

End-to-End Deep Learning for Predicting Metric Space-Valued Outputs

Yidong Zhou*

*Department of Statistics
University of California, Davis
Davis, CA 95616, USA*

YDZHOU@UCDAVIS.EDU

Su I Iao*

*Department of Statistics
University of California, Davis
Davis, CA 95616, USA*

SIAO@UCDAVIS.EDU

Hans-Georg Müller

*Department of Statistics
University of California, Davis
Davis, CA 95616, USA*

HGMUELLER@UCDAVIS.EDU

Editor: Aryeh Kontorovich

Abstract

Many modern applications involve predicting structured, non-Euclidean outputs such as probability distributions, networks, and symmetric positive-definite matrices. These outputs are naturally modeled as elements of general metric spaces, where classical regression techniques that rely on vector space structure no longer apply. We introduce E2M (End-to-End Metric regression), a deep learning framework for predicting metric space-valued outputs. E2M performs prediction via weighted Fréchet means over training outputs, where the weights are learned by a neural network conditioned on the input. This construction provides a principled mechanism for geometry-aware prediction that avoids surrogate embeddings and restrictive parametric assumptions, while fully preserving the intrinsic geometry of the output space. We establish theoretical guarantees, including a universal approximation theorem that characterizes the expressive capacity of the model and a convergence analysis of the entropy-regularized training objective. Through extensive simulations involving probability distributions, networks, and symmetric positive-definite matrices, we show that E2M consistently achieves state-of-the-art performance, with its advantages becoming more pronounced at larger sample sizes. Applications to human mortality distributions and New York City taxi networks further demonstrate the flexibility and practical utility of this framework.

Keywords: end-to-end models, Fréchet mean, neural networks, non-Euclidean outputs, Wasserstein space

1. Introduction

The rapid growth of complex, structured data across science and engineering increasingly challenges traditional learning paradigms and demands fundamentally new modeling tools.

*. The first two authors contributed equally to this work.

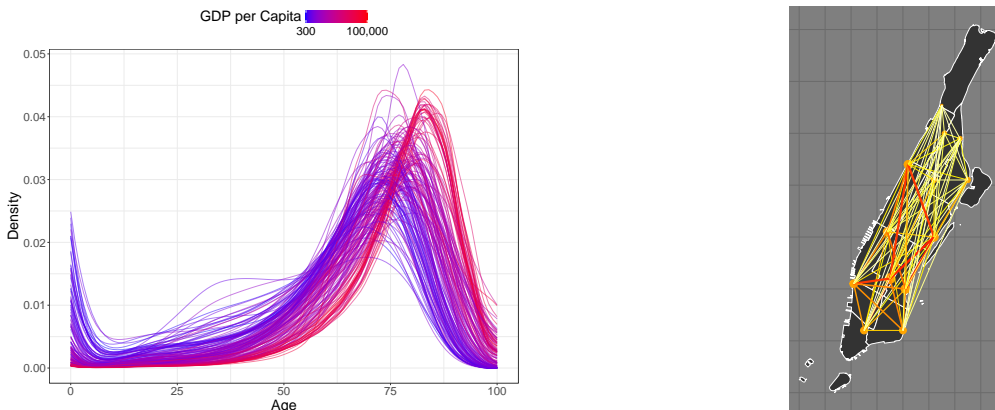


Figure 1: Motivating examples of non-Euclidean outputs. Left: age-at-death densities for 162 countries in 2015, colored by GDP per capita. Right: daily transportation network in Manhattan on January 1, 2018.

In fields such as neuroscience (Dryden et al., 2009), social science (Li et al., 2023), and genomics (Kapli et al., 2020), observations are increasingly recorded as non-Euclidean entities, which have been referred to as *random objects* (Müller, 2016). Examples include functional data (Wang et al., 2016), networks (Zhou and Müller, 2022), trees (Nye et al., 2017), probability distributions (Petersen et al., 2022), and data residing in manifolds such as symmetric positive-definite (SPD) matrices (Huang and Van Gool, 2017). These data types can be modeled as elements of general metric spaces, where the lack of algebraic operations such as addition, subtraction, or scalar multiplication renders standard statistical and machine learning techniques inapplicable.

To motivate the problem, Figure 1 illustrates two representative examples. The left panel shows age-at-death densities for 162 countries in 2015, with curves colored by GDP per capita. Here, the regression problem of interest is to model how the age-at-death distribution varies with demographic, economic, and environmental indicators, enabling comparisons across populations with different socioeconomic conditions. The right panel displays a daily traffic network in Manhattan on January 1, 2018, constructed from New York City yellow taxi trip data. In this setting, the goal is to predict the structure of daily transportation networks from predictors such as weather conditions, calendar effects, and aggregated trip statistics. Both examples highlight outputs that are complex objects such as probability distributions and networks, rather than vectors; such objects are inherently non-Euclidean.

Inspired by these applications, we study supervised learning when the input is a vector $X \in \mathbb{R}^p$ and the output Y resides in a general metric space (Ω, d) . While the proposed methodology applies to general metric spaces, the theoretical results developed in Section 5 require additional assumptions for the space (Ω, d) , including boundedness and for some results that it is a Hadamard space. These assumptions are used to establish continuity, Lipschitz properties, and optimization guarantees. Learning in this setting is fundamentally more challenging than for vector-valued outputs due to the lack of vector space structure in Ω . Existing solutions often rely on Euclidean embeddings (Faraway, 2014; Zhang et al.,

2024; Iao et al., 2025) or restrictive model assumptions (Hein, 2009; Petersen and Müller, 2019; Song and Han, 2023), which either distort the geometry of the output space or lack robustness in real-world scenarios. These limitations highlight the need for principled intrinsic methods that operate directly on non-Euclidean data, preserving the structural characteristics of the outputs while avoiding oversimplifying assumptions.

In this paper, we propose **E2M** (End-to-End Metric regression), a deep learning framework for regression with outputs in general metric spaces. E2M revisits the classical idea of regression as a weighted average over training outputs, adapting it to metric spaces using the *weighted Fréchet mean* (Fréchet, 1948). The weighted Fréchet mean minimizes a weighted sum of squared distances and remains well-defined without requiring algebraic operations within the output space. In E2M, a neural network conditioned on the input generates weights that encode the relevance of each training output in the weighted Fréchet mean computation. This approach preserves the intrinsic geometry of the output space while enabling flexible data-driven prediction.

Our main contributions are as follows:

- We propose E2M, the first end-to-end deep learning framework for supervised learning with metric space-valued outputs. E2M uses a neural network to learn sample-specific weights and constructs predictions via a weighted Fréchet mean. The key idea is to represent the regression function as a weighted average over training outputs. This formulation circumvents the lack of vector space operations in the output space and avoids reliance on surrogate embeddings. The model preserves output geometry and incorporates entropy regularization to control the sparsity and adaptivity of the weights.
- We establish theoretical guarantees for E2M, including a universal approximation theorem that characterizes the expressive power of the model and convergence results for the entropy-regularized training objective. Unlike standard regression settings, the analysis is complicated by the lack of linear structure in the output space. To address this challenge, we employ tools from metric geometry, including properties of Fréchet means and variance inequalities in Hadamard spaces (Sturm, 2003), to prove Lipschitz continuity and derive optimization guarantees.
- We demonstrate the effectiveness of E2M through extensive simulations and real-world applications involving probability distributions, networks, and SPD matrices. Across various scenarios, E2M consistently outperforms existing regression methods for non-Euclidean targets, demonstrating its flexibility, accuracy, and geometric robustness.

2. Related Work

Our work intersects several research areas, including neural networks for adaptive weighting, geometric deep learning and regression for metric space-valued outputs. We briefly review these areas and highlight how E2M conveys major advances over existing approaches.

Neural networks for weighted predictions. Neural networks often produce adaptive weights as part of their prediction pipeline. Attention mechanisms, introduced in Transformers (Vaswani et al., 2017), use neural networks to compute softmax-normalized weights

for aggregating context-aware representations. Similarly, geographically weighted neural networks (Hagenauer and Helbich, 2022) assign spatially varying weights to combine neural outputs based on input location. However, in these approaches, weights are used to aggregate intermediate features, and predictions are typically constrained to Euclidean spaces. In contrast, E2M directly generates predictions in general metric spaces using neural network-derived weights that determine a weighted Fréchet mean of training outputs.

Geometric deep learning. This line of work adapts deep learning to structured input domains such as graphs, manifolds, and sets (Bronstein et al., 2017). Methods like graph neural networks (Kipf and Welling, 2017) and manifold-based networks (Chakraborty et al., 2020) effectively handle complex input geometries for tasks like graph node classification or shape analysis. However, these techniques generally assume outputs reside in Euclidean spaces, with geometric structure considered only in input space. Our work addresses the complementary and much less explored problem of predicting outputs that reside in general metric spaces, which is relevant for many real-world applications, thus broadening the scope of geometry-aware modeling.

Regression models for metric space-valued outputs. A growing body of work extends learning and regression to cover metric space-valued outputs. Early methods involved embedding metric spaces into Euclidean frameworks (Faraway, 2014) or applying kernel-based techniques (Hein, 2009). More recently, Fréchet regression generalized linear and local linear regression to metric space-valued outputs (Petersen and Müller, 2019). Subsequent extensions of this framework include methods for sufficient dimension reduction (Ying and Yu, 2022; Zhang et al., 2024), single-index models (Bhattacharjee and Müller, 2023; Ghosal et al., 2023), principal component regression (Song and Han, 2023), and adaptations to tree-based approaches (Capitaine et al., 2024; Qiu et al., 2024; Bulté and Sørensen, 2024; Zhou et al., 2025). Related learning-theoretic work studies Bayes consistency in general metric spaces, including universal Bayes consistency for nearest-neighbor-type classification rules under metric inputs (Hanneke et al., 2021) and Bayes-consistent prediction under general metric losses using data-supported, Fréchet-type aggregation rules (Cohen and Kontorovich, 2022). A recent method (Iao et al., 2025) incorporates neural networks into Fréchet regression in a three-step approach, where neural networks are used to model the relationship between the Euclidean input and the low-dimensional manifold representation of the metric space-valued output. While effective, this approach relies on a low-dimensional manifold assumption, which may not hold in practice. In contrast, E2M provides a fully end-to-end learning framework, where both feature extraction and predictive weighting are jointly optimized via backpropagation, without restrictive assumptions on the intrinsic dimensionality of the output space.

3. Preliminaries

Let (Ω, d) be a metric space and consider a random object Y taking values in Ω . The classical concept of expectation in Euclidean space extends naturally to this setting via the Fréchet mean (Fréchet, 1948), defined by

$$E_{\oplus}[Y] = \arg \min_{y \in \Omega} E[d^2(y, Y)],$$

where the existence and uniqueness of the minimizer depend on the geometry of the underlying metric space and are guaranteed in Hadamard spaces (Sturm, 2003); see Definition 3.

To illustrate the scope of E2M, we present several representative examples of metric spaces that arise frequently in modern applications. These examples are used throughout our simulations and empirical analyses to demonstrate the generality and flexibility of the proposed framework.

Example 1 (One-dimensional probability distributions) Consider the Wasserstein space $(\mathcal{W}, d_{\mathcal{W}})$ (Panaretos and Zemel, 2020) of one-dimensional probability distributions with finite second moments, equipped with the 2-Wasserstein metric $d_{\mathcal{W}}$. The 2-Wasserstein metric between two distributions μ_1 and μ_2 is

$$d_{\mathcal{W}}^2(\mu_1, \mu_2) = \int_0^1 \{F_{\mu_1}^{-1}(p) - F_{\mu_2}^{-1}(p)\}^2 dp,$$

where $F_{\mu_1}^{-1}$ and $F_{\mu_2}^{-1}$ are the quantile functions of μ_1 and μ_2 , respectively. The Wasserstein space has attracted considerable attention across statistics and data science, as the Wasserstein metric provides a meaningful similarity measure between probability distributions that reflects the geometry of the underlying sample space (Villani, 2003). It has been widely used in modern applications, including Wasserstein generative adversarial networks (Arjovsky et al., 2017) and Wasserstein autoencoders (Tolstikhin et al., 2018), where it enables more stable training and better capture of distributional structures compared to classical divergence-based approaches.

Example 2 (Networks) Consider the space of simple, undirected, weighted networks with a fixed number of nodes and bounded edge weights. Each network can be represented uniquely by its graph Laplacian. The space of graph Laplacians equipped with the Frobenius metric can thus be used to characterize the space of networks (Kolaczyk and Csárdi, 2020; Severn et al., 2022; Zhou and Müller, 2022). Graph Laplacians have been widely used in both spectral methods and modern approaches to network representation and learning. For example, the graph convolutional network (Kipf and Welling, 2017) builds convolutional operations directly from the graph Laplacian, highlighting its role in capturing the intrinsic geometry of network data.

