Metric Learning in an RKHS

Gokcan Tatli[1] **Yi Chen**[1] **Blake Mason**[2] **Robert Nowak**[1] **Ramya Korlakai Vinayak**[1]

[1]Dept. of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, Wisconsin, USA
[2]Amazon.com, USA

Abstract

This paper investigates metric learning in a Reproducing Kernel Hilbert Space (RKHS) based on a set of random triplet comparisons in the form of "*Do you think item h is more similar to item i or item j?*" indicating similarity and differences between various items. The goal is to learn a metric in the RKHS that reflects the comparisons. Nonlinear metric learning using kernel methods and neural networks has shown great empirical promise. While previous works have addressed certain aspects of this problem, there is little or no theoretical understanding of such methods. The exception is the special (linear) case in which the RKHS is the standard d-dimensional Euclidean space; there is a comprehensive theory for metric learning in the d-dimensional Euclidean space. This paper develops a general RKHS framework for metric learning and provides novel generalization guarantees and sample complexity bounds. We validate our findings through a set of simulations and experiments on real datasets. Our code is publicly available at `https://github.com/RamyaLab/metric-learning-RKHS`.

1 INTRODUCTION

Understanding how human perceive objects is essential in many areas from machine learning [Hu et al., 2015, Hsieh et al., 2017] to psychology [Cao et al., 2013, Roads and Mozer, 2019] and policy learning [Liu et al., 2021b]. Learning representations over objects that reflects similarities and dissimilarities on human perception is key to this understanding. Metric learning is the study of learning such a distance function that represents similarities and dissimilarities among objects. This is particularly useful in computer vision applications such as image retrieval [Hoi et al., 2010, Yao et al., 2020] and face recognition [Guillaumin et al., 2009, Cao et al., 2013], and recommendation systems [Zhang et al., 2019, Wu et al., 2020], where the notion of similarity plays a central role on the performance. Comparative judgments over objects has been widely used as a powerful tool in those applications and many others to understand similarities and dissimilarities. In this paper, we provide a theoretical foundation to the task of metric learning from triplet comparisons in the form of *"is item h more similar to item i or to item j?"* (see Figure 1 for an example triplet comparison query for Food-100 dataset Wilber et al. [2014]). We aim to learn a metric that predicts triplet comparisons as well as possible by learning a distance function. Let $\boldsymbol{x} \in \mathbb{R}^d$ be the representation of objects. We are given a random set of triplet comparisons in the form of

$$\operatorname{sign}(\operatorname{dist}^2(\boldsymbol{x}_h, \boldsymbol{x}_i) - \operatorname{dist}^2(\boldsymbol{x}_h, \boldsymbol{x}_j)),$$

which compare relative distances between a head item $\boldsymbol{x}_h$ to two alternates $\boldsymbol{x}_i, \boldsymbol{x}_j$. As an example, items may be images of products sold in an online marketplace and the features $\boldsymbol{x}_i$ could either be constructed from metadata about each product or extracted automatically from the image via a neural network. As human judgments are complex and involve higher order interactions of features, we seek a sufficiently expressive family of distance metrics to model these judgments. Hence we consider learning a nonlinear metric represented with a kernelized setting.

In the special case of a linear kernel, it corresponds to learning the Mahalanobis metric represented by a positive semidefinite matrix $\mathbf{M}$. As $\mathbf{M}$ is positive semidefinite, we can write $\mathbf{M} = \mathbf{L}^T\mathbf{L}$ using the Cholesky decomposition. Thus, learning the positive semidefinite matrix $\mathbf{M}$ can be also cast as learning the linear transformation $\mathbf{L}$ such that the distances are interpreted as Euclidean distances between points transformed by the matrix $\mathbf{L}$. Our work extends this to the kernelized scenario. We focus on learning a linear metric on a reproducing kernel Hilbert space (RKHS) in this work.

We assume that we have access to a feature map ϕ that maps

Proceedings of the 41st Conference on Uncertainty in Artificial Intelligence (UAI 2025), PMLR 244:4188–4207.

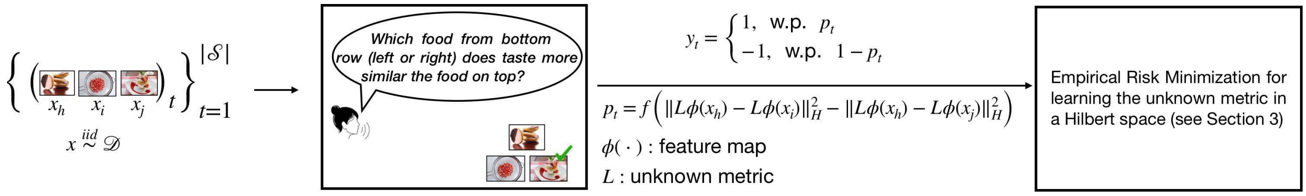
$\left\{ \left(x_h \; x_i \; x_j \right)_t \right\}_{t=1}^{|\mathcal{S}|}$
$x \overset{iid}{\sim} \mathscr{D}$
Which food from bottom row (left or right) does taste more similar the food on top?
$y_t = \begin{cases} 1, & \text{w.p. } p_t \\ -1, & \text{w.p. } 1 - p_t \end{cases}$
$p_t = f\left(\|L\phi(x_h) - L\phi(x_i)\|_H^2 - \|L\phi(x_h) - L\phi(x_j)\|_H^2 \right)$
$\phi(\cdot)$: feature map
L : unknown metric
Empirical Risk Minimization for learning the unknown metric in a Hilbert space (see Section 3)

Figure 1: Metric Learning from triplet comparisons (example triplets from Food-100 dataset [Wilber et al., 2014]). $\mathcal{S}$ is the set of triplets and y_t is the label collected from human for each triplet t.

