline{B}(0,\pi/4)$.

Proof. Write $\eta = \mu - \rho^*$. Both μ and ρ^* are centered probability measures, so

$$\int d\eta = 0, \qquad \int x\, d\eta(x) = 0. \tag{22}$$

Since S in equation 2 is a quadratic form, write

$$B(\alpha,\beta) = \iint K(x,y)\, d\alpha(x)\, d\beta(y)$$

for the symmetric bilinear form with kernel $K(x,y) = (\|x-y\|-1)^2$, so that $S[\rho] = \frac{1}{2}B(\rho,\rho)$. Expanding $S[\rho^*+\eta]$ and using the symmetry $B(\rho^*,\eta) = B(\eta,\rho^*)$,

$$S[\mu] \;=\; \tfrac{1}{2}B(\rho^*,\rho^*) + B(\rho^*,\eta) + \tfrac{1}{2}B(\eta,\eta) \;=\; S[\rho^*] + B(\rho^*,\eta) + S_2[\eta], \tag{23}$$

$$\text{with } S_2[\eta] := \tfrac{1}{2}B(\eta,\eta) = \tfrac{1}{2}\iint K(x,y)\, d\eta(x)\, d\eta(y). \tag{24}$$

To evaluate the linear coefficient $B(\rho^*, \eta)$, integrate the kernel against ρ^* first. Using $\int x\, d\rho^*(x) = 0$ and $\int \|x\|^2\, d\rho^*(x) = \sigma_*^2$ (with $\sigma_*^2 = \pi^2/24$ from Step 8),

$$\begin{aligned}\int K(x,y)\, d\rho^*(x) &= \int \big(\|x-y\|^2 - 2\|x-y\| + 1\big)\, d\rho^*(x) \\ &= \big(\|y\|^2 + \sigma_*^2\big) - 2\int \|x-y\|\, d\rho^*(x) + 1 \\ &= \underbrace{\|y\|^2 - 2\int \|x-y\|\, d\rho^*(x)}_{= g(y) - \pi^2/8} + (\sigma_*^2 + 1),\end{aligned}$$

where the underbraced term is the first variation $\delta S/\delta\rho$ from Step 2 evaluated at ρ^*, equal to $g(y) - \pi^2/8$ by the definition of g. Therefore

$$B(\rho^*, \eta) = \int \big(g(y) - \tfrac{\pi^2}{8}\big)\, d\eta(y) + (\sigma_*^2 + 1) \int d\eta(y),$$

and since $\int d\eta = 0$ the configuration-independent constant $\sigma_*^2 + 1$ drops out, so equation 23 becomes

$$S[\mu] \;=\; S[\rho^*] + \int \Big(g(x) - \tfrac{\pi^2}{8}\Big)\, d\eta(x) + S_2[\eta]. \tag{25}$$

Linear term. Since $\int d\eta = 0$, the constant $\pi^2/8$ in equation 25 integrates to zero, leaving $\int \big(g - \frac{\pi^2}{8}\big)\, d\eta = \int g\, d\eta$. Because $g \equiv 0$ on $\operatorname{supp}\rho^*$ we have $\int g\, d\rho^* = 0$, hence $\int g\, d\eta = \int g\, d\mu$; and $\int g\, d\mu \geq 0$ because $g \geq 0$ everywhere (Step 7) and μ is a nonnegative measure.

Quadratic term. Expand $K(x,y) = \|x-y\|^2 - 2\|x-y\| + 1$. The identity $\|x-y\|^2 = \|x\|^2 - 2x^\top y + \|y\|^2$ together with the moment conditions equation 22 makes the $\|x-y\|^2$ and the constant 1 integrate to zero against $\eta \otimes \eta$ (exactly as at steps (a)–(b) of Step 3), leaving

$$S_2[\eta] \;=\; -\iint \|x-y\|\, d\eta(x)\, d\eta(y) \;=\; D_E^2(\mu, \rho^*),$$

where the second equality is the bilinear expansion of $\eta = \mu - \rho^*$, matching equation 20.