Example 3 (Symmetric positive-definite matrices) Consider the space of $l \times l$ symmetric positive-definite (SPD) matrices, denoted Sym_l^+ , with important examples including covariance and correlation matrices. Various metrics endow Sym_l^+ with rich geometric structure, including the Frobenius metric, the affine-invariant metric (Pennec et al., 2006), the power metric (Dryden et al., 2009), the Log-Cholesky metric (Lin, 2019), and the Bures–Wasserstein (BW) metric (Bhatia et al., 2019). The non-Euclidean geometry of Sym_l^+ plays a crucial role in machine learning and signal processing and recent methods have incorporated its Riemannian manifold structure directly into learning frameworks. A prominent example is SPDNet (Huang and Van Gool, 2017), a deep neural network where intermediate layers preserve the manifold structure through bilinear and eigenvalue operations and the final logarithm mapping layer projects SPD matrices into Euclidean features for downstream tasks such as classification.

In the presence of predictors $X \in \mathbb{R}^p$, one can consider conditional Fréchet means (Petersen and Müller, 2019):

$$E_{\oplus}(Y|X = x) = \arg \min_{y \in \Omega} E[d^2(y, Y)|X = x],$$

where the expectation is taken with respect to the conditional distribution of Y given X . This definition generalizes the classical conditional expectation, which is recovered when $\Omega = \mathbb{R}$ and d is the Euclidean distance. A more detailed discussion of the relationship between classical and Fréchet means is provided in Appendix A. As in classical regression, the conditional Fréchet mean $m(x) = E_{\oplus}(Y|X = x)$ serves as the target for regression with metric space-valued outputs.

A wide range of classical regression methods estimate the regression function by expressing predictions as weighted averages of the observed outputs. Notable examples include Nadaraya–Watson kernel regression (Nadaraya, 1964; Watson, 1964), k -nearest neighbors regression (Altman, 1992), inverse distance weighting (Shepard, 1968), linear regression, and local linear regression (Fan and Gijbels, 1996). Given training samples $\{(X_i, Y_i)\}_{i=1}^n$ and an input x , these methods assign a weight $w_i(x)$ to each training pair (X_i, Y_i) based on the proximity between x and X_i . Specifically, when $Y_i \in \mathbb{R}$, the prediction can be interpreted as a weighted average:

$$\hat{m}(x) = \sum_{i=1}^n w_i(x) Y_i = \arg \min_{y \in \mathbb{R}} \sum_{i=1}^n w_i(x) (Y_i - y)^2.$$

This variational form reveals that such methods estimate $m(x)$ by minimizing a weighted squared loss, where the weights encode relevance via predictor similarity. Further details on how classical linear and local linear regression also admit such characterizations are provided in Appendix B.

This weighted formulation generalizes naturally to metric space-valued outputs by replacing the squared Euclidean loss $(Y_i - y)^2$ with a squared distance induced by the metric d . The resulting prediction takes the form of a weighted Fréchet mean,

$$\hat{m}(x) = \arg \min_{y \in \Omega} \sum_{i=1}^n w_i(x) d^2(y, Y_i).$$

Weighted Fréchet means and related barycenter constructions have been extensively studied in the literature, including weighted Wasserstein barycenters (Agueh and Carlier, 2011; Le Gouic and Loubes, 2017) and Fréchet regression (Petersen and Müller, 2019). This generalization preserves the intuition of proximity-weighted averaging while accommodating outputs in structured or nonlinear spaces. A key advantage is that the learned weights are directly interpretable: each $w_i(x)$ quantifies the contribution of training sample Y_i to the prediction at input x . Concentrated weights highlight locally influential samples, whereas more uniform weights reflect global smoothing. In this way, the method provides insight into sample relevance that is difficult to obtain from models that compress training information into fixed parameters.

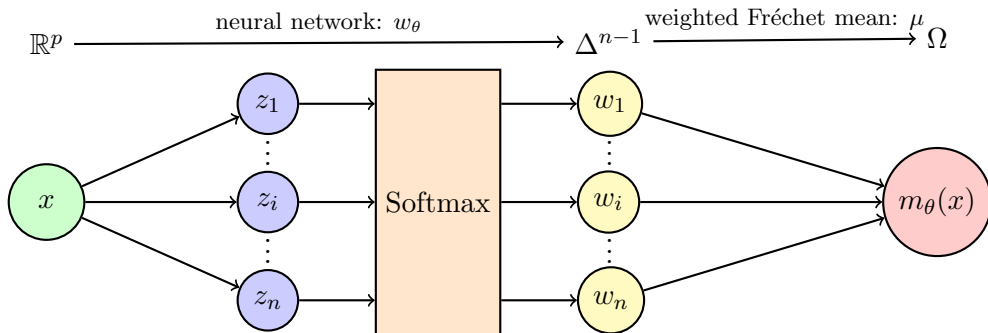


Figure 2: Schematic diagram for E2M. Here $m_\theta = \mu \circ w_\theta$, where w_θ is a neural network parameterized by θ and μ denotes the weighted Fréchet mean. The input $x \in \mathbb{R}^p$ is mapped to \mathbb{R}^n , passed through softmax to Δ^{n-1} via neural network w_θ and then mapped via the weighted Fréchet mean μ to the metric space Ω .

4. Methodology

4.1 E2M

Rather than specifying the weight function $w_i(x)$ through fixed kernels or distance rules, we propose to learn it directly from data. To this end, we introduce E2M, a regression framework for metric space-valued outputs, where a neural network is trained end-to-end to generate adaptive, task-specific weights that minimize prediction error in the target metric space. The architecture of the proposed model is illustrated in Figure 2.

The model is $m_\theta = \mu \circ w_\theta$, where the first component, $w_\theta : \mathbb{R}^p \mapsto \Delta^{n-1}$, is a fully connected neural network with multiple hidden layers using rectified linear unit (ReLU) activations, followed by a softmax output layer. Here θ represents the network parameters, and $\Delta^{n-1} = \{w \in \mathbb{R}^n : w_i \geq 0 \text{ for all } i, \sum_{i=1}^n w_i = 1\}$ is the $(n-1)$ -simplex. Given an input x , the network outputs a weight vector $w_\theta(x) \in \Delta^{n-1}$ that assigns relevance scores to each of the training samples. The second component, $\mu : \Delta^{n-1} \mapsto \Omega$, is a weighted Fréchet mean,

$$\mu(w) = \arg \min_{y \in \Omega} \sum_{i=1}^n w_i d^2(y, Y_i),$$

mapping the learned weights to a prediction in the metric space Ω , and w_i denotes the i th coordinate of the softmax output.

By learning w_θ end-to-end together with the weighting and prediction mechanism, E2M bypasses the need for hand-crafted feature engineering or local distance computations. To learn the optimal network parameters, we minimize the empirical loss

$$\frac{1}{n} \sum_{i=1}^n d^2(m_\theta(X_i), Y_i),$$

which measures the discrepancy between the predicted outputs and the observed outputs in the metric space Ω . The learned network implicitly captures statistical dependencies between the input and the output. This feedback mechanism allows the model to go beyond

Algorithm 1 E2M Training Procedure

Input: Training samples $\{(X_i, Y_i)\}_{i=1}^n$, entropy regularization parameter λ

Output: Trained neural network w_θ

- 1: Initialize parameters θ of the neural network w_θ
- 2: Optimize θ using the Adam optimizer with entropy-regularized empirical loss:

$$\mathcal{L}_n(\theta) = \frac{1}{n} \sum_{i=1}^n d^2(m_\theta(X_i), Y_i) + \lambda \frac{1}{n} \sum_{i=1}^n H(w_\theta(X_i)),$$

where $m_\theta = \mu \circ w_\theta$ and $\mu(w) = \arg \min_{y \in \Omega} \sum_{i=1}^n w_i d^2(y, Y_i)$

proximity-based heuristics and discover weighting schemes that reflect predictive relevance. In particular, E2M can assign low weights to samples that are geometrically close but uninformative or noisy with respect to the target.

4.2 Regularization

To enhance the stability and generalization of the learned model, E2M incorporates regularization techniques that directly act on the learned weight distribution. We primarily adopt entropy regularization, which provides a principled and interpretable mechanism to control the sharpness and dispersion of the weights. The empirical loss with entropy regularization is defined as

$$\mathcal{L}_n(\theta) = \frac{1}{n} \sum_{i=1}^n d^2(m_\theta(X_i), Y_i) + \lambda \frac{1}{n} \sum_{i=1}^n H(w_\theta(X_i)),$$

where $H(w) = -\sum_{i=1}^n w_i \log(w_i + \delta)$ denotes the entropy of the weight vector, $\lambda \in \mathbb{R}$ is a hyperparameter controlling the strength of regularization, and δ is a small positive constant for numerical stability (set to 10^{-10} in the implementation).

Entropy regularization offers a simple yet effective way to control the behavior of the model through the concentration of weights. Positive values of λ encourage sharper, low-entropy distributions that prioritize a small subset of influential training samples—akin to local regression. In contrast, negative values of λ promote higher-entropy, more uniform weight distributions, leading to a global smoothing effect. This flexibility allows E2M to adapt to varying data structures and modeling needs. In practice, the regularization parameter λ is chosen through cross-validated grid search; see Appendix D for details. To further assess robustness, we conducted a sensitivity analysis, reported in Appendix E, which demonstrates that performance is stable over a range of values. The full training procedure is outlined in Algorithm 1.

5. Theory

We analyze the theoretical properties of the E2M framework using tools from metric geometry and optimization. We begin by proving a universal approximation theorem, which establishes the expressive capacity of the model. Next, we show that the weighted Fréchet mean is Lipschitz continuous with respect to the weight vector, using a variance inequality

from metric geometry (Sturm, 2003). This Lipschitz property is instrumental in analyzing the convergence of the training algorithm. Proofs of all lemmas and theorems are provided in Appendix C.

We consider the target mapping $m = \mu \circ w$, where $w : \mathbb{R}^p \mapsto \Delta^{n-1}$ is a continuous function and $\mu : \Delta^{n-1} \mapsto \Omega$ is a deterministic map corresponding to the weighted Fréchet mean. The learning task is to approximate w using a neural network w_θ , so that $m_\theta = \mu \circ w_\theta$ approximates m .

Assumption 1 *For every $w \in \Delta^{n-1}$, the weighted Fréchet mean $\mu(w)$ exists and is unique.*

This assumption guarantees that the map μ is well-defined and holds for the spaces described in Examples 1 and 2. For Example 3 it will depend on the specific metric that one adopts whether the assumption holds or not. It holds when equipping the space with any of the metrics mentioned except for the BW metric and even then it may be satisfied, depending on the specifics of the sample. For simulated data we found it to be still satisfied for most realizations. Under this condition, we apply the Berge Maximum Theorem (Aliprantis and Border, 2006, Theorem 17.31) to establish the continuity of the weighted Fréchet mean map μ .

Lemma 1 *Suppose Assumption 1 holds and (Ω, d) is compact. Then the weighted Fréchet mean map $\mu : \Delta^{n-1} \mapsto \Omega$ is continuous on Δ^{n-1} .*

This continuity enables the following universal approximation result.

Theorem 2 *Suppose Assumption 1 holds and (Ω, d) is compact. For any $\epsilon > 0$, there exists a neural network w_{θ^*} such that the function $m_{\theta^*} = \mu \circ w_{\theta^*}$ satisfies*

$$\sup_{\|x\| \leq 1} d(m_{\theta^*}(x), m(x)) < \epsilon.$$

If X is stochastically bounded, then for any $\delta > 0$ there exists a neural network w_{θ^} such that*

$$P(d(m_{\theta^*}(X), m(X)) < \epsilon) > 1 - \delta.$$

Next, we analyze the convergence behavior of the proposed architecture within the geometric framework of Hadamard spaces, which provide a natural setting for our analysis.