from $\mathbb{R}^d$ to a real reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ such that $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\|\phi(\boldsymbol{x})\|_{\mathcal{H}} = \sqrt{k(\boldsymbol{x}, \boldsymbol{x})}$ for a known kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^1$. Therefore, $k(\cdot, \cdot)$ satisfies the reproducing property that $\langle f, k(\cdot, \boldsymbol{x}) \rangle = f(\boldsymbol{x})$ for any $f \in \mathcal{H}$ and $\boldsymbol{x} \in \mathbb{R}^d$. Then for any bounded linear operator $L : \mathcal{H} \to \mathcal{H}$, we define an associated nonlinear Mahalanobis metric, d_L, as

$$\begin{aligned} d_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j) &= \|L\phi(\boldsymbol{x}_i) - L\phi(\boldsymbol{x}_j)\|_{\mathcal{H}}^2 \\ &= \langle L\phi(\boldsymbol{x}_i) - L\phi(\boldsymbol{x}_j), L\phi(\boldsymbol{x}_i) - L\phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}}. \end{aligned}$$

For simplicity, we use ϕ_i for $\phi(\boldsymbol{x}_i)$ for the rest of the paper. With the kernelized metric setting, we can write triplet queries as

$$\operatorname{sign}\left(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2 \right).$$

This paper advances the understanding of the empirically powerful tasks of nonlinear metric learning via two core theoretical contributions:

- We establish the first generalization error and sample complexity guarantees for kernelized metric learning from triplet comparisons.
- We provide insights into how regularization affects the sample complexity and generalization bounds for kernelized metric learning from triplet comparisons.

As a byproduct, our analysis extends the results of the linear metric learning setting of Mason et al. [2017], overcoming its limited applicability, which required the number of items n to be larger than the dimensionality d.

1.1 RELATED WORK

Metric learning (also known as distance learning) has gained significant interest due to its power of effectively learning similarities and dissimilarities among objects. Here, we summarize most relevant contributions from the rich literature on metric learning. Kulis et al. [2013], Bellet et al. [2015] provide comprehensive summaries of the literature on classical techniques. In this paper, our focus is a specific type of query known as triplet comparisons. There exist methods and efficient algorithms for a variety of feedback such as class labels [Weinberger and Saul, 2009, Davis et al., 2007], triplet comparisons [Schultz and Joachims, 2003, Mason et al., 2017], perceptual adjustment queries [Xu et al., 2024] and nearest neighbor queries [Nadagouda et al., 2023]. A recent study [Tatli and Vinayak, 2024] uses triplet comparison queries to perform metric clustering, enabling the discovery of latent subgroups within the population before proceeding to metric learning from triplet comparisons.

Verma and Branson [2015] provide sample complexity of Mahalanobis distance learning from class labels, which is also known as linear metric learning, where the metric is parametrized by a positive semidefinite matrix. Mason et al. [2017], Ye et al. [2019] present tight generalization error bounds for Mahalanobis distance metric learning from triplet comparisons. Recently, there has been increased interest in nonlinear metric learning to better fit complex, real-world data sources. Kernelized approaches to the metric learning, similar to the setting considered in this work, are proposed by Martinel et al. [2015], Liu et al. [2021a], Wang et al. [2011], Chatpatanasiri et al. [2010], Kleindessner and von Luxburg [2017] and many others. More generally, the nonlinear variant has received attention through the study of deep Siamese networks [Guo et al., 2017].

Recent interest in using deep learning to extract useful representations from data is followed by triplet network models and its variations [Hoffer and Ailon, 2015]. Kaya and Bilge [2019] provide a comprehensive survey on deep metric learning. Despite the empirical success and popularity of deep metric learning techniques on metric learning, theoretical advancements in this area remain sparse. Zhou et al. [2024] provides a generalization analysis with deep ReLU networks for metric learning using the hinge loss. Other studies provide generalization guarantees for deep metric learning using neural tangent kernel [Liu et al., 2021a] and using Rademacher complexity based analysis [Huai et al., 2019].

Another line of work focuses on metric learning from pairwise comparisons. Pairwise comparisons can be viewed as a variation of triplet comparisons when it is assumed that there is a reference point u (responder) substituting for leading item $\boldsymbol{x}_h$. How to infer preferences from pairwise comparisons is a well studied problem in a diverse set of areas including machine learning, social choice theory, psychology, social sciences and political science (see the work of Fürnkranz and Hüllermeier [2010] for a comprehensive summary). Xu and Davenport [2020] uses a passive algorithm to learn a linear metric and preferences. This can be seen as simultaneously performing metric and preference learning. Later, Canal et al. [2022] extend the results to learning multiple preference points with a shared metric and provide theoretical guarantees for this task. Wang et al. [2024] analyzes linear metric learning problem with limited pairwise comparisons per user. Chen et al. [2024] proposes leveraging preference structure to reduce sample complexity of pairwise comparisons.

2 PROBLEM SETTING

Let objects be represented by the points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots$, where each $\boldsymbol{x}_i$ is drawn from the distribution $\mathcal{D}'$. In the noiseless setting, we are given a set of triplet comparisons in the form of

$$\operatorname{sign}(\operatorname{dist}^2(\boldsymbol{x}_h, \boldsymbol{x}_i) - \operatorname{dist}^2(\boldsymbol{x}_h, \boldsymbol{x}_j)).$$

We are interested in providing a theoretical understanding on the problem of learning kernelized Mahalanobis metric from triplet comparison queries. Our work extends the learning theoretic results of Mason et al. [2017] for linear metric learning to more general nonlinear metrics.