Substituting both terms into equation 25 yields the identity equation 21; dropping $\int g\, d\mu \geq 0$ gives the inequality. If $\operatorname{supp}\mu \subseteq \overline{B}(0, \pi/4)$ then $\int g\, d\mu = 0$ and equality holds. □

D.2 Exact targets: finite minimizers converge to ρ^*

We first record how the finite stress equation 17 sits inside the continuum functional equation 2. Evaluating S at the empirical measure counts all n^2 ordered pairs, and the n diagonal pairs each contribute $(\|x_i - x_i\| - 1)^2 = 1$:

$$S[\mu_X] \;=\; \frac{1}{2n^2}\sum_{i,j}\big(\|x_i - x_j\| - 1\big)^2 \;=\; \frac{1}{2n^2}\Big[\, m\, S_n(X) + n\,\Big] \;=\; \frac{n-1}{2n}\, S_n(X) + \frac{1}{2n}. \tag{26}$$

The additive $\frac{1}{2n}$ is the self-pair contribution. With the diagonal targets set to zero it is absent from S_n, but the atomic measure μ_X places mass $1/n^2$ on each diagonal pair of $\mu_X \otimes \mu_X$. Crucially this term is independent of X, so it neither affects which configuration minimizes nor survives in the leading order of the bounds below. Since the scalar map $t \mapsto \frac{n-1}{2n}t + \frac{1}{2n}$ is strictly increasing, equation 26 gives $S[\mu_X] \leq S[\mu_{X'}]$ if and only if $S_n(X) \leq S_n(X')$ for any two configurations X, X'; hence S_n and $X \mapsto S[\mu_X]$ share the same minimizers, and in particular the finite minimizer X_n of S_n also minimizes $X \mapsto S[\mu_X]$ over all n-point configurations.

Proposition 9 (Finite minimizers under exact targets)**.** *For $a_n = S[\mu_{X_n}] - S^* \geq 0$ we have*

$$0 \;\leq\; a_n \;\leq\; \frac{1 - 2S^*}{2n} \;=\; \frac{\pi^2}{24n}, \tag{27}$$

and hence the empirical measure μ_{X_n} of the finite minimizer deviates from the continuum minimizer ρ^* by

$$D_E(\mu_{X_n}, \rho^*) \;\leq\; \frac{\pi}{\sqrt{24\, n}}. \tag{28}$$

Moreover, the minimal off-diagonal stress is squeezed against the continuum optimizer,

$$1 - \frac{n}{n-1}\frac{\pi^2}{12} = 2S^* - \frac{1-2S^*}{n-1} \;\le\; \alpha_n \;\le\; 2S^* = 1 - \frac{\pi^2}{12}. \tag{29}$$

Proof. Let $Y_1, \dots, Y_n \overset{\text{i.i.d.}}{\sim} \rho^*$. Since the summands of $S_n(X)$ equation 17 are identically distributed with

$$\mathbb{E}(\|Y_i - Y_j\| - 1)^2 = 2S[\rho^*] = 2S^* \quad \text{for } i \neq j,$$

we have

$$\mathbb{E}\, S_n(Y) \;=\; 2S^*. \tag{30}$$

As a result, there must be some configuration that does not exceed this expectation, so that

$$\alpha_n \;=\; \min_X S_n(X) \;\le\; \mathbb{E}\, S_n(Y) \;=\; 2S^*, \tag{31}$$

which is the upper bound in equation 29. Applying the affine relation equation 26 and using that X_n minimizes $S[\mu_X]$,

$$S[\mu_{X_n}] \;=\; \min_X S[\mu_X] \;\le\; \mathbb{E}\, S[\mu_Y] \;=\; \frac{n-1}{2n}\,\mathbb{E}S_n(Y) + \frac{1}{2n} \;=\; \frac{n-1}{n}\,S^* + \frac{1}{2n}.$$