Definition 3 (Hadamard space) *A metric space (Ω, d) is called a Hadamard space if it is complete and if for each pair of points $\omega_1, \omega_2 \in \Omega$, there exists a point $\alpha \in \Omega$ satisfying:*

$$d^2(\beta, \alpha) \leq \frac{1}{2}d^2(\beta, \omega_1) + \frac{1}{2}d^2(\beta, \omega_2) - \frac{1}{4}d^2(\omega_1, \omega_2), \quad \text{for all } \beta \in \Omega.$$

Hadamard spaces, also known as globally non-positively curved spaces (Sturm, 2003), are uniquely geodesic and admit well-behaved notions of distance and convexity. In particular, the weighted Fréchet mean $\mu(w)$ is guaranteed to exist and be unique for any weight vector $w \in \Delta^{n-1}$ (Bacák, 2014a). Common examples of Hadamard spaces include Euclidean and Hilbert spaces, hyperbolic spaces and various other spaces frequently encountered in

Algorithm 2 Adam

Input: Initial parameter θ_0 , learning rate η , decay parameters $\beta_1, \beta_2 \in [0, 1]$, $\epsilon > 0$

- 1: Set $m_0 = 0, v_0 = 0$
 - 2: **for** $k = 1$ to T **do**
 - 3: Draw a minibatch size of b : $\{(X_j, Y_j)\}_{j=1}^b$
 - 4: Compute $g_k = \frac{1}{b} \sum_{j=1}^b \nabla \ell(\theta_k; (X_j, Y_j))$
 - 5: $m_k = \beta_1 m_{k-1} + (1 - \beta_1) g_k$
 - 6: $v_k = \beta_2 v_{k-1} + (1 - \beta_2) g_k^2$
 - 7: $\hat{m}_k = m_k / (1 - \beta_1^k)$
 - 8: $\hat{v}_k = v_k / (1 - \beta_2^k)$
 - 9: $\theta_k = \theta_{k-1} - \eta \hat{m}_k / (\sqrt{\hat{v}_k} + \epsilon)$
 - 10: **end for**
-

applications, including those discussed in Examples 1 and 2. In Example 3, the space of SPD matrices equipped with any of the metrics mentioned there except for the BW metric is Hadamard. These spaces have been widely studied in optimization (Bacák, 2014b), regression (Lin and Müller, 2021) and geometric deep learning (Ganea et al., 2018).

Remark 4 *The theoretical analysis assumes that the output space (Ω, d) is a Hadamard space. This condition is mainly required for theory, as it guarantees convexity and variance inequalities that yield Lipschitz continuity of the weighted Fréchet mean map and enable convergence analysis. Several experimental settings in this paper, such as the Wasserstein space for univariate distributions, SPD matrices with power metrics and networks with the Frobenius metric satisfy this property. To further examine the scope of the method, an additional simulation in Section 6 with SPD matrices under the BW metric, a positively curved space that is not Hadamard (Thanwerdas and Pennec, 2023), shows that E2M continues to perform effectively. Thus, while the Hadamard assumption is important for establishing theoretical guarantees, the method remains practically applicable in a broader class of metric spaces. From a theoretical perspective, extending the analysis to metric spaces where weighted Fréchet means may fail to be unique, or where the Fréchet functional is not geodesically convex, remains an important open problem. Developing guarantees for optimization and statistical behavior of E2M under such conditions is a promising direction for future research.*

We train the neural network w_θ using the Adam optimization algorithm (Kingma and Ba, 2015) to minimize the entropy-regularized empirical loss. The procedure is summarized in Algorithm 2. Let $\ell(\theta; (X, Y)) = d^2(m_\theta(X), Y) + \lambda H(w_\theta(X))$ denote the per-sample loss with entropy regularization and let $\mathcal{L}(\theta) = E[\ell(\theta; (X, Y))]$ be the corresponding population loss. While \mathcal{L} is convex in certain special cases (e.g., Examples 1–3), it is generally non-convex due to the nested minimization in the weighted Fréchet mean μ . To address this, we establish a Lipschitz bound for the weighted Fréchet mean map using the variance inequality (Sturm, 2003).

Lemma 5 *If (Ω, d) is a bounded Hadamard space, then the weighted Fréchet mean map $\mu : \Delta^{n-1} \rightarrow \Omega$ is Lipschitz continuous on Δ^{n-1} . Specifically, for any $w_1, w_2 \in \Delta^{n-1}$, we*

have

$$d(\mu(w_1), \mu(w_2)) \leq D\sqrt{n} \|w_1 - w_2\|_2,$$

where $D = \sup_{u,v \in \Omega} d(u, v)$ is the diameter of Ω .

Lemma 5 implies the Lipschitz continuity of the loss function $\ell(\theta; (X, Y))$, a property essential for convergence analysis, and is broadly useful for studying the stability of weighted Fréchet means with respect to weights.

Remark 6 (Scalability via anchors) *The Lipschitz constant in Lemma 5 contains a \sqrt{n} factor. While this may appear undesirable at first glance, note that n here refers to the number of anchor points used in computing the weighted Fréchet mean, not necessarily the overall sample size. For simplicity, the implementation uses all training outputs $\{Y_i\}_{i=1}^n$ as anchors, hence the notation. However, our framework does not require this choice. In large-scale settings, one can subsample a fixed set of anchors independent of the dataset size, in which case the Lipschitz constant becomes independent of the total sample size. This anchor-based strategy provides a natural extension for scalability and Section 6.4 demonstrates through large-scale experiments that it preserves predictive accuracy while substantially improving computational efficiency.*

Assumption 2 *We impose the following assumptions:*

- (i) **Lipschitz neural network:** *The neural network $w_\theta : \mathbb{R}^p \mapsto \Delta^{n-1}$ is L -Lipschitz continuous with respect to its parameters θ .*
- (ii) **Smoothness:** *The loss $\ell(\theta; (X, Y))$ is β -smooth with respect to θ , i.e., its gradient $\nabla_\theta \ell(\theta; (X, Y))$ is β -Lipschitz continuous.*
- (iii) **Bounded variance:** *The variance of the stochastic gradient is bounded, i.e., for some $\sigma^2 > 0$ it holds that $E[\|\nabla_\theta \ell(\theta; (X, Y)) - \nabla \mathcal{L}(\theta)\|_2^2] \leq \sigma^2$.*

These assumptions are standard in non-convex optimization settings (Zaheer et al., 2018; Chen et al., 2019). Assumption 2(i) is mild, as neural networks are Lipschitz continuous when weight matrices are bounded and the activation functions are Lipschitz, which holds for commonly used choices such as ReLU, Tanh or Sigmoid. In practice, techniques such as spectral normalization (Miyato et al., 2018), weight clipping (Arjovsky et al., 2017) and Lipschitz regularization (Gouk et al., 2021) are frequently employed to enforce Lipschitz continuity of w_θ .

Theorem 7 *Suppose (Ω, d) is a bounded Hadamard space and Assumption 2 holds. Let Algorithm 2 be run with mini-batch size b and hyperparameters satisfying $\eta \leq \epsilon/(2\beta)$ and $1 - \beta_2 \leq \epsilon^2/(16G^2)$, where $G = L\sqrt{n} (2D^2 + |\lambda|(|\log \delta| + 1))$ is the Lipschitz constant of the loss $\ell(\theta; (X, Y))$ with respect to θ . Then for an iterate θ_τ chosen uniformly at random from $\{\theta_1, \dots, \theta_T\}$, we have*

$$E [\|\nabla \mathcal{L}(\theta_\tau)\|_2^2] = O\left(\frac{1}{T} + \frac{1}{b}\right).$$

This result shows that the algorithm converges to a stationary point, with the $1/T$ term reflecting the effect of training duration and the $1/b$ term capturing the variance reduction from larger mini-batches.

6. Numerical Experiments

We evaluate the performance of E2M through comprehensive simulations involving three types of non-Euclidean outputs: probability distributions modeled in the Wasserstein space with the Wasserstein metric (Example 1), networks represented by graph Laplacians with the Frobenius metric (Example 2) and SPD matrices equipped either with the power metric with exponent $1/2$ or with the BW metric (Example 3). Each setting is tested across sample sizes $n = 500, 1000, 2000$, with 200 Monte Carlo replications per scenario. We compare the performance of E2M with that of deep Fréchet regression (DFR) (Iao et al., 2025), global Fréchet regression (GFR) (Petersen and Müller, 2019), sufficient dimension reduction (SDR) (Zhang et al., 2024), single-index Fréchet regression (IFR) (Bhattacharjee and Müller, 2023) and random forest weighted local Fréchet regression (RFWLFR) (Qiu et al., 2024). Note that SDR and IFR could not be applied to SPD outputs with the power metric because no compatible implementations are available. Moreover, current implementations of DFR, GFR, SDR, IFR and RFWLFR do not handle SPD outputs under the BW metric; extending them would require nontrivial adaptations that are specific to both the space and the metric. To further assess robustness, we include additional experiments evaluating performance in small sample regimes (Appendix F) and in settings where the true input-output relationship is strictly linear (Appendix G).

6.1 Implementation Details

The computation of the weighted Fréchet mean $\mu(w)$ depends on the geometry of the output space and is carried out using space-specific numerical routines. We summarize below the implementations used in our numerical experiments.

One-dimensional probability distributions. For one-dimensional probability distributions equipped with the 2-Wasserstein metric, the weighted Fréchet mean admits a closed-form expression in terms of quantile functions. Specifically, $\mu(w)$ is computed as the distribution whose quantile function is given by the weighted average of the empirical quantile functions of the training distributions. This computation is exact and does not require iterative optimization.

Networks. For network-valued outputs represented by graph Laplacian matrices and equipped with the Frobenius metric, the weighted Fréchet mean reduces to a weighted Euclidean average of the Laplacian matrices. In this case, $\mu(w)$ is computed directly by matrix summation and normalization; uniqueness of the Fréchet mean is guaranteed.

SPD matrices (power metric). For SPD matrices equipped with the power metric, the weighted Fréchet mean is computed via the power transform. Specifically, each matrix is first mapped to the power-transformed space, where the Fréchet mean reduces to a weighted Euclidean average. The resulting matrix is then mapped back using the inverse power transform. This computation is direct, avoids iterative optimization, and was numerically stable in all experiments.

SPD matrices (BW metric). For SPD matrices equipped with the BW metric, the underlying space is positively curved and the weighted Fréchet mean need not be unique in general. In our implementation, we compute $\mu(w)$ using a standard fixed-point iteration

for the BW barycenter (Álvarez-Esteban et al., 2016). The algorithm is initialized at the identity matrix and iteratively updates the current estimate via matrix square-root and inverse square-root operations obtained from eigendecomposition. The procedure returns one minimizer of the weighted Fréchet functional. Although multiple minimizers may exist in principle, we did not observe numerical instability or non-convergence in our experiments. The empirical loss is therefore well-defined with respect to the selected barycenter.

Across all settings, Fréchet mean computations are embedded within the training loop and are differentiable with respect to the weights w , enabling end-to-end optimization via backpropagation. The implementation details and code are publicly available at <https://github.com/SUIIA0/E2M>.

6.2 Experimental Setup

In both the numerical experiments and real-world data applications, E2M was trained for 2,000 epochs using mini-batches of size 32, a learning rate of 5×10^{-4} , and a dropout rate of 30%. Other hyperparameters, including the regularization strength, the number of hidden layers and the number of neurons per layer, were selected via grid search based on cross-validated empirical risk (see Appendix D for details). For each training run, 10% of the training data was held out for early stopping. Performance was evaluated via mean squared prediction error (MSPE) over 200 independent test points. For the q -th Monte Carlo run, with \hat{m}_q denoting the estimator and m the true regression function, the MSPE is

$$\text{MSPE}_q = \frac{1}{200} \sum_{i=1}^{200} d^2\{\hat{m}_q(X_i^{\text{test}}), m(X_i^{\text{test}})\},$$

where d is the metric for the corresponding metric space. The average performance over 200 Monte Carlo runs is quantified by

$$\text{AMSPE} = \frac{1}{200} \sum_{q=1}^{200} \text{MSPE}_q.$$

Distributions. We consider a regression setting where the output Y is a Gaussian distribution. The input is a vector $X \in \mathbb{R}^{12}$ with components generated as follows:

$$\begin{aligned} X_1 &\sim U(-1, 0), & X_2 &\sim U(-1, 0), & X_3 &\sim U(0, 1), & X_4 &\sim U(0, 1), \\ X_5 &\sim \text{Gamma}(2, 2), & X_6 &\sim \text{Gamma}(3, 2), & X_7 &\sim \text{Gamma}(4, 2), & X_8 &\sim \text{Gamma}(5, 2), \\ X_9 &\sim \text{Ber}(0.6), & X_{10} &\sim \text{Ber}(0.5), & X_{11} &\sim \text{Ber}(0.4), & X_{12} &\sim \text{Ber}(0.3). \end{aligned}$$

The mean η and standard deviation σ of the distributional output Y are generated conditional on the input, where $\eta \sim N(\mu(X), 0.5^2)$, $\sigma \sim \text{Gamma}(\theta(X)^2, \theta(X)^{-1})$ with

$$\begin{aligned} \mu(X) &= 2 + 2 \cos(\pi X_1)^2 + \sin(\pi X_2)^2 X_9 + \sqrt{X_5 X_6} (1 - X_9), \\ \theta(X) &= 1 + \cos(\pi X_2/2) + \sin(\pi X_3) X_{10} + \sqrt{X_6 X_7} (1 - X_{10})/3. \end{aligned}$$