Let $\mathcal{S}$ denote the set of triplets generated from random triples $t = \{\boldsymbol{x}_h, \boldsymbol{x}_i, \boldsymbol{x}_j\}$, where each triple is independent and randomly chosen from the distribution $\mathcal{D}$, i.e., given that $\boldsymbol{x}_i \sim \mathcal{D}'$, each triple $t_{\{h,i,j\}} \in \mathcal{S}$ is randomly sampled from the stacked distribution $\mathcal{D}$. Therefore, the total number of objects is $3|\mathcal{S}|$ for $|\mathcal{S}|$ triplets in the general case. For each random triplet $t_{\{h,i,j\}}$, we observe a possibly noisy answer $y_t \in \{\pm 1\}$, which is an indication of $\operatorname{sign}\left(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2\right)$. Specifically, we assume that there exists an unknown kernelized metric that is consistent with the data and classifies any triplet t correctly with a probability greater than $1/2$ where this probability is taken with respect to any randomness in y_t and may depend on the specific triplet t. This is a common practical assumption when working with human judgment that some queries are inherently more noisy than others [Coombs, 1964, Rau et al., 2016]. We further assume that the y_t's are statistically independent. Our goal is to learn a metric parameterized by a linear map L that predicts triplets well on average. Namely, we seek an L that minimizes the misclassification probability:

$$\Pr\left(y_t \neq \operatorname{sign}\left(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2\right)\right). \quad (1)$$

Note that (1) is equal to the expected $0/1$ loss. In practice, minimizing $0/1$-loss is intractable and the above objective is relaxed to minimizing the true risk, which is defined below:

$$R(L) := \quad (2)$$
$$\mathbb{E}_{t\sim\mathcal{D}, y_t\in\{\pm1\}}[l(y_t(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2))],$$

for an arbitrary convex and α-Lipschitz loss $\ell : \mathbb{R} \to \mathbb{R}_{\geq 0}$, where the expectation is over random triplet coming from a distribution $\mathcal{D}$ and binary random label y_t conditioned on t, where $t = \{\boldsymbol{x}_h, \boldsymbol{x}_i, \boldsymbol{x}_j\}$ and $\{\boldsymbol{x}_h, \boldsymbol{x}_i, \boldsymbol{x}_j\} \sim \mathcal{D}$. If ℓ is chosen to upper bound the $0/1$-loss (e.g., the hinge loss $\ell(z) = \max(1-z, 0)$ or the logistic loss $\ell(z) = \log(1 + \exp^{-z})$), then $R(L)$ upper bounds the misclassification probability.

Unfortunately, we cannot minimize $R(L)$ directly as the joint distribution of (t, y_t) is unknown. Instead, given a set of triplets $\mathcal{S}$ and their labels y_t, we wish to learn a kernelized metric parameterized by a bounded linear map $L : \mathcal{H} \to \mathcal{H}$ that predicts triplets as well as possible on the observed data.

$$\widehat{R}_{\mathcal{S}}(L) := \quad (3)$$
$$\frac{1}{|\mathcal{S}|} \sum_{(t,y_t)\in\mathcal{S}} l(y_t(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2)).$$

We refer to $\widehat{R}_{\mathcal{S}}(L)$ as the empirical risk as it is an unbiased estimator of the true risk $R(L)$. For any given ℓ, we wish to answer three questions:

1. Regularizing a norm on L controls the flexibility of the metric and hence the model's predictions. What is the appropriate way to regularize to balance the bias-variance tradeoff of metric learning?
2. What can we guarantee about the generalization performance of the solution to (3) and how does this depend on the norm we choose to regularize on L?
3. As written, (3) is a potentially infinite dimensional, nonconvex optimization problem. How can it be made computationally tractable?

We refer to Section 3.1 for the first and second questions, and Section 4 for the last question.

3 KERNELIZED METRIC LEARNING

Traditional Mahalanobis distance metric learning is equivalent to learning a linear mapping of the data such that Euclidean distance in the mapped space agrees with a set of labels, such as class labels or triplet comparisons. Often, we are interested in a richer set of mappings than linear ones. Indeed, this is the idea that underlies deep learning and kernel learning. In this section, we provide the first theoretical study of nonlinear metric learning in an RKHS from triplet data, extending the linear results of Mason et al. [2017].

3.1 THEORETICAL GUARANTEES FOR KERNELIZED METRIC LEARNING

Frequently in optimization and learning theory, we wish to characterize *model classes of functions*– classes of metrics on $\mathcal{H}$ in this case. This is important to define optimal performance within a class for theoretical results and has tight connections to regularization in optimization which is used to prevent overfitting the data and ensure good generalization performance. We define model classes of kernelized Mahalanobis metrics by bounding the Schatten$-p$ norm of their map L (e.g., all kernelized metrics with a map L such that $\|L\|_{S_p} \leq \lambda$). For a compact, bounded linear operator T, its Schatten $p-$norm is defined to be $\|T\|_{S_p} := \left(\sum_{i\geq 1} s_i\,(T)^p\right)^{1/p}$ where $s_i(T)$ is the i^{th} singular value of T and may be equivalently written as $\sqrt{\lambda_i(T^\dagger T)}$ where † denotes the conjugate transpose and $\lambda_i(T^\dagger T)$ is the i^{th} eigenvalue of the Hermitian operator. We focus on two particular Schatten norms. First we consider the Schatten 2-norm which is a Hilbert-Schmidt norm. Specifically, we restrict solutions L to (3) to additionally satisfy $\|L^\dagger L\|_{S_2} \leq \lambda_F$ for a given $\lambda_F > 0$. Furthermore, we consider the Schatten 1-norm, also referred to as the trace or nuclear norm. In this setting, we assume that $\|L^\dagger L\|_{S_1} \leq \lambda_*$ and again restrict solutions to satisfy this constraint.