Subtracting S^* gives

$$a_n \le \frac{1-2S^*}{2n} = \frac{\pi^2}{24n},$$

and $a_n \ge 0$ because ρ^* is the global minimizer of S over all probability measures (Theorem 1), so $S[\mu_{X_n}] \ge S^*$. This is equation 27. Lemma 8 then yields

$$D_E^2(\mu_{X_n}, \rho^*) \le S[\mu_{X_n}] - S^* = a_n \le \frac{\pi^2}{24n},$$

which is equation 28. Finally, $a_n \ge 0$ combined with equation 26 reads

$$\frac{n-1}{2n}\alpha_n + \frac{1}{2n} \ge S^*,$$

which we can rearrange to the lower bound in equation 29. □

D.3 Noisy targets: a deterministic perturbation bound

The passage from exact to noisy targets rests on a single deterministic inequality, the reverse triangle inequality in the space of pairwise residuals.

The statement holds for arbitrary target distances, so we phrase it that way. For a symmetric off-diagonal dissimilarity matrix $D = (d_{ij})_{i \neq j}$ write

$$S_D(X) \;=\; \frac{1}{m}\sum_{i \neq j}\big(\|x_i - x_j\| - d_{ij}\big)^2, \qquad \|D - D'\| \;=\; \Big(\frac{1}{m}\sum_{i \neq j}(d_{ij} - d'_{ij})^2\Big)^{1/2},$$

so that $S_n = S_{\mathbf{1}}$ and $S_n^\varepsilon = S_{\mathbf{1}+\varepsilon}$ with $\|(\mathbf{1}+\varepsilon) - \mathbf{1}\| = \delta_n$.

Lemma 10 (Reverse triangle inequality for the stress)**.** *Let D, D' be dissimilarity matrices.*

(i) For every configuration X,

$$\Big|\sqrt{S_D(X)} - \sqrt{S_{D'}(X)}\Big| \;\le\; \|D - D'\|. \tag{32}$$

(ii) If X_D and $X_{D'}$ minimize S_D and $S_{D'}$, then

$$S_D(X_{D'}) - S_D(X_D) \;\le\; 4\,\|D - D'\|\sqrt{S_D(X_D)} + 4\,\|D - D'\|^2. \tag{33}$$

The square root of a raw stress is the Euclidean norm of its residual vector (Borg & Groenen, 2005), which makes *(i)* an instance of the reverse triangle inequality. Related perturbation bounds for stress minimization under noise, under stronger assumptions and by a different argument, appear in (Arias-Castro & Vishwanath, 2023).

Proof. Fix X and collect the residuals into vectors in $\mathbb{R}^m$,

$$a = \tfrac{1}{\sqrt{m}}\big(\|x_i - x_j\| - d_{ij}\big)_{i\neq j}, \qquad a' = \tfrac{1}{\sqrt{m}}\big(\|x_i - x_j\| - d'_{ij}\big)_{i\neq j},$$

so that

$$\|a\|_2 = \sqrt{S_D(X)}, \quad \|a'\|_2 = \sqrt{S_{D'}(X)},$$

and

$$\|a - a'\|_2 = \|D - D'\|.$$

The triangle inequality gives $\|a\|_2 \le \|a'\|_2 + \|a - a'\|_2$ and, interchanging a and a', $\|a'\|_2 \le \|a\|_2 + \|a - a'\|_2$, hence

$$\big|\, \|a\|_2 - \|a'\|_2 \,\big| \le \|a - a'\|_2 = \|D - D'\|,$$

which is equation 32.