To better reflect real-world settings where the underlying probability distributions are not directly observable, we simulate independent samples from each distribution. Specifically,

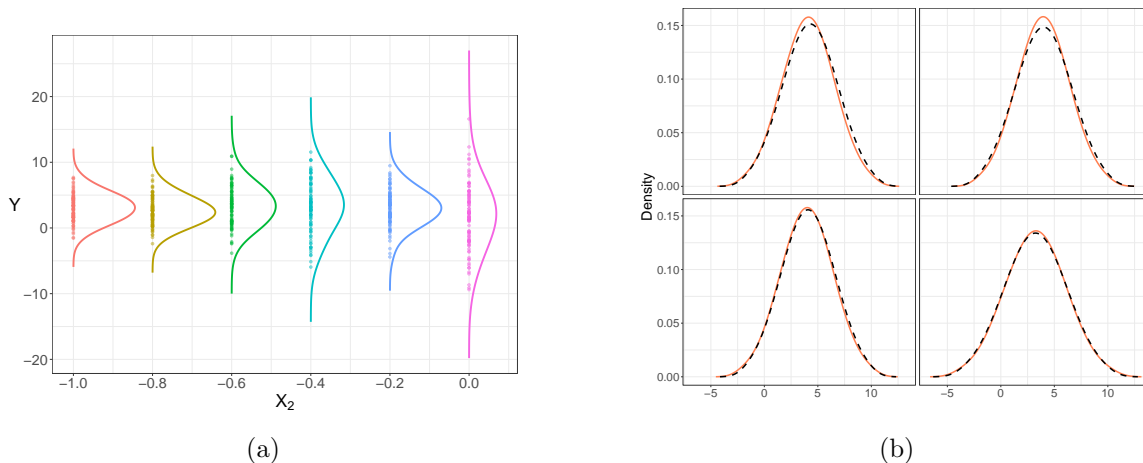


Figure 3: (a) Simulated distributional outputs as a function of X_2 , with all other covariates fixed at their mean (continuous variables) or median (discrete variables) values. (b) Fitted versus true densities in the distributional setting with $n = 500$. Four randomly selected sample units are shown for illustration. Coral solid curves denote the E2M fitted densities; the black dashed curves denote the corresponding true densities.

for each distribution Y_i , we generate 100 observations $\{y_{ij}\}_{j=1}^{100}$. E2M must then operate on a noisy version of Y_i , constructed from the empirical distribution of these samples; see also (Zhou and Müller, 2024). To provide additional insight into the data-generating mechanism, Figure 3 (a) visualizes simulated distributional outputs as a function of a representative covariate, X_2 , while holding all remaining covariates fixed at their mean values (for continuous variables) or median levels (for binary variables). This plot illustrates the dependence of mean and variance of the distributional output Y on X_2 implied by the model specification.

Networks. The output is a graph Laplacian derived from a weighted stochastic block model with two communities. Each network contains 10 nodes, equally divided into two blocks. The block connectivity structure is governed by probabilities $p_{11} = p_{22} = 0.5$ for within-community connections and $p_{12} = 0.2$ for between-community connections. The input is a vector $X \in \mathbb{R}^9$ with components generated as follows:

$$\begin{aligned} X_1 &\sim U(0, 1), & X_2 &\sim U(-1/2, 1/2), & X_3 &\sim U(1, 2), & X_4 &\sim N(0, 1), \\ X_5 &\sim N(0, 1), & X_6 &\sim N(5, 5), & X_7 &\sim \text{Ber}(0.4), & X_8 &\sim \text{Ber}(0.3), & X_9 &\sim \text{Ber}(0.6). \end{aligned}$$

For each edge, we assign a Beta-distributed weight with shape parameters depending on X and the block membership of the connected nodes. In Block 1, the shape parameters are $\alpha_1 = 2 \sin(\pi X_1) X_8 + \cos(\pi X_2)(1 - X_8)$ and $\beta_1 = 2X_4^2 X_7 + X_5^2(1 - X_7)$. In Block 2, the parameters are $\alpha_3 = \sin(\pi X_1) X_8 + 2 \cos(\pi X_2)(1 - X_8)$ and $\beta_3 = X_4^2 X_7 + 2X_5^2(1 - X_7)$. Between blocks, we set $\alpha_2 = 2 \sin(\pi X_1) X_8 + \cos(\pi X_2)(1 - X_8)$ and $\beta_2 = X_4^2 X_7 + 2X_5^2(1 - X_7)$. These weights are assembled into an adjacency matrix and the corresponding graph Laplacian serves as the output.

SPD matrices with the power metric. The third simulation generates SPD matrix outputs from a Wishart distribution, $Y \sim \mathcal{W}_l(\Sigma, df)$, where $l = 5$ is the dimension of the

matrices, $df = l + 1$ is the degrees of freedom, and Σ is the scale matrix with input-dependent diagonal entries. The input is a vector $X \in \mathbb{R}^{12}$ with components generated as follows:

$$\begin{aligned} X_1 &\sim U(0, 1), & X_2 &\sim U\left(-\frac{1}{2}, \frac{1}{2}\right), & X_3 &\sim U(1, 2), & X_4 &\sim \text{Gamma}(3, 2), \\ X_5 &\sim \text{Gamma}(4, 2), & X_6 &\sim \text{Gamma}(5, 2), & X_7 &\sim N(0, 1), & X_8 &\sim N(0, 1), \\ X_9 &\sim N(0, 1), & X_{10} &\sim \text{Ber}(0.4), & X_{11} &\sim \text{Ber}(0.5), & X_{12} &\sim \text{Ber}(0.6). \end{aligned}$$

The diagonal entries of the scale matrix Σ are generated conditional on the input, where

$$\begin{aligned} \Sigma_{11} &= \{\sin(\pi X_1)X_{10} + \cos(\pi X_2)(1 - X_{10})\}^2, & \Sigma_{22} &= \sin^2(\pi X_1) \cos^2(\pi X_2), \\ \Sigma_{33} &= \left\{ \frac{X_4}{X_5} \cdot \frac{1}{10} X_{11} + \sqrt{\frac{X_5}{X_4}} \cdot \frac{1}{10} (1 - X_{11}) \right\}^2, & \Sigma_{44} &= \frac{|X_7 X_8|}{25}, & \Sigma_{55} &= \frac{|X_9 / X_6|}{9}. \end{aligned}$$

Distances between SPD matrices are computed using the power metric with exponent $1/2$ (Dryden et al., 2009).

SPD matrices with the BW metric. To further investigate the applicability of E2M beyond the Hadamard setting, we conducted an additional experiment with SPD matrices equipped with the BW metric, a classical example of a positively curved space that is not Hadamard (Thanwerdas and Pennec, 2023). The BW distance between two SPD matrices $A, B \in \text{Sym}_l^+$ is given by

$$d_{\text{BW}}^2(A, B) = \text{Tr}(A) + \text{Tr}(B) - 2 \text{Tr}\left((A^{1/2} B A^{1/2})^{1/2}\right).$$

We demonstrate this setting with 2×2 SPD outputs, while noting that the same implementation readily extends to higher dimensions. For each sample, predictors $X = (X_1, \dots, X_5) \in \mathbb{R}^5$ were generated as

$$X_1 \sim U(0, 1), \quad X_2 \sim U(-0.5, 0.5), \quad X_3 \sim U(1, 2), \quad X_4 \sim \text{Ber}(0.6), \quad X_5 \sim \text{Ber}(0.5).$$

Each SPD output Y was drawn from a Wishart distribution with degrees of freedom $df = 3$ and scale matrix $\Sigma(X) = \text{diag}(\sigma_{11}^2, \sigma_{22}^2)$, where

$$\sigma_{11} = \sin(\pi X_1)X_4 + \cos(\pi X_2)(1 - X_4), \quad \sigma_{22} = \sin(\pi X_2) \cos(\pi X_3).$$

6.3 Discussion of the Simulation Results

Table 1 summarizes the predictive performance of E2M and baseline methods across all simulation settings. For distributional, network and SPD (power metric) outputs, E2M consistently achieves the lowest average prediction error among all competing methods, with its advantage becoming more pronounced at larger sample sizes. For SPD outputs with the BW metric, which correspond to a positively curved space outside the Hadamard class, E2M remains fully implementable and delivers low prediction error, whereas DFR, GFR, RFWLFR, SDR and IFR cannot currently handle this setting. These findings demonstrate that E2M effectively models complex nonlinear relationships between inputs and a wide range of non-Euclidean outputs, encompassing both Hadamard and non-Hadamard spaces.

Table 1: Average mean squared prediction errors (mean on first line, standard deviation in parentheses on second line) of E2M, deep Fréchet regression (DFR) (Iao et al., 2025), global Fréchet regression (GFR) (Petersen and Müller, 2019), sufficient dimension reduction (SDR) (Zhang et al., 2024), single index Fréchet regression (IFR) (Bhattacharjee and Müller, 2023) and random forest weighted local Fréchet regression (RFWLFR) (Qiu et al., 2024) for distribution, network and SPD matrix outputs. SDR and IFR were not included for SPD outputs with the power metric, due to the lack of available implementations, and none of the other competing methods currently support SPD outputs under the BW metric.

Output	n	E2M	DFR	GFR	RFWLFR	SDR	IFR
Distribution	500	0.562	0.869	0.766	0.927	0.753	0.929
		(0.120)	(0.125)	(0.058)	(0.110)	(0.105)	(0.078)
	1000	0.415	0.541	0.742	0.811	0.660	0.933
		(0.058)	(0.189)	(0.055)	(0.087)	(0.080)	(0.074)
	2000	0.218	0.295	0.729	0.691	0.623	0.930
		(0.048)	(0.093)	(0.049)	(0.065)	(0.064)	(0.071)
Network	500	4.672	7.114	9.901	7.278	7.049	9.792
		(0.983)	(1.108)	(0.623)	(0.766)	(0.770)	(0.657)
	1000	2.849	4.565	9.683	5.607	6.580	9.622
		(0.623)	(0.750)	(0.570)	(0.544)	(0.698)	(0.642)
	2000	1.729	3.018	9.582	4.430	6.403	9.561
		(0.381)	(0.515)	(0.583)	(0.393)	(0.679)	(0.696)
SPD matrix (power metric)	500	0.443	1.084	1.118	0.865	—	—
		(0.090)	(0.283)	(0.076)	(0.093)	—	—
	1000	0.279	0.582	1.099	0.726	—	—
		(0.045)	(0.103)	(0.068)	(0.080)	—	—
	2000	0.187	0.346	1.083	0.637	—	—
		(0.034)	(0.039)	(0.057)	(0.067)	—	—
SPD matrix (BW metric)	500	0.342	—	—	—	—	—
		(0.029)	—	—	—	—	—
	1000	0.288	—	—	—	—	—
		(0.026)	—	—	—	—	—
	2000	0.250	—	—	—	—	—
		(0.025)	—	—	—	—	—

The distributional outputs in our simulations correspond to Gaussian distributions characterized by their mean and standard deviation and therefore lie on a two-dimensional manifold embedded in the Wasserstein space. This structure is highly aligned with the assumptions of DFR, which is specifically designed to exploit low-dimensional geometry in the output space. Despite this favorable setting, E2M consistently outperforms DFR across all sample sizes, achieving at least a 23% reduction in prediction error. This result further underscores the flexibility and effectiveness of E2M, even in scenarios where competing methods are specifically adapted to handle the underlying data structure.

Table 2: Average mean squared prediction errors and standard deviations (in parentheses) for distributional and network outputs at $n = 10,000$ using the anchor-based strategy.

Output	E2M	GFR
Distribution	0.088 (0.017)	0.717 (0.048)
Network	1.072 (0.356)	9.416 (0.565)

To complement the MSPE comparisons, Figure 3 (b) provides representative visualizations of fitted and true distributional outputs for four randomly selected sample units in the case of $n = 500$. The fitted distributions produced by E2M closely track both the location and dispersion of the corresponding true distributions.

6.4 Scalability via Anchor-Based Strategy

To evaluate scalability, we conducted an additional experiment using the anchor-based strategy introduced in Remark 6. Competing methods SDR and IFR become impractical beyond $n = 2000$, requiring more than 20 minutes and one hour per run, respectively, and DFR is similarly limited due to the need to compute large pairwise distance matrices and run Dijkstra’s algorithm. For this reason, we focused on comparing E2M against GFR at larger scales.