We define the optimal (possibly) infinite dimensional operator L^* as the minimizer of following optimization:

$$\begin{aligned} \min_{L} \quad & R(L) \\ \text{s.t.} \quad & \|L^\dagger L\|_{S_2} \leq \lambda_F. \end{aligned} \tag{P1}$$

Similarly we define $\widehat{L}$ as the solution to the optimization problem (P2) given below, i.e., the empirical risk minimizer:

$$\begin{aligned} \min_{L} \quad & \widehat{R}_{\mathcal{S}}(L) \\ \text{s.t.} \quad & \|L^\dagger L\|_{S_2} \leq \lambda_F. \end{aligned} \tag{P2}$$

Suppose that $\mathcal{S}_{\mathcal{X}} \subset \mathcal{H}$ represents the subspace spanned by the set $\{\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \ldots \phi(\boldsymbol{x}_n)\}$ corresponding to the features of random observations. Furthermore, let the potentially infinite dimensional linear operator $\widehat{L}_0$ denote the solution to (P3), obtained from random observations and the associated kernel features, where the norm constraint is imposed solely on the component of L whose domain lies within the span of features, i.e., denoted by $\mathcal{S}_{\mathcal{X}}$:

$$\begin{aligned} \min_{L} \quad & \widehat{R}_{\mathcal{S}}(L) \\ \text{s.t.} \quad & \|\mathcal{P}_{\mathcal{S}_{\mathcal{X}}}^\dagger L^\dagger L \mathcal{P}_{\mathcal{S}_{\mathcal{X}}}\|_{S_2} \leq \lambda_F, \end{aligned} \tag{P3}$$

where $\mathcal{P}_{\mathcal{S}_{\mathcal{X}}}$ denotes the projection onto $\mathcal{S}_{\mathcal{X}}$.

Remark 1. *Assume $\widehat{L}_0$ denotes the solution of (P3) whose domain is restricted to the span of features, i.e., $\mathcal{S}_{\mathcal{X}}$. This is a reasonable assumption, because $L_0\mathcal{P}_{\mathcal{S}_{\mathcal{X}}}$ also optimally solves (P3) for any solution L_0. Therefore, optimizing (P3) can be interpreted as seeking such an $\widehat{L}_0$.*

Lemma 1. *Recall that for a compact, bounded linear operator T, its Schatten p-norm is denoted as $\|T\|_{S_p}$. We have, for $p \geq 1$,*

$$\|\mathcal{P}_{\mathcal{S}_{\mathcal{X}}}^\dagger L^\dagger L \mathcal{P}_{\mathcal{S}_{\mathcal{X}}}\|_{S_p} \leq \|L^\dagger L\|_{S_p}.$$

Note that Schatten $2-$norm is the Hilbert-Schmidt norm.

Lemma 1 allows us to establish a relation between solutions of (P2) and (P3), explained in Proposition 1. Note that optimization settings (P2) and (P3) have the same objective function. The distinction lies in the the norm constraint $\|\cdot\|_{S_2}$ imposed on L. In (P3), the constraint applies only to the component of L whose domain is restricted to the span of features, denoted by $\mathcal{S}_{\mathcal{X}}$. Consequently, solving (P3) for an operator $\widehat{L}_0$, as in Remark 1, corresponds to minimizing the empirical risk in (P2) under the additional constraint that the search is restricted to the span $\mathcal{S}_{\mathcal{X}}$.

Proposition 1. *We observe that $\widehat{L}_0$ is in the solution set of (P2). More precisely, any L within the feasible set of (P2) is an optimal solution, provided that $L\mathcal{P}_{\mathcal{S}_{\mathcal{X}}} = \widehat{L}_0$. As a result, (P2) and (P3) have the same optimal value, i.e.,*

$$\widehat{R}_{\mathcal{S}}(\widehat{L}) = \widehat{R}_{\mathcal{S}}(\widehat{L}_0).$$

Therefore, optimizing the empirical risk in (P2) with a search restricted to $\mathcal{S}_{\mathcal{X}}$ suffices to assign optimal value for (P2).

Recall that we wish to learn a kernelized metric parametrized by a bounded linear map $L : \mathcal{H} \to \mathcal{H}$ that predicts triplets effectively based on random observations. We establish a bound on the generalization error of $\widehat{L}_0$, which is a solution to the empirical risk minimization. Note that $\widehat{L}_0$ solves both (P3) and (P2). We compare it to the optimal infinite dimensional operator L^*, which minimizes the true risk.

The following theorem demonstrates that, with a sufficiently large set of triplets $\mathcal{S}$, the performance of $\widehat{L}_0$ is nearly as good as that of L^*.

Theorem 1. *Fix $\delta, \lambda_F > 0$ and let ℓ be α-Lipschitz. Assume $\|\phi(\boldsymbol{x})\|_{\mathcal{H}} \leq B$ for any $\boldsymbol{x}$. Then, with probability at least $1 - \delta$,*

$$R(\widehat{L}_0) - R(L^*) \leq 4\alpha B^2 \lambda_F \sqrt{\frac{6}{|\mathcal{S}|}} + 12\alpha B^2 \lambda_F \sqrt{\frac{2\ln 2/\delta}{|\mathcal{S}|}}$$

For any loss $\ell(\cdot)$ which upper bounds the $0/1-$loss, such as the logistic or hinge losses, the left hand side is an upper bound on the expected prediction accuracy for predicting triplets. Hence, the above result also provides a generalization error guarantee for prediction accuracy.