For *(ii)*, apply equation 32 at $X_{D'}$, then optimality of $X_{D'}$ for $S_{D'}$, then equation 32 at X_D,

$$\sqrt{S_D(X_{D'})} \;\le\; \sqrt{S_{D'}(X_{D'})}+\|D-D'\| \;\le\; \sqrt{S_{D'}(X_D)}+\|D-D'\| \;\le\; \sqrt{S_D(X_D)}+2\,\|D-D'\|.$$

The term $2\|D - D'\|$ arises because the perturbation is used twice. Squaring gives equation 33. □

Taking $D = \mathbf{1}$ and $D' = \mathbf{1} + \varepsilon$, so that $\|D - D'\| = \delta_n$, $X_D = X_n$, $X_{D'} = X_n^\varepsilon$, and $\alpha_n = S_n(X_n)$ as in equation 19, inequality equation 33 reads

$$S_n(X_n^\varepsilon) - \alpha_n \;\le\; 4\,\delta_n\sqrt{\alpha_n} + 4\,\delta_n^2. \tag{34}$$

Corollary 11 (Finite minimizers under noisy targets)**.** *The minimizer X_n^ε of the perturbed stress satisfies*

$$D_E^2(\mu_{X_n^\varepsilon}, \rho^*) \;\le\; \frac{\pi^2}{24n} \;+\; 2\sqrt{2S^*}\,\delta_n \;+\; 2\,\delta_n^2 \;\approx\; \frac{0.411}{n} + 0.843\,\delta_n + 2\,\delta_n^2. \tag{35}$$

In particular, perturbed finite minimizers remain close to ρ^* whenever both n^{-1} and the RMS perturbation δ_n are small; the worst-case dependence on the noise is $D_E = O(\sqrt{\delta_n})$.

Proof. By Lemma 8, adding and subtracting the exact-target value $S[\mu_{X_n}]$,

$$D_E^2(\mu_{X_n^\varepsilon}, \rho^*) \;\le\; S[\mu_{X_n^\varepsilon}] - S^* \;=\; \big(S[\mu_{X_n^\varepsilon}] - S[\mu_{X_n}]\big) + \big(S[\mu_{X_n}] - S^*\big).$$

The second bracket is $S[\mu_{X_n}] - S^* = a_n \le \frac{\pi^2}{24n}$ by equation 27. In the first bracket, we use the affine relation equation 26,

$$S[\mu_{X_n^\varepsilon}]-S[\mu_{X_n}] \;=\; \frac{n-1}{2n}\big(S_n(X_n^\varepsilon)-S_n(X_n)\big) \;=\; \frac{n-1}{2n}\big(S_n(X_n^\varepsilon)-\alpha_n\big) \;\le\; \frac{n-1}{2n}\big(4\delta_n\sqrt{\alpha_n}+4\delta_n^2\big),$$

using equation 34. Bounding $\frac{n-1}{n} \le 1$ and $\sqrt{\alpha_n} \le \sqrt{2S^*}$ from equation 29 gives equation 35, with numerical constants from $\pi^2/24 \approx 0.411$ and $2\sqrt{2S^*} = 2\sqrt{1 - \pi^2/12} \approx 0.843$. □

Remark 12 (Evaluating D_E numerically). The quantity most aligned with the theory is $D_E(\mu_n, \rho^*)$, computable from equation 20 as

$$D_E^2(\mu_n, \rho^*) \;=\; \frac{2}{n}\sum_{i=1}^{n} \mathbb{E}_{Y\sim\rho^*}\|x_i - Y\| \;-\; \frac{1}{n^2}\sum_{i,j}\|x_i - x_j\| \;-\; \mathbb{E}_{Y,Y'\sim\rho^*}\|Y - Y'\|. \tag{36}$$

The self-interaction constant is $\mathbb{E}_{Y,Y'\sim\rho^*}\|Y - Y'\| = \frac{\pi^2}{12} \approx 0.822$ (Step 8). For a point inside the support, $\|x\| \le \pi/4$, the on-support identity equation $\star$ gives the closed form $\mathbb{E}_{Y\sim\rho^*}\|x - Y\| = \frac{1}{2}\big(\|x\|^2 + \frac{\pi^2}{8}\big)$

Remark 13. Note that if the mean m of the ε_{ij} over ij is zero, this δ_n is equal to the standard deviation of the perturbation values. If the mean is not zero, our bound still holds. But a potentially better bound could be obtained by attributing m of ε_{ij} to the equidistant part of the target distances and rescaling ρ^* and the constants in the bound by $1 + m$.