Instead of using all training outputs as anchors for computing the weighted Fréchet mean, we fixed a random subset of 1000 outputs as anchors, which makes the optimization complexity independent of the total sample size. Following the same simulation setup as before for both distributional and network outputs, we considered a large-scale setting with $n = 10,000$ and compared E2M against GFR over 200 Monte Carlo replications.

Table 2 reports the results for $n = 10,000$, while results for $n = 500, 1000, 2000$ can be found in Table 1 for comparison. E2M scales efficiently: predictive accuracy continues to improve with n and training with $n = 10,000$ completed in about 5 minutes on a standard laptop, only slightly longer than the 4 minutes required at $n = 2000$ when using all outputs as anchors. These findings confirm that the anchor-based strategy provides strong scalability while preserving predictive performance.

7. Data Applications

We evaluate the effectiveness of E2M on two real-world regression tasks: modeling age-at-death distributions from international human mortality data and predicting daily transportation networks from New York City yellow taxi data. Both applications involve complex non-Euclidean outputs, probability distributions and networks, that cannot be adequately modeled using classical regression techniques.

7.1 Human Mortality Data

We analyze age-at-death distributions across 162 countries in the year 2015 using life table data published by the United Nations World Population Prospects 2024 (<https://www.un.org/en/development/desa/population/publications/wpp2024/>).

Table 3: Predictors of human mortality data.

Category	Predictor	Explanation
Demography	1. Population Density	population per square kilometer
	2. Sex Ratio	number of males per 100 females in the population
	3. Mean Childbearing Age	average age of mothers at the birth of their children
Economics	4. GDP	gross domestic product per capita
	5. GVA by Agriculture	percentage of agriculture, hunting, forestry and fishing activities of gross value added
	6. CPI	consumer price index treating 2010 as the base year
	7. Unemployment Rate	percentage of unemployed people in the labor force
	8. Health Expenditure	percentage of expenditure on health of GDP
Environment	9. Arable Land	percentage of total land area

[//population.un.org/wpp/downloads](https://population.un.org/wpp/downloads)). For each country and age group, the life table reports the number of deaths aggregated in five-year age intervals, forming histograms with uniform bin width. Using the `frechet` package (Chen et al., 2023), we apply local linear smoothing to these histograms and standardize them using trapezoidal integration, yielding continuous probability density functions that serve as regression outputs. Each country is associated with a nine-dimensional predictor vector comprising demographic, economic and environmental indicators, listed in Table 3. These predictors capture key aspects of socioeconomic conditions, such as population density, GDP per capita and healthcare expenditures, which are known to influence life expectancy and mortality patterns.

Predictive performance is assessed via leave-one-out cross-validation, with MSPE as the evaluation criterion. Table 4 reports the results, showing that E2M achieves the lowest MSPE among all methods considered. Despite the modest sample size, E2M demonstrates clear gains over competing approaches, underscoring its ability to capture nonlinear relationships between country-level covariates and complex distributional outcomes. To further illustrate prediction quality, Figure 4 (a) compares the observed and fitted age-at-death densities for four countries (Canada, Portugal, Oman and India). The fitted densities closely track the main structural features of the observed distributions, providing qualitative evidence of the model’s adequacy in the $n = 162$ setting. To explore covariate effects, Figure 4 (b) illustrates the predicted age-at-death densities obtained by varying GDP per capita from low to high values while holding all other predictors fixed at their median levels; the

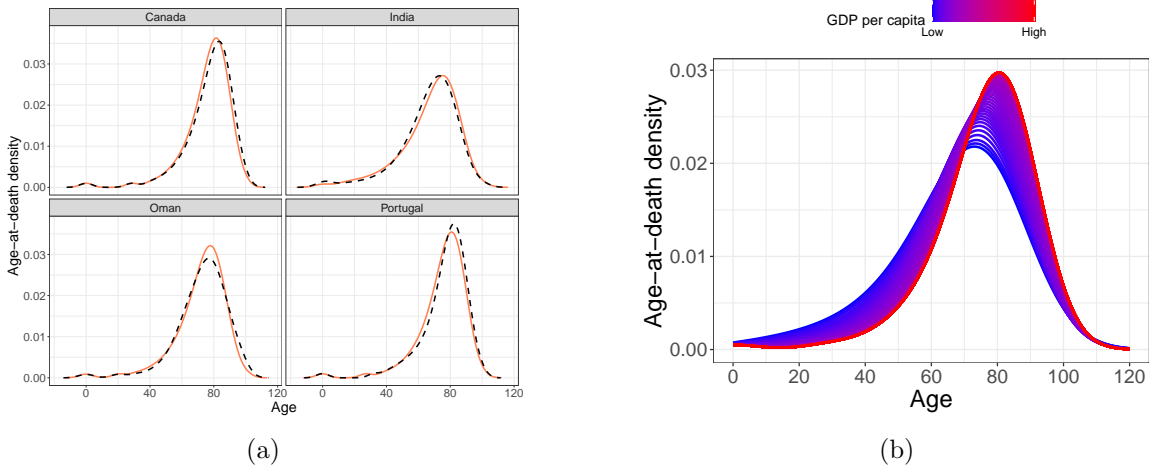


Figure 4: (a) Comparison between observed and predicted age-at-death densities for four countries: Canada, Portugal, Oman and India. The coral solid curves denote the densities estimated by E2M; the black dashed curves denote the corresponding observed empirical densities. (b) Age-at-death densities at different levels of GDP.

Table 4: Average mean squared prediction errors and standard deviations (in parentheses) of E2M, deep Fréchet regression (DFR) (Iao et al., 2025), global Fréchet regression (GFR) (Petersen and Müller, 2019), sufficient dimension reduction (SDR) (Zhang et al., 2024) and single index Fréchet regression (IFR) (Bhattacharjee and Müller, 2023) for human mortality and taxi network data.

Data	E2M	DFR	GFR	SDR	IFR
Human mortality	22.64 (41.32)	26.75 (51.19)	31.32 (58.83)	27.60 (44.40)	42.57 (76.22)
Taxi network	6.83 (0.41)	7.93 (0.50)	12.40 (0.18)	13.38 (0.30)	42.66 (5.56)

resulting rightward shift of the distributions indicates increasing life expectancy as GDP rises.

7.2 New York City Yellow Taxi Data

We study daily transportation patterns in Manhattan using yellow taxi trip records released by the New York City Taxi and Limousine Commission (<https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page>). These records include detailed information such as pick-up and drop-off locations, trip distances, passenger counts, fares and payment methods. To capture external influences, we also collect daily weather history from Weather Underground (<https://www.wunderground.com/history/daily/us/ny/new-york-city/KLGA/date>), including temperature, humidity, wind speed, pressure and precipitation.

Table 5: Predictors of New York City taxi network data.

Category	Predictor	Explanation
Weather	1. Temp	daily average temperature
	2. Humidity	daily average humidity
	3. Wind	daily average windspeed
	4. Pressure	daily average barometric pressure
	5. Precipitation	daily total precipitation
Year	6. Year	indicator for the year of 2018
Day	7. Mon to Thur	indicator for Monday to Thursday
	8. Friday or Saturday	indicator for Friday or Saturday
Trip	9. Passenger Count	daily average number of passengers
	10. Trip Distance	daily average trip distance
	11. Fare Amount	daily average fare amount
	12. Tip Amount	daily average tip amount
	13. Tolls Amount	daily average tolls amount

Following the preprocessing steps described in Zhou and Müller (2022), the 66 original taxi zones were grouped into 13 regions. For each day between January 1, 2018 and December 31, 2019, we constructed a weighted directed network where nodes represent regions and edge weights denote the number of passengers traveling between region pairs. Each network is represented by a 13×13 graph Laplacian matrix and paired with a 13-dimensional predictor vector comprising daily weather information, calendar effects (e.g., day-of-week indicators) and aggregated trip statistics. A full list of predictors is shown in Table 5.

To evaluate predictive performance, ten-fold cross-validation was repeated across 100 Monte Carlo runs, computing MSPE for each method. Results in Table 4 show that E2M achieves the lowest prediction error, outperforming all competing methods by a substantial margin. By explicitly respecting the geometry of network outputs, E2M is able to capture complex dependencies in daily passenger flows that methods relying on embeddings or restrictive assumptions fail to model adequately.

8. Discussion

This paper introduces E2M, a novel end-to-end regression framework for metric space-valued outputs that fully exploits the representational capacity of deep neural networks while respecting the geometry of the output space. By incorporating entropy regularization into the learned weight distribution, E2M enables a data-driven trade-off between localized regression and global smoothing. We establish a universal approximation theorem that demonstrates the expressive power of E2M for approximating conditional Fréchet means and analyze the algorithmic convergence for the proposed training algorithm. Empirically, E2M achieves superior performance across diverse simulated and real-world datasets with complex metric space-valued outputs.

From a learning theoretic perspective, prior work on prediction in metric spaces has primarily focused on characterizing when Bayes consistent prediction is possible and on constructing nonparametric, data-supported rules that achieve such guarantees. Existing results address classification with metric inputs and discrete labels (Hanneke et al., 2021) or regression under general metric losses using fixed aggregation schemes such as nearest-neighbor or medoid-based rules (Cohen and Kontorovich, 2022). In contrast, E2M adopts a complementary viewpoint centered on expressive modeling and optimization: predictions are defined implicitly through learned, input-dependent weighting mechanisms parameterized by neural networks. Extending learning-theoretic guarantees, such as Bayes consistency or finite-sample risk bounds, to such end-to-end geometry-aware models remains an important direction for future work.

Beyond prediction, an important practical question concerns the assessment of covariate effects and their uncertainty. In Appendix H, we provide a permutation-based variable importance procedure adapted to the Fréchet regression setting. This model-agnostic approach quantifies the predictive relevance of each covariate by measuring performance degradation under permutation, offering a practical tool for exploratory analysis in complex metric space-valued models. However, this procedure does not constitute formal hypothesis testing in the classical regression sense. Developing principled methods for statistical inference and uncertainty quantification in Fréchet regression remains technically challenging due to the nonlinear geometry of the output space and the implicit definition of the Fréchet mean. Constructing confidence regions, testing covariate effects and deriving asymptotic distributions for learned predictors are important open problems that warrant further investigation.

Several additional limitations suggest future research directions. First, E2M assumes access to a well defined metric on the output space. In practice, multiple plausible metrics may exist, each inducing different geometric and statistical properties; the choice of metric can substantially influence model behavior. Integrating metric learning techniques (Xing et al., 2002; Kaya and Bilge, 2019) with geometry-aware regression may provide a principled way to tailor the metric to the data and task at hand. Second, although many metric spaces of practical interest are Hadamard spaces, part of our theoretical analysis, in particular the convergence guarantees for the training algorithm, relies on this assumption. Extending the theoretical framework to broader classes of metric spaces would further enhance the generality of the method. Finally, a promising extension is to settings where both inputs and outputs lie in general metric spaces. Combining geometry-aware output modeling as in E2M with recent advances in geometric deep learning for non-Euclidean inputs (Bronstein et al., 2017) could enable flexible regression frameworks for tasks such as network to network prediction or mapping between structured geometric objects.

Acknowledgments

We wish to thank the editor and the reviewers for their valuable comments and suggestions, which have significantly improved the quality of the paper. This research was partially supported by NSF grant DMS-2310450.

Appendix A. Characterizations of Mean and Conditional Mean

Let $Y \in \mathbb{R}$ be a real-valued random variable with finite second moment. Then, the expectation of Y can be equivalently characterized as the minimizer of the expected squared deviation:

$$E[Y] = \arg \min_{y \in \mathbb{R}} E[(Y - y)^2].$$

This identity follows by expanding the square and minimizing

$$E[(Y - y)^2] = E[Y^2] - 2yE[Y] + y^2.$$

Solving

$$\frac{d}{dy} E[(Y - y)^2] = 0$$

yields $y = E[Y]$ as the unique minimizer.

For a random pair $(X, Y) \in \mathbb{R}^p \times \mathbb{R}$, the conditional expectation of Y given $X = x$ can be similarly expressed as

$$E[Y|X = x] = \arg \min_{y \in \mathbb{R}} E[(Y - y)^2|X = x].$$

These characterizations naturally extend to random objects taking values in a general metric space (Ω, d) . When $Y \in \Omega$, the Fréchet mean (Fréchet, 1948) is defined as the minimizer of the expected squared distance,

$$E_{\oplus}[Y] = \arg \min_{y \in \Omega} E[d^2(y, Y)].$$

For a random pair $(X, Y) \in \mathbb{R}^p \times \Omega$, the conditional Fréchet mean (Petersen and Müller, 2019) is given by

$$E_{\oplus}[Y|X = x] = \arg \min_{y \in \Omega} E[d^2(y, Y)|X = x].$$

When $\Omega = \mathbb{R}$ and $d(y, Y) = |y - Y|$, these definitions recover the classical mean and conditional mean.