To further interpret the result of Theorem 1, consider the case of a linear kernel where the points used to generate triplets live in the unit ball in $\mathbb{R}^d$. In this case, one can directly learn $L^T L = \mathbf{M} \in \mathbb{R}^{d\times d}$. Setting $\lambda_F = O(d)$, which is sufficient to ensure that the average entry of $\mathbf{M}$ is dimensionless, Theorem 1 shows that sampling $O(d^2 \log(1/\delta))$ triplets is sufficient to ensure good generalization. As the number of degrees of freedom for a $d \times d$ matrix is d^2, this matches intuition that the sample complexity should scale with degrees of freedom. In general, $\|L^\dagger L\|_{S_2}$ behaves like a notion of the effective dimensionality d_{eff} of

L [Zhang, 2005]. Indeed, if fewer eigenvalues of $L^\dagger L$ are large, then λ_F is smaller and the space is nearly low dimensional. Hence, we may interpret Theorem 1 as suggesting a sample complexity of $O(d_{\text{eff}}^2 \log(1/\delta))$.

Next, we bound the excess risk under the constraint $\|L^\dagger L\|_{S_1} \leq \lambda_*$. Specifically, consider the optimization problems (P1), (P2) and (P3), now with Schatten 1-norm constraints of the form $\|\cdot\|_{S_1} \leq \lambda_*$. Let L_n^*, $\widehat{L}_n$ and $\widehat{L}_{n_0}$ denote the solutions to the modified versions of problems (P1), (P2) and (P3), respectively, where the Schatten 1-norm constraints replace the Schatten 2-norm constraints.

The following theorem establishes a bound on the generalization error of $\widehat{L}_{n_0}$ by comparing it to the true risk minimizer L_n^*.

Theorem 2. *Fix $\delta, \lambda_* > 0$ and let ℓ be α-Lipschitz. Assume $\|\phi(\boldsymbol{x})\|_{\mathcal{H}} \leq B$ for any $\boldsymbol{x}$. Then, with probability at least $1-\delta$,*

$$R(\widehat{L}_{n_0}) - R(L_n^*) \leq 4\alpha\lambda_* \left(B^2 \sqrt{12\frac{\log 3|\mathcal{S}|}{|\mathcal{S}|}} + \frac{2\log 3|\mathcal{S}|}{|\mathcal{S}|} \right) + 12\alpha B^2 \lambda_* \sqrt{\frac{2\ln 2/\delta}{|\mathcal{S}|}},$$

where $\mathcal{S}$ is the set of triplets chosen and $|\mathcal{S}|$ represents the size of this set. Note that restricting the Schatten-1 norm encourages solution L (and correspondingly operationalized version $\mathbf{M}$ (see Section 4.2)) to have low rank. This corresponds to learning a low-dimensional metric over data. This is reasonable in settings where though the ambient dimension of data is large, one expects that the triplet comparisons are well explained by a projection of the data points onto a low dimensional space $\mathcal{S}^o$. As an example, consider ϕ corresponding to a polynomial kernel of degree 2: $\phi(\boldsymbol{x}) = [\boldsymbol{x}_1^2, \boldsymbol{x}_1 \cdot \boldsymbol{x}_2, \dots, \boldsymbol{x}_2^2, \boldsymbol{x}_2 \cdot \boldsymbol{x}_3, \dots, \boldsymbol{x}_d^2]^T$ for $\boldsymbol{x} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_d]^T$. Suppose the data is generated according to a true map L^* which is a projection onto $\mathcal{S}^o$, the span of a sparse subset of $k \ll d^2$ monomials. Then, taking $\lambda_* = \|L^\dagger L\|_{S_1} = k$, Theorem 2 guarantees that sampling $O(k^2 \log(k/\delta))$ triplets is sufficient. By contrast, if L was the identity map on degree 2 polynomials, the same result would suggest a sample complexity of $O(d^4 \log(d/\delta))$ which is much larger. Hence, this result is especially powerful for low or approximately low dimensional metrics.

4 PRACTICAL IMPLEMENTATION

In Section 3, we show that solving (P2) with a search restricted to $\mathcal{S}_{\mathcal{X}}$, i.e., solving for $\widehat{L}_0$ in (P3), presents a solution for both (P2) and (P3). We bound the generalization error based on $\widehat{L}_0$ (see Theorems 1 and 2). Our goal in this part is to solve (P3) to learn $\widehat{L}_0$, which is a nonlinear Mahalanobis metric. Note that in addition to being possibly infinite dimensional, the optimization (P3) is also nonconvex. In this section, we carefully demonstrate how to learn $\widehat{L}_0$ from a random set of independent triplets $\mathcal{S}$ with associated labels y_t via convex optimization. We show that solving $(P3)$ is equivalent to solving a finite dimensional convex optimization problem. We use a representer theorem (see Proposition 2) to reduce finding $\widehat{L}_0$ to an optimization over finite dimensional vectors. We use the idea of Kernelized Principle Component Analysis (KPCA) to compute all distances using KPCA vectors $\varphi_1, \varphi_2, \dots, \varphi_n \in \mathbb{R}^n$ and reduce the problem to learning an $n-$dimensional metric parameterized by a semidefinite matrix denoted $\mathbf{M}$:

$$\widehat{\widehat{R}}_{\mathcal{S}}(\mathbf{M}) := \frac{1}{|\mathcal{S}|} \sum_{(t,y_t)\in\mathcal{S}} l(y_t(\|\varphi_h - \varphi_i\|_{\mathbf{M}}^2 - \|\varphi_h - \varphi_j\|_{\mathbf{M}}^2)) \tag{4}$$

where $n = 3|\mathcal{S}|$ and $\varphi_i \in \mathbb{R}^n$ denotes the KPCA representation of feature $\phi_i \in \mathcal{H}$ for the random set $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \dots \phi(\boldsymbol{x}_n)$. We refer to the quantity $\widehat{\widehat{R}}_{\mathcal{S}}(\mathbf{M})$ as the (finite dimensional) empirical risk of $\mathbf{M}$. We can express $\widehat{L}_0$ using the solution of (finite dimensional) empirical risk minimization with corresponding constraints. In Section 4.1 we use known results to explain how to perform KPCA, how to calculate distances with finite dimensional vectors in KPCA and how to relate norm constraints over L with finite dimensional metric $\mathbf{M}$. Then, in Section 4.2, we provide the finite dimensional optimization with all constraints that is equivalent to (P3) and express $\widehat{L}_0$ from its solution.