Appendix B. Characterization of Regression via Weighted Average

The variational view of conditional expectation leads to a natural formulation of regression as the minimization of weighted squared loss. In linear regression, the regression function is assumed to be

$$m(x) = E[Y|X = x] = \beta_0 + \beta_1^{\top}(x - \mu), \tag{1}$$

where $\mu = E[X]$. The coefficients β_0 and β_1 are chosen to minimize the expected squared residual:

$$(\beta_0, \beta_1) = \arg \min_{\beta_0, \beta_1} E_X \left[E_{Y|X} \left\{ \left(Y - \beta_0 - \beta_1^{\top}(X - \mu) \right)^2 \right\} \right].$$

Letting $\Sigma_{XX} = \text{Cov}(X)$ and $\Sigma_{XY} = E[(X - \mu)Y]$, the optimal coefficients are

$$\beta_0 = E[Y], \quad \beta_1 = \Sigma_{XX}^{-1} \Sigma_{XY}.$$

Substituting these into (1) gives

$$\begin{aligned} m(x) &= E[Y] + \Sigma_{XY}^\top \Sigma_{XX}^{-1} (x - \mu) \\ &= E \left[Y + Y(X - \mu)^\top \Sigma_{XX}^{-1} (x - \mu) \right] \\ &= E[w(x; X)Y], \end{aligned}$$

with weight function

$$w(x; X) = 1 + (X - \mu)^\top \Sigma^{-1} (x - \mu).$$

Since $E[w(x; X)] = 1$, we may express the regression function as the minimizer of a weighted squared deviation:

$$\begin{aligned} m(x) &= \arg \min_{y \in \mathbb{R}} \{y - E[w(x; X)Y]\}^2 \\ &= \arg \min_{y \in \mathbb{R}} \{y^2 - 2yE[w(x; X)Y]\} \\ &= \arg \min_{y \in \mathbb{R}} E[y^2 w(x; X) - 2yw(x; X)Y + w(x; X)Y^2] \\ &= \arg \min_{y \in \mathbb{R}} E[w(x; X)(Y - y)^2]. \end{aligned}$$

This formulation reveals that linear regression solves a weighted least squares problem, where weights reflect the alignment between input and the target point x .

This perspective also applies to local linear regression. For simplicity, we consider scalar predictors $X \in \mathbb{R}$. The local linear estimator (Fan and Gijbels, 1996) $m(x) = \beta_0(x)$ is defined via a locally weighted least squares problem:

$$(\beta_0, \beta_1) = \arg \min_{\beta_0, \beta_1} E \left[K_h(X - x) \{Y - \beta_0 - \beta_1(X - x)\}^2 \right], \quad (2)$$

where K_h is a kernel function with bandwidth h .

Writing $\mu_j = E[K_h(X - x)(X - x)^j]$, $r_j = E[K_h(X - x)(X - x)^j Y]$ and $\sigma_0^2 = \mu_0 \mu_2 - \mu_1^2$, the solutions to (2) are

$$\beta_0(x) = \frac{\mu_2 r_0 - \mu_1 r_1}{\sigma_0^2}, \quad \beta_1(x) = \frac{\mu_0 r_1 - \mu_1 r_0}{\sigma_0^2}.$$

Therefore, the local linear regression function is

$$\begin{aligned} m(x) &= \beta_0(x) = \frac{\mu_2 r_0 - \mu_1 r_1}{\sigma_0^2} \\ &= \frac{1}{\sigma_0^2} E[\mu_2 K_h(X - x)Y - \mu_1 K_h(X - x)(X - x)Y] \\ &= E[w(x; X)Y], \end{aligned}$$

where the weight function is given by

$$w(x; X) = \frac{K_h(X - x)(\mu_2 - \mu_1(X - x))}{\sigma_0^2}.$$

Observe that

$$\begin{aligned} E[w(x; X)] &= E\left[\frac{K_h(X-x)(\mu_2 - \mu_1(X-x))}{\sigma_0^2}\right] \\ &= \frac{\mu_2\mu_0 - \mu_1^2}{\sigma_0^2} \\ &= 1. \end{aligned}$$

Similarly to linear regression, the local linear regression function can be alternatively represented as the minimizer of a weighted squared deviation:

$$\begin{aligned} m(x) &= \arg \min_{y \in \mathbb{R}} \{y - E[w(x; X)Y]\}^2 \\ &= \arg \min_{y \in \mathbb{R}} E[w(x; X)(Y - y)^2]. \end{aligned}$$

The local linear estimator can therefore be viewed as solving a localized version of the weighted least squares problem, where weights adapt to the target x through a kernel mechanism.

These characterizations establish a unifying framework in which both linear and local linear regression estimate the conditional mean through weighted minimization of squared deviations. The nature of the weight function $w(x; X)$, whether globally defined or locally adaptive, determines the behavior and flexibility of the estimator. This weighting perspective motivated the development of Fréchet regression (Petersen and Müller, 2019), which generalizes classical regression techniques to settings with metric space-valued outputs. Specifically, linear regression extends naturally to global Fréchet regression, while local linear regression corresponds to local Fréchet regression. These extensions are achieved by replacing Euclidean distances with general metric distances, thereby preserving the interpretation of regression as a weighted deviation minimization. The resulting estimators retain the structural intuition of their classical counterparts while enabling flexible modeling of complex, structured data.

Appendix C. Proofs

C.1 Proof of Lemma 1

Proof We apply the Berge Maximum Theorem (Aliprantis and Border, 2006, Theorem 17.31) to the weighted Fréchet mean minimization problem. For completeness, we restate the theorem below.

Berge Maximum Theorem (Aliprantis and Border, 2006, Theorem 17.31) Let $\varphi : X \rightarrow Y$ be a continuous correspondence between topological spaces with nonempty compact values. Suppose the function $f : \text{Gr } \varphi \mapsto \mathbb{R}$ is continuous, where $\text{Gr } \varphi = \{(x, y) \in X \times Y | y \in \varphi(x)\}$ denotes the graph of φ . Define the “value function” $m : X \mapsto \mathbb{R}$ by

$$m(x) = \max_{y \in \varphi(x)} f(x, y),$$

and the correspondence $\mu : X \rightarrow Y$ of maximizers by

$$\mu(x) = \{y \in \varphi(x) \mid f(x, y) = m(x)\}.$$

Then:

1. The value function m is continuous.
2. The “argmax” correspondence μ has nonempty compact values.
3. If either f has a continuous extension to all of $X \times Y$ or Y is Hausdorff, then the “argmax” correspondence μ is upper hemicontinuous.

We now match our setting to the theorem. Let $X = \Delta^{n-1}$, the $(n - 1)$ -simplex and $Y = \Omega$, the ambient metric space. Define the correspondence $\varphi : \Delta^{n-1} \rightarrow \Omega$ by $\varphi(w) = \Omega$ for all $w \in \Delta^{n-1}$, so that $\text{Gr } \varphi = \Delta^{n-1} \times \Omega$. Since Ω is compact and Hausdorff, $\varphi(w)$ has nonempty compact values and the graph is well-defined.

We define the objective function $f : \Delta^{n-1} \times \Omega \mapsto \mathbb{R}$ by

$$f(w, y) = - \sum_{i=1}^n w_i d^2(y, Y_i).$$

Here we include a minus sign so that minimizing $\sum_{i=1}^n w_i d^2(y, Y_i)$ becomes equivalent to maximizing $f(w, y)$, as required for the application of the Berge Maximum Theorem.

We now verify the continuity of f . For fixed $y \in \Omega$, $w \mapsto f(w, y)$ is linear, hence continuous. Next, fix $w \in \Delta^{n-1}$ and consider continuity in y . For any $y_1, y_2 \in \Omega$, we have

$$\begin{aligned} |f(w, y_2) - f(w, y_1)| &= \left| \sum_{i=1}^n w_i (d^2(y_2, Y_i) - d^2(y_1, Y_i)) \right| \\ &\leq \sum_{i=1}^n w_i |d^2(y_2, Y_i) - d^2(y_1, Y_i)|. \end{aligned}$$

Using the identity

$$d^2(y_2, Y_i) - d^2(y_1, Y_i) = (d(y_2, Y_i) + d(y_1, Y_i))(d(y_2, Y_i) - d(y_1, Y_i)),$$

and applying the triangle inequality for the metric d , we obtain

$$|d^2(y_2, Y_i) - d^2(y_1, Y_i)| \leq 2Dd(y_2, y_1),$$

where $D = \sup_{u,v \in \Omega} d(u, v)$ is the diameter of Ω and is finite by compactness.

Thus,

$$|f(w, y_2) - f(w, y_1)| \leq 2Dd(y_2, y_1) \sum_{i=1}^n w_i = 2Dd(y_2, y_1),$$

since $\sum_{i=1}^n w_i = 1$. Therefore, $f(w, \cdot)$ is Lipschitz continuous with constant $2D$ in y and hence continuous.

Since f is continuous in both w and y , and $\text{Gr } \varphi = \Delta^{n-1} \times \Omega$ is compact, f is jointly continuous on $\text{Gr } \varphi$ and we conclude that all conditions of the Berge Maximum Theorem are satisfied. It follows that the value function

$$m(w) = \max_{y \in \Omega} f(w, y)$$

is continuous, and that the correspondence

$$\mu(w) = \{y \in \Omega \mid f(w, y) = m(w)\}$$

has nonempty compact values and is upper hemicontinuous.

Under Assumption 1, the weighted Fréchet mean $\mu(w)$ is unique for each $w \in \Delta^{n-1}$. Hence $\mu : \Delta^{n-1} \rightarrow \Omega$ is singleton-valued, and for singleton-valued correspondences, upper hemicontinuity implies continuity. Therefore, μ is continuous on Δ^{n-1} . ■

C.2 Proof of Theorem 2

Proof We aim to show that there exists a neural network w_{θ^*} such that the composed function $m_{\theta^*} = \mu \circ w_{\theta^*}$ uniformly approximates the true function $m = \mu \circ w$ over $\{x : \|x\| \leq 1\}$.

Since w is continuous, by the universal approximation theorem (Cybenko, 1989; Hornik, 1991), for any $\delta > 0$, there exists a neural network w_{θ^*} such that:

$$\sup_{\|x\| \leq 1} \|w_{\theta^*}(x) - w(x)\|_2 < \delta.$$

The weighted Fréchet mean map $\mu : \Delta^{n-1} \mapsto \Omega$ is continuous by Lemma 1. Since both w_{θ^*} and w are continuous, their images over $\{x : \|x\| \leq 1\}$ form a compact subset of \mathbb{R}^n . Hence, the continuity of μ implies uniform continuity on this set. Therefore, for any $\epsilon > 0$, there exists $\delta > 0$ such that

$$\|w_{\theta^*}(x) - w(x)\|_2 < \delta \quad \Rightarrow \quad d(\mu(w_{\theta^*}(x)), \mu(w(x))) < \epsilon.$$

We conclude that for any $\epsilon > 0$, there exists a neural network w_{θ^*} such that

$$\sup_{\|x\| \leq 1} d(m_{\theta^*}(x), m(x)) = \sup_{\|x\| \leq 1} d(\mu(w_{\theta^*}(x)), \mu(w(x))) < \epsilon.$$

Now consider the stochastic case. Since X is stochastically bounded, for any $\delta > 0$, there exists a constant $M_\delta > 0$ such that $P(\|X\| \leq M_\delta) > 1 - \delta$. Repeating the same argument over the compact set $\{X : \|X\| \leq M_\delta\}$, we conclude

$$P(d(m_{\theta^*}(X), m(X)) < \epsilon) > 1 - \delta.$$

This completes the proof. ■

C.3 Proof of Lemma 5

Proof Let $\mu_1 = \mu(w_1)$ and $\mu_2 = \mu(w_2)$ be the weighted Fréchet means corresponding to weights $w_1, w_2 \in \Delta^{n-1}$. We apply the variance inequality for Hadamard spaces (Sturm, 2003, Proposition 4.4). For $\mu_1 = \mu(w_1)$ and any $z \in \Omega$, we have:

$$\sum_{i=1}^n w_{1,i} d^2(z, Y_i) - \sum_{i=1}^n w_{1,i} d^2(\mu_1, Y_i) \geq d^2(z, \mu_1).$$

Taking $z = \mu_2$ yields:

$$\sum_{i=1}^n w_{1,i} d^2(\mu_2, Y_i) - \sum_{i=1}^n w_{1,i} d^2(\mu_1, Y_i) \geq d^2(\mu_1, \mu_2). \quad (3)$$

Similarly, for $\mu_2 = \mu(w_2)$ and $z = \mu_1$:

$$\sum_{i=1}^n w_{2,i} d^2(\mu_1, Y_i) - \sum_{i=1}^n w_{2,i} d^2(\mu_2, Y_i) \geq d^2(\mu_1, \mu_2). \quad (4)$$

Adding (3) and (4), we obtain:

$$\sum_{i=1}^n (w_{1,i} - w_{2,i}) (d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i)) \geq 2d^2(\mu_1, \mu_2). \quad (5)$$

Let $a_i = w_{1,i} - w_{2,i}$ and $b_i = d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i)$. Applying the Cauchy-Schwarz inequality:

$$\left| \sum_{i=1}^n a_i b_i \right| \leq \left(\sum_{i=1}^n a_i^2 \right)^{1/2} \left(\sum_{i=1}^n b_i^2 \right)^{1/2} = \|w_1 - w_2\|_2 \left(\sum_{i=1}^n (d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i))^2 \right)^{1/2}.$$

To bound the b_i terms, observe that by the triangle inequality:

$$|d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i)| \leq |d(\mu_2, Y_i) - d(\mu_1, Y_i)| (d(\mu_2, Y_i) + d(\mu_1, Y_i)) \leq 2Dd(\mu_1, \mu_2),$$

where $D = \sup_{u,v \in \Omega} d(u, v)$ is the diameter of Ω and is finite by boundedness.

Therefore,

$$(d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i))^2 \leq 4D^2 d^2(\mu_1, \mu_2),$$

and

$$\sum_{i=1}^n (d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i))^2 \leq 4D^2 n d^2(\mu_1, \mu_2).$$

So we conclude:

$$\left| \sum_{i=1}^n (w_{1,i} - w_{2,i}) (d^2(\mu_2, Y_i) - d^2(\mu_1, Y_i)) \right| \leq 2D\sqrt{n} \|w_1 - w_2\|_2 d(\mu_1, \mu_2).$$

Combining with (5),

$$2D\sqrt{n} \|w_1 - w_2\|_2 d(\mu_1, \mu_2) \geq 2d^2(\mu_1, \mu_2).$$

Divide both sides by $2d(\mu_1, \mu_2)$ (noting that the inequality holds trivially if $d(\mu_1, \mu_2) = 0$), we get

$$d(\mu_1, \mu_2) \leq D\sqrt{n} \|w_1 - w_2\|_2,$$

which completes the proof. ■

C.4 Proof of Theorem 7

Proof We first show that the loss $\ell(\theta; (X, Y))$ is bounded below. By definition,

$$\ell(\theta; (X, Y)) = d^2(m_\theta(X), Y) + \lambda H(w_\theta(X)),$$

where $m_\theta = \mu \circ w_\theta$. Since $d^2(\cdot, \cdot) \geq 0$, it suffices to lower bound the second term.

Recall that

$$H(w) = - \sum_{i=1}^n w_i \log(w_i + \delta),$$

where $\delta > 0$ is a small regularization parameter to avoid taking the log of zero. The function $H(w)$ is minimized when the distribution w is as concentrated as possible, that is, when $w_i = 1$ for some i and $w_j = 0$ for all $j \neq i$. Therefore, for any $w \in \Delta^{n-1}$,

$$-\log(1 + \delta) \leq H(w) \leq -\log(\delta).$$

Thus, for all θ ,

$$\ell(\theta; (X, Y)) \geq -|\lambda| \max\{|\log(1 + \delta)|, |\log(\delta)|\},$$

and taking expectations yields

$$\mathcal{L}(\theta) = E[\ell(\theta; (X, Y))] \geq -|\lambda| \max\{|\log(1 + \delta)|, |\log(\delta)|\}.$$

We now establish the Lipschitz continuity of $\ell(\theta; (X, Y))$ with respect to θ . By Assumption 2(i), w_θ is L -Lipschitz. Moreover, by Lemma 5, the weighted Fréchet mean map $\mu : \Delta^{n-1} \mapsto \Omega$ is $D\sqrt{n}$ -Lipschitz, where $D = \sup_{u, v \in \Omega} d(u, v)$ is the diameter of Ω . Thus, the overall map $m_\theta = \mu \circ w_\theta$ is $LD\sqrt{n}$ -Lipschitz continuous with respect to θ .

The squared distance term $d^2(m_\theta(X), Y)$ satisfies

$$\begin{aligned} & |d^2(m_{\theta_2}(X), Y) - d^2(m_{\theta_1}(X), Y)| \\ &= |d(m_{\theta_2}(X), Y) - d(m_{\theta_1}(X), Y)| (d(m_{\theta_2}(X), Y) + d(m_{\theta_1}(X), Y)), \end{aligned}$$

and using the triangle inequality,

$$|d(m_{\theta_2}(X), Y) - d(m_{\theta_1}(X), Y)| \leq d(m_{\theta_2}(X), m_{\theta_1}(X)).$$

Since $d(m_{\theta_2}(X), Y), d(m_{\theta_1}(X), Y) \leq D$, we obtain

$$|d^2(m_{\theta_2}(X), Y) - d^2(m_{\theta_1}(X), Y)| \leq 2Dd(m_{\theta_2}(X), m_{\theta_1}(X)).$$

Thus, the contribution of the distance term to the Lipschitz constant is at most $2LD^2\sqrt{n}$.

Next, we bound the entropy term. Recall

$$H(w) = - \sum_{i=1}^n w_i \log(w_i + \delta),$$

and its gradient with respect to w has coordinates

$$\frac{\partial H}{\partial w_i} = -\log(w_i + \delta) - \frac{w_i}{w_i + \delta}.$$

Since $w_i \in [0, 1]$, we have

$$\left| \frac{\partial H}{\partial w_i} \right| \leq |\log \delta| + 1.$$

Thus, the gradient of H is bounded in ℓ_∞ norm by $|\log \delta| + 1$, and in ℓ_2 norm by

$$\|\nabla H(w)\|_2 \leq \sqrt{n}(|\log \delta| + 1).$$

Therefore,

$$|H(w_{\theta_2}(X)) - H(w_{\theta_1}(X))| \leq \sqrt{n}(|\log \delta| + 1)L\|\theta_2 - \theta_1\|_2.$$

Multiplying by λ gives a contribution of $|\lambda|\sqrt{n}(|\log \delta| + 1)L$ to the Lipschitz constant.

Summing the contributions of the two terms, the total Lipschitz constant is

$$G = L\sqrt{n} (2D^2 + |\lambda|(|\log \delta| + 1)).$$

Since ℓ is differentiable and Lipschitz continuous with constant G , we have

$$\|\nabla_{\theta} \ell(\theta; (X, Y))\|_2 \leq G.$$

Finally, applying Corollary 2 from [Zaheer et al. \(2018\)](#) yields

$$E [\|\nabla \mathcal{L}(\theta_\tau)\|_2^2] \leq O\left(\frac{1}{T} + \frac{1}{b}\right),$$

completing the proof. ■

Appendix D. Choice of Hyperparameters

The hyperparameters for E2M can be selected using a grid search over the candidate values listed in Table 6. The optimal combination of hyperparameters is chosen to minimize the mean squared prediction error for the validation data.

Table 6: Hyperparameter settings.

Regularization parameter	-0.01	-0.001	0	0.001	0.01
Number of hidden layer	2	3	4	5	6
Number of neurons	8	16	32	64	128

Table 7: Average mean squared prediction errors with standard deviations (in parentheses) for sensitivity analysis of entropy regularization in distributional outputs.

n	-0.1	-0.05	-0.01	0	0.01	0.05	0.1
500	0.661 (0.172)	0.557 (0.174)	0.552 (0.195)	0.717 (0.199)	0.718 (0.163)	0.974 (0.122)	1.154 (0.177)
1000	0.507 (0.105)	0.405 (0.116)	0.368 (0.131)	0.509 (0.176)	0.584 (0.169)	0.886 (0.163)	1.104 (0.173)
2000	0.457 (0.071)	0.341 (0.080)	0.291 (0.104)	0.444 (0.136)	0.492 (0.140)	0.816 (0.187)	1.033 (0.210)

Appendix E. Sensitivity Analysis on Entropy Regularization

Entropy regularization controls the sharpness of the learned weight distribution. Negative values of the regularization parameter λ encourage higher-entropy (more uniform) weights, leading to a global smoothing effect. Positive values of λ favor lower-entropy (more concentrated) weights, pushing the model toward stronger localization.

To assess robustness, we conducted a sensitivity analysis under the same distributional simulation setup as in Section 6, fixing the neural network to two hidden layers with eight neurons each. The entropy regularization parameter was varied over

$$\lambda \in \{-0.1, -0.05, -0.01, 0, 0.01, 0.05, 0.1\},$$

and performance was evaluated in terms of MSPE averaged over 200 Monte Carlo replications. Table 7 reports the average MSPE along with standard deviations.

The sensitivity analysis reveals several clear trends. The best performance across all sample sizes occurs at $\lambda = -0.01$, indicating that mild negative regularization provides the most effective balance between global smoothing and local adaptivity. Slightly less negative values such as $\lambda = -0.05$ also perform well, while stronger negative regularization (e.g., $\lambda = -0.1$) leads to oversmoothing and reduced accuracy at larger sample sizes. When $\lambda \geq 0$, performance deteriorates because the model already incorporates natural localization through the softmax weighting, and additional positive entropy penalties force the weights to concentrate further. This over-concentration reduces the effective sample size, increases variance and leads to poor generalization, particularly for large positive values such as $\lambda = 0.05$ or 0.1 .

Overall, these results confirm that E2M is robust to moderate variations in λ and benefits most from mild negative values that encourage balanced weighting across training samples.

Table 8: Average mean squared prediction errors (mean on first line, standard deviation in parentheses on second line) of E2M, deep Fréchet regression (DFR) (Iao et al., 2025), global Fréchet regression (GFR) (Petersen and Müller, 2019), sufficient dimension reduction (SDR) (Zhang et al., 2024) and single index Fréchet regression (IFR) (Bhattacharjee and Müller, 2023) for distributional outputs in small-sample regimes.

n	E2M	DFR	GFR	SDR	IFR
50	1.151 (0.154)	1.832 (0.411)	1.433 (0.254)	4.983 (10.645)	1.103 (0.229)
100	1.048 (0.135)	1.720 (0.311)	1.022 (0.112)	1.741 (1.315)	1.002 (0.254)
200	0.851 (0.158)	1.634 (0.289)	0.855 (0.076)	1.135 (0.887)	0.971 (0.240)

Appendix F. Additional Experiments for Small Sample Sizes

To examine the performance of E2M in small-sample regimes, we conduct additional experiments with sample sizes $n \in \{50, 100, 200\}$ under the same distributional simulation setup as in Section 6. Model architectures and tuning parameters are selected using the same cross-validation procedures as in the main experiments. Performance is evaluated using mean squared prediction error over repeated simulation runs.

The results, summarized in Table 8, show that E2M remains competitive for $n = 50$ and $n = 100$, achieving prediction errors comparable to strong baseline methods such as single-index Fréchet regression (IFR) and global Fréchet regression (GFR). As the sample size increases, the advantage of E2M becomes more pronounced: when $n = 200$, E2M attains the lowest mean squared prediction error among all compared methods. These findings indicate that E2M does not rely on very large sample sizes to perform well and can already deliver strong predictive performance at relatively small sample sizes in practice.

Appendix G. Additional Simulation under Linear Relationships

To further assess the behavior of E2M in settings where the true input-output relationship is linear, we conducted an additional simulation study. This experiment addresses the question of how E2M performs when the data-generating mechanism aligns with classical linear regression.