4.1 KERNELIZED PRINCIPLE COMPONENT ANALYSIS (KPCA)

In this part, we explain how to perform kernelized PCA in a reproducing kernel Hilbert space (RKHS). Consider the set of items $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3 \dots \boldsymbol{x}_n \in \mathbb{R}^d$ and corresponding features $\phi_1, \phi_2, \dots \phi_n$. We assume that ϕ_i's are linearly independent. Recall that $\mathcal{S}_{\mathcal{X}} \subset \mathcal{H}$ represents the subspace spanned by $\{\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \dots \phi(\boldsymbol{x}_n)\}$. Let $\psi_1, \psi_2, \dots \psi_n$ be the n principal component directions in this space. We show how to efficiently compute projections onto this subspace using the idea of Kernelized Principle Component Analysis (KPCA). This is important as the principal components live in the possibly infinite dimensional space $\mathcal{H}$ making traditional optimization either intractable or impossible. The following procedure, which we summarize for completeness from Chatpatanasiri et al. [2010] can be used to compute the projection of any point $\boldsymbol{x} \in \mathbb{R}^d$ onto the principal component directions in time that is polynomial in $n = 3|\mathcal{S}|$:

1. Form the Gram matrix: $\mathbf{K} \in \mathbb{R}^{n\times n}$ such that $\mathbf{K}_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$.
2. Center the Gram matrix: $\overline{\mathbf{K}} = \mathbf{K} - \frac{1}{n}\mathbf{1}_{n\times n}\mathbf{K} - \frac{1}{n}\mathbf{K}\mathbf{1}_{n\times n} + \frac{1}{n^2}\mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times n}$, where $\mathbf{1}_{n\times n}$ is the n by n matrix of all ones.
3. Compute all n eigenvectors of $\overline{\mathbf{K}}$, $\alpha_1, \dots, \alpha_n$ and form matrix $\mathbf{A} = [\alpha_1, \dots, \alpha_n]$.

4. For any $\boldsymbol{x} \in \mathbb{R}^d$ and any principal component ψ_j with eigenvector α_j, we have that $\langle \phi(\boldsymbol{x}), \psi_j \rangle_{\mathcal{H}} = \sum_{i=1}^{n} \alpha_{i,j} k(\boldsymbol{x}, \boldsymbol{x}_i)$.
5. Therefore, for any $\boldsymbol{x} \in \mathbb{R}^d$ we may represent $\phi(\boldsymbol{x})$ in terms of its projection onto $\psi_1, \dots, \psi_n$ as

$$\varphi(\boldsymbol{x}) = \mathbf{A}^T [k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_n)]^T$$

For the remainder, we will let $\varphi_i \in \mathbb{R}^n$ denote the KPCA representation of random feature $\phi_i \in \mathcal{H}$ for the set $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \dots \phi(\boldsymbol{x}_n)$. The following representer theorem demonstrates that we may instead use finite dimensional vectors $\varphi_1, \dots, \varphi_n$ for the optimization without loss in performance for a given set $\phi(\boldsymbol{x}_1), \phi(\boldsymbol{x}_2), \dots \phi(\boldsymbol{x}_n)$.

Proposition 2. *(Theorem 1 of Chatpatanasiri et al. [2010]) Let $\{\overline{\psi}_i\}_{i=1}^n$ be any set of points in $\mathcal{H}$ such that Span$\big(\{\overline{\psi}_i\}_{i=1}^n\big) = \mathcal{S}_\mathcal{X}$ and let $\mathcal{H}'$ be a Hilbert space such that $\mathcal{H}$ and $\mathcal{H}'$ are separable. For any objective function f, the optimization*

$$\min_{L} f\left(\{\langle L\phi_i, L\phi_j \rangle_{\mathcal{H}'}\}_{i,j \in [n]}\right)$$

such that $L : \mathcal{H} \to \mathcal{H}'$ is a bounded linear map, has the same optimal value as

$$\min_{L' \in \mathbb{R}^{n \times n}} f\left(\{\overline{\psi}(\boldsymbol{x}_i)^T {L'}^T L' \overline{\psi}(\boldsymbol{x}_j)\}_{i,j \in [n]}\right)$$

where $\overline{\psi}(\boldsymbol{x}) = [\langle \phi(\boldsymbol{x}), \overline{\psi}_1 \rangle, \dots, \langle \phi(\boldsymbol{x}), \overline{\psi}_n \rangle]^T \in \mathbb{R}^n$.

Calculating Kernelized Mahalanobis Distances using KPCA: Proposition 2 provides that one can learn $\widehat{L}_0$ using the KPCA representations of $\boldsymbol{x}_1, \boldsymbol{x}_2 \dots \boldsymbol{x}_n$. To be precise, given a linear map $L : \mathcal{H} \to \mathcal{H}$, we may expand the distance $\|L\phi_i - L\phi_j\|^2 = \langle L\phi_i, L\phi_i \rangle - 2\langle L\phi_i, L\phi_j \rangle + \langle L\phi_j, L\phi_j \rangle$. Let $\mathbf{A}$ be as defined in kernelized PCA and $\Phi := [\phi_1, \phi_2, \dots \phi_n]$, the matrix whose columns are ϕ_i's. As the ϕ_i's are linearly independent, Φ is full rank[1]. For any ϕ_k within the set $\{\phi_1, \phi_2, \dots \phi_n\}$, we have $L\phi_k = \mathbf{U}\mathbf{A}^T \Phi^T \phi_k$ for a linear map $\mathbf{U}$ from $\mathbb{R}^n$ to $\mathcal{H}$. Additionally, by definition of the kernel function $k(\cdot, \cdot)$, $\Phi^T \phi(\boldsymbol{x}_k) = [k(\boldsymbol{x}_k, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}_k, \boldsymbol{x}_n)]^T$. Hence,