We consider Gaussian distributional outputs as in Section 6, but modify the conditional mean structure to be linear in the predictors. The input vector $X \in \mathbb{R}^{12}$ is generated from a multivariate normal distribution with mean zero and Toeplitz covariance structure $\Sigma_{ij} = 0.8^{|i-j|}$. The conditional mean of the Gaussian distribution is specified as

$$\mu(X) = \sum_{j=1}^{12} \beta_j X_j,$$

with coefficient vector

$$\beta = (1, 0, 1, 0, 1, -1, 1, -1, 1, 0, 1, 0),$$

Table 9: Average mean squared prediction errors (mean on first line, standard deviation in parentheses on second line) of E2M, deep Fréchet regression (DFR) (Iao et al., 2025), global Fréchet regression (GFR) (Petersen and Müller, 2019), sufficient dimension reduction (SDR) (Zhang et al., 2024) and single index Fréchet regression (IFR) (Bhattacharjee and Müller, 2023) for distributional outputs under linear relationships.

n	E2M	DFR	GFR	SDR	IFR
50	1.388 (0.591)	2.167 (0.586)	0.463 (0.178)	1.834 (1.443)	1.587 (0.841)
100	0.814 (0.289)	1.409 (0.300)	0.204 (0.083)	0.609 (0.363)	1.556 (0.708)
200	0.589 (0.205)	1.173 (0.241)	0.102 (0.030)	0.275 (0.124)	1.689 (0.942)
500	0.391 (0.160)	1.041 (0.227)	0.039 (0.009)	0.151 (0.104)	1.611 (0.740)
1000	0.286 (0.135)	0.821 (0.129)	0.022 (0.005)	0.095 (0.085)	1.189 (0.538)
2000	0.188 (0.086)	0.567 (0.082)	0.015 (0.004)	0.074 (0.061)	1.492 (0.605)

while the conditional standard deviation is constant, $\sigma(X) = 2$. To mimic practical settings, we generate N independent samples from each Gaussian distribution and construct empirical distributions as noisy proxies for the latent outputs, following the same procedure as in Section 6.

Table 9 reports the mean squared prediction errors across different sample sizes $n \in \{50, 100, 200, 500, 1000, 2000\}$. In this linear setting, global Fréchet regression (GFR), which directly models a linear relationship between predictors and the conditional Fréchet mean, achieves the lowest prediction error across all sample sizes. Sufficient dimension reduction (SDR) also performs well. E2M remains competitive and improves as n increases, but does not outperform GFR or SDR in this regime. These findings are consistent with the design of E2M. While E2M provides flexible nonlinear modeling capacity, it does not offer a structural advantage in strictly linear settings where simpler models are correctly specified. In such cases, global Fréchet regression is statistically more efficient due to its lower variance and reduced model complexity. This result further highlights that the strength of E2M lies in capturing nonlinear input-output relationships rather than improving performance in settings already well described by linear models.

Appendix H. Permutation-Based Variable Importance

While E2M is primarily designed for flexible prediction in general metric spaces, it is also of interest to assess which covariates contribute most to predictive performance. To this end, we employ a permutation-based variable importance measure, adapted to the Fréchet regression setting. This approach is model-agnostic and has been widely used in machine learning for assessing predictive relevance (Fisher et al., 2019; Molnar, 2025).

Let \widehat{m} denote the trained E2M predictor and let $\mathcal{D}_{\text{test}} = \{(X_i, Y_i)\}_{i=1}^{n_{\text{test}}}$ denote a test dataset. The baseline prediction error is defined as

$$\text{Err}_{\text{base}} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} d^2(Y_i, \widehat{m}(X_i)),$$

where d denotes the metric on the output space.

For each covariate, we generate B independent random permutations. For permutation $b \in \{1, \dots, B\}$, let $X_i^{(j,b)}$ denote the predictor vector in which the j -th coordinate has been randomly permuted across test samples, while all other coordinates remain unchanged. The corresponding prediction error is

$$\text{Err}_j^{(b)} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} d^2(Y_i, \widehat{m}(X_i^{(j,b)})).$$

The permutation importance score for covariate X_j is defined as the average increase in prediction error:

$$\text{VI}_j = \frac{1}{B} \sum_{b=1}^B \text{Err}_j^{(b)} - \text{Err}_{\text{base}}.$$

A larger value of VI_j indicates that permuting X_j leads to a greater degradation in predictive performance, suggesting higher predictive importance.

References

- M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.
- C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis: A Hitchhiker’s Guide*. Springer, Berlin, Heidelberg, 3rd edition, 2006.
- N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.
- P. C. Álvarez-Esteban, E. Del Barrio, J. Cuesta-Albertos, and C. Matrán. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.
- M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL <https://proceedings.mlr.press/v70/arjovsky17a.html>.
- M. Bacák. Computing medians and means in Hadamard spaces. *SIAM Journal on Optimization*, 24(3):1542–1566, 2014a.

- M. Bacák. *Convex Analysis and Optimization in Hadamard Spaces*. De Gruyter Series in Nonlinear Analysis and Applications. De Gruyter, 2014b. ISBN 9783110361629. URL <https://books.google.com/books?id=bI3nBQAAQBAJ>.
- R. Bhatia, T. Jain, and Y. Lim. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019.
- S. Bhattacharjee and H.-G. Müller. Single index Fréchet regression. *Annals of Statistics*, 51(4):1770–1798, 2023.
- M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst. Geometric deep learning: going beyond Euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.
- M. Bulté and H. Sørensen. Medoid splits for efficient random forests in metric spaces. *Computational Statistics & Data Analysis*, 198:107995, 2024.
- L. Capitaine, J. Bigot, R. Thiébaud, and R. Genuer. Fréchet random forests for metric space valued regression with non Euclidean predictors. *Journal of Machine Learning Research*, 25(355):1–41, 2024.
- R. Chakraborty, J. Bouza, J. H. Manton, and B. C. Vemuri. ManifoldNet: A deep neural network for manifold-valued data with applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(2):799–810, 2020.
- X. Chen, S. Liu, R. Sun, and M. Hong. On the convergence of a class of Adam-type algorithms for non-convex optimization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=H1x-x309tm>.
- Y. Chen, Y. Zhou, H. Chen, A. Gajardo, J. Fan, Q. Zhong, P. Dubey, K. Han, S. Bhattacharjee, C. Zhu, S. I. Iao, P. Kundu, A. Petersen, and H.-G. Müller. frechet: Statistical Analysis for Random Objects and Non-Euclidean Data. *R package version 0.3.0*, 2023.
- D. T. Cohen and A. Kontorovich. Learning with metric losses. In P.-L. Loh and M. Raginsky, editors, *Proceedings of Thirty Fifth Conference on Learning Theory*, volume 178 of *Proceedings of Machine Learning Research*, pages 662–700. PMLR, 02–05 Jul 2022.
- G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.
- I. L. Dryden, A. Koloydenko, and D. Zhou. Non-Euclidean statistics for covariance matrices, with applications to diffusion tensor imaging. *Annals of Applied Statistics*, 3(3):1102–1123, 2009.
- J. Fan and I. Gijbels. *Local Polynomial Modelling and its Applications*. Chapman & Hall, London, 1996.
- J. J. Faraway. Regression for non-Euclidean data using distance matrices. *Journal of Applied Statistics*, 41(11):2342–2357, 2014.

- A. Fisher, C. Rudin, and F. Dominici. All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177):1–81, 2019. URL <http://jmlr.org/papers/v20/18-760.html>.
- M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l’Institut Henri Poincaré*, 10(4):215–310, 1948.
- O. Ganea, G. Becigneul, and T. Hofmann. Hyperbolic neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/dbab2adc8f9d078009ee3fa810bea142-Paper.pdf.
- A. Ghosal, W. Meiring, and A. Petersen. Fréchet single index models for object response regression. *Electronic Journal of Statistics*, 17(1):1074–1112, 2023.
- H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning*, 110:393–416, 2021.
- J. Hagenauer and M. Helbich. A geographically weighted artificial neural network. *International Journal of Geographical Information Science*, 36(2):215–235, 2022.
- S. Hanneke, A. Kontorovich, S. Sabato, and R. Weiss. Universal Bayes consistency in metric spaces. *Annals of Statistics*, 49(4):2129–2150, 2021.
- M. Hein. Robust nonparametric regression with metric-space valued output. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/92977ae4d2ba21425a59afb269c2a14e-Paper.pdf.
- K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.
- Z. Huang and L. Van Gool. A Riemannian network for SPD matrix learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- S. I. Iao, Y. Zhou, and H.-G. Müller. Deep Fréchet regression. *Journal of the American Statistical Association*, 120(551):1437–1448, 2025.
- P. Kapli, Z. Yang, and M. J. Telford. Phylogenetic tree building in the genomic age. *Nature Reviews Genetics*, 21(7):428–444, 2020.
- M. Kaya and H. Ş. Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.
- D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL <http://arxiv.org/abs/1412.6980>.

- T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=SJU4ayYgl>.
- E. D. Kolaczyk and G. Csárdi. *Statistical Analysis of Network Data with R*, volume 65. Springer Cham, 2nd edition, 2020.
- T. Le Gouic and J.-M. Loubes. Existence and consistency of Wasserstein barycenters. *Probability Theory and Related Fields*, 168(3):901–917, 2017.
- X. Li, L. Sun, M. Ling, and Y. Peng. A survey of graph neural network based recommendation in social networks. *Neurocomputing*, 549:126441, 2023.
- Z. Lin. Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019.
- Z. Lin and H.-G. Müller. Total variation regularized Fréchet regression for metric-space valued data. *Annals of Statistics*, 49(6):3510–3533, 2021.
- T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=B1QRgziT->.
- C. Molnar. *Interpretable Machine Learning*. Leanpub, 3 edition, 2025. ISBN 978-3-911578-03-5. URL <https://christophm.github.io/interpretable-ml-book>.
- H.-G. Müller. Peter Hall, Functional Data Analysis and Random Objects. *Annals of Statistics*, 44:1867–1887, 2016.
- E. Nadaraya. On Estimating Regression. *Theory of Probability and Its Applications*, 9(1):141–142, 1964.
- T. M. Nye, X. Tang, G. Weyenberg, and R. Yoshida. Principal component analysis and the locus of the Fréchet mean in the space of phylogenetic trees. *Biometrika*, 104(4):901–922, 2017.
- V. M. Panaretos and Y. Zemel. *An Invitation to Statistics in Wasserstein Space*. Springer New York, 2020.
- X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *International Journal of Computer Vision*, 66(1):41–66, 2006.
- A. Petersen and H.-G. Müller. Fréchet regression for random objects with Euclidean predictors. *Annals of Statistics*, 47(2):691–719, 2019.
- A. Petersen, C. Zhang, and P. Kokoszka. Modeling probability density functions as data objects. *Econometrics and Statistics*, 21:159–178, 2022.
- R. Qiu, Z. Yu, and R. Zhu. Random forest weighted local Fréchet regression with random objects. *Journal of Machine Learning Research*, 25(107):1–69, 2024.

- K. E. Severn, I. L. Dryden, and S. P. Preston. Manifold valued data analysis of samples of networks, with applications in corpus linguistics. *Annals of Applied Statistics*, 16(1): 368–390, 2022.
- D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM National Conference*, pages 517–524, 1968.
- D. Song and K. Han. Errors-in-variables Fréchet regression with low-rank covariate approximation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=Sg3aCpWUQP>.
- K.-T. Sturm. Probability measures on metric spaces of nonpositive curvature. *Heat Kernels and Analysis on Manifolds, Graphs, and Metric Spaces (Paris, 2002)*. *Contemp. Math.*, 338. *Amer. Math. Soc., Providence, RI*, 338:357–390, 2003.
- Y. Thanwerdas and X. Pennec. $O(n)$ -invariant Riemannian metrics on SPD matrices. *Linear Algebra and its Applications*, 661:163–201, 2023.
- I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018. URL <https://openreview.net/forum?id=HkL7n1-0b>.
- A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
- C. Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.
- J.-L. Wang, J.-M. Chiou, and H.-G. Müller. Functional Data Analysis. *Annual Review of Statistics and its Application*, 3:257–295, 2016.
- G. S. Watson. Smooth regression analysis. *Sankhyā Series A*, 26(4):359–372, 1964.
- E. Xing, M. Jordan, S. J. Russell, and A. Ng. Distance metric learning with application to clustering with side-information. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL https://proceedings.neurips.cc/paper_files/paper/2002/file/c3e4035af2a1cde9f21e1ae1951ac80b-Paper.pdf.
- C. Ying and Z. Yu. Fréchet sufficient dimension reduction for random objects. *Biometrika*, 109(4):975–992, 2022.
- M. Zaheer, S. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/90365351ccc7437a1309dc64e4db32a3-Paper.pdf.

- Q. Zhang, L. Xue, and B. Li. Dimension reduction for Fréchet regression. *Journal of the American Statistical Association*, 119(548):2733–2747, 2024.
- Y. Zhou and H.-G. Müller. Network regression with graph Laplacians. *Journal of Machine Learning Research*, 23(320):1–41, 2022.
- Y. Zhou and H.-G. Müller. Wasserstein regression with empirical measures and density estimation for sparse data. *Biometrics*, 80(4):ujae127, 2024.
- Y. Zhou, S. I. Iao, and H.-G. Müller. Fréchet geodesic boosting. In D. Belgrave, C. Zhang, L. Montoya, H. Lin, N. Chen, M. Ghassemi, P. Koniusz, and R. Pascanu, editors, *Advances in Neural Information Processing Systems*, 2025.