$$\begin{aligned} \|L\phi_i - L\phi_j\|_{\mathcal{H}}^2 &= \langle \mathbf{U}\varphi_i, \mathbf{U}\varphi_i \rangle - 2\langle \mathbf{U}\varphi_i, \mathbf{U}\varphi_j \rangle \\ &+ \langle \mathbf{U}\varphi_j, \mathbf{U}\varphi_j \rangle \\ &= \|\mathbf{U}\varphi_i - \mathbf{U}\varphi_j\|^2 \\ &= \|\varphi_i - \varphi_j\|_{\mathbf{M}}^2 \end{aligned}$$

for $\varphi_i \in \mathbf{R}^n$ defined by kernelized PCA on $\phi_1, \phi_2, \dots \phi_n$, and $\mathbf{M} = \mathbf{U}^T \mathbf{U} \in \mathbb{R}^{n \times n}$. Therefore, we may use kernelized PCA to efficiently compute distances in $\mathbb{R}^n$ as opposed to in $\mathcal{H}$ for a given set $\phi_1, \phi_2, \dots \phi_n$.

Relating norms in $\mathcal{H}$ and $\mathbb{R}^n$: Above lines demonstrate how, for a given L, we may find a specific $\mathbf{M}$ that defines a metric on $\mathbb{R}^n$ which computes distances between points in $\mathcal{S}_\mathcal{X} \subset \mathcal{H}$ equally to L using the KPCA basis for $\mathcal{S}_\mathcal{X}$.

We consider Schatten norm constraints on L to rigorously define model classes for L in (P1), (P2) and (P3). Hence, it is necessary to relate the Schatten norms of L to Schatten norms of $\mathbf{M}$ so that constraints placed on L in (P3) are comparable to those placed on $\mathbf{M}$ in $\mathbb{R}^n$. Following Lemma relate these norms.

Lemma 2. *Let $\phi_1, \dots, \phi_n$ be a set of features corresponding to a random set of triplets and $\mathcal{S}_\mathcal{X}$ is the span of feature points. For any $\phi_{\boldsymbol{x}} \in \mathcal{S}_\mathcal{X}$ and $L : \mathcal{H} \to \mathcal{H}$, there exists a semidefinite matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ such that $\|L\phi_{\boldsymbol{x}}\|_{\mathcal{H}} = \|\varphi_{\boldsymbol{x}}\|_{\mathbf{M}}$ and $\|\mathcal{P}^{\dagger}_{\mathcal{S}_\mathcal{X}} L^\dagger L \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_p} = \|\mathbf{M}\|_p \; \forall p$.*

Now, we can set up the finite dimensional optimization problem to learn a finite dimensional metric $\mathbf{M}$ that will enable us to find $\widehat{L}_0$.

4.2 LEARNING KERNELIZED METRICS IN PRACTICE

We define following finite dimensional constrained convex program to learn a kernelized Mahalanobis metric from a random set of triplets $\mathcal{S}$:

$$\begin{aligned} \min_{\mathbf{M} \succeq 0} \quad & \widehat{\widehat{R}}_{\mathcal{S}}(\mathbf{M}) \\ \text{s.t.} \quad & \|\mathbf{M}\|_F \leq \lambda_F \end{aligned} \tag{P4}$$

where $\mathbf{M} \succeq 0$ denotes that $\mathbf{M}$ is positive semidefinite and the condition on the norm prevents overfitting as in (P1), (P2) and (P3). Let $\widehat{\mathbf{M}}$ denote an optimal solution to (P4) referred as the empirical risk minimizer. Likewise, if we instead consider $\|\mathcal{P}^{\dagger}_{\mathcal{S}_\mathcal{X}} L^\dagger L \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \leq \lambda_*$, this is corresponding to $\|\mathbf{M}\|_* \leq \lambda_*$ where $\|\cdot\|_*$ denotes the nuclear norm. In this setting, we may likewise solve for $\widehat{\mathbf{M}}$ satisfying this constraint instead. Below, Proposition 3 presents the relation between $(P3)$ and $(P4)$. Then, we show how to obtain $\widehat{L}_0$ from the finite dimensional solution.

Proposition 3. *Optimization problems (P4) and (P3) are equivalent. Solving $(P4)$ is equal to learning $\widehat{L}_0$. Likewise, $\widehat{L}_0$ can be considered as the Hilbert space counterpart of finite dimensional space operator $\widehat{\mathbf{M}}$. Furthermore, let $\Psi_1, \dots, \Psi_n \in \mathcal{H}$ be KPCA directions for the span $\mathcal{S}_\mathcal{X}$. We can write $\widehat{L}_0$ as*

$$\widehat{L}_0 : \widehat{L}_0 \phi_x = \sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} \Psi_i \otimes \Psi_j \mathcal{P}_{\mathcal{S}_\mathcal{X}} \phi_x \tag{5}$$

where $\Psi_i \otimes \Psi_j \phi_x = \langle \Psi_j, \phi_x \rangle_{\mathcal{H}} \Psi_i$ and $\mathbf{W} =$ Chol$(\widehat{\mathbf{M}})$ such that $\mathbf{W}\mathbf{W}^T = \widehat{\mathbf{M}}$, i.e., $\mathbf{W}$ is from Cholesky decomposition of $\widehat{\mathbf{M}}$.

[1] In the case where the ϕ_i's are not linearly independent and Φ is no longer full rank, KPCA can be modified by projecting onto the $k < n$ eigenvectors corresponding to the nonzero eigenvalues.

Kernel	Formula	Parameter
Linear	$k(x,y) = x^\top y$	N/A
Gaussian	$k(x,y) = e^{\frac{-\|x-y\|_2^2}{2\sigma^2}}$	σ
Sigmoid	$k(x,y) = \tanh\left(c + \alpha x^\top y\right)$	c, α
Polynomial	$k(x,y) = (c + x^\top y)^p$	c, p
Laplacian	$k(x,y) = e^{\alpha\|x-y\|_1}$	α

Table 1: List of kernel functions and parameters used in our simulations and experiments.

Proposition 3 allows us to operationalize (P3) with a finite dimensional convex optimization problem and express $\widehat{L}_0$ from its solution.

5 EXPERIMENTAL RESULTS

In this part, we present simulations and experiments on real datasets to validate our theoretical results. Table 1 presents the kernel functions and the kernel parameters that we used in simulations and experiments. In all of our simulations and experiments, we use CVXPY [Diamond and Boyd, 2016, Agrawal et al., 2018] and MOSEK [ApS, 2024] to solve the convex program (P4). We use the nuclear norm constraint for $\mathbf{M}$ in (P4).

To apply these kernel functions efficiently, especially on large datasets, we consider the computational complexity of the Kernelized Principal Component Analysis (KPCA) operation, which is $O(n^3)$, where n is the number of items used in queries. To mitigate this cost, one can adapt low-rank approximations of the Gram matrix (Nyström method [Reinhardt, 2012, Williams, 1998]) by randomly sampling $m \ll n$ items from n. The Nyström KPCA method [Williams and Seeger, 2000] has a complexity of $O(nm^2)$. Another approach, the randomly pivoted Cholesky algorithm [Chen et al., 2025], requires only $O(k^2 n)$ kernel evaluations for a rank-k approximation. In our work, we leverage the Nyström KPCA [Williams and Seeger, 2000] with $m = 500$ to efficiently approximate the Gram matrix.

5.1 SIMULATIONS

Generating Noisy Labels for a Known Distance Function: We assume an explicit link function $f(\cdot)$, where $f(\cdot)$ generates noisy labels for each triplet following that $y_t = -1$ with probability p_t as a noisy indication of $\text{sign}(d_L^2(\boldsymbol{x}_h, \boldsymbol{x}_i) - d_L^2(\boldsymbol{x}_i, \boldsymbol{x}_j))$, where

$$p_t = f\left(d_L^2(\boldsymbol{x}_h, \boldsymbol{x}_i) - d_L^2(\boldsymbol{x}_h, \boldsymbol{x}_j)\right).$$

We use $f(x) = 1/(1 + e^{\rho x})$ as the link function, where the parameter ρ controls the noise level. We first consider a spiral shape in 2D.

Spiral with Geodesic Distance: We generate triplets uniformly along the spiral. We assume the true distance function is the geodesic distance (see Figure 2) along the 2D curve. We provide train and test accuracy for different kernel functions with varying number of triplets. Figure 3 illustrates the performance of various kernels. We observe that polynomial, Gaussian, and Laplacian kernels outperform linear and sigmoid kernels. We defer the details of the simulation setting to the Appendix.

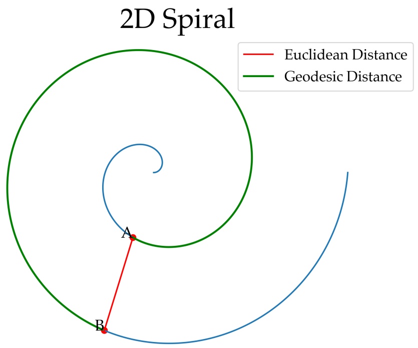

Figure 2: A 2D spiral. We sample triplets uniformly along this curve. The geodesic distance between point A and B is the length of the green curve, whereas the Euclidean distance between the two points is the length of the red line.

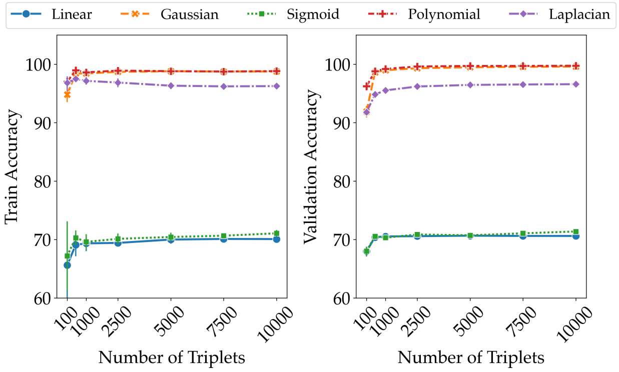

Figure 3: Performance of various kernels in the 2D spiral setting. For the Gaussian kernel, we use $\sigma = 2$; sigmoid kernel, $c = 1, \alpha = 1$; polynomial kernel, $c = 1, p = 2$, Laplacian kernel, $\alpha = 1$. For the link function f, we use $\rho = 30$ to set the noise level around 0.01. We repeat each run 50 times.

Next, we assume we have access to a feature map ϕ such that $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ with a Gaussian kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^1$, where $\sigma = 1$.

Gaussian Kernel Map: We assume there exists a linear functional $L^* : \mathcal{H} \to \mathcal{H}$ that lies on an $r-$dimensional manifold. In Figure 4, we provide our results with a Gaussian kernel for $r = 2$ (see the Appendix for details of data generation and more extensive results). We also defer the details of the simulation setting to the Appendix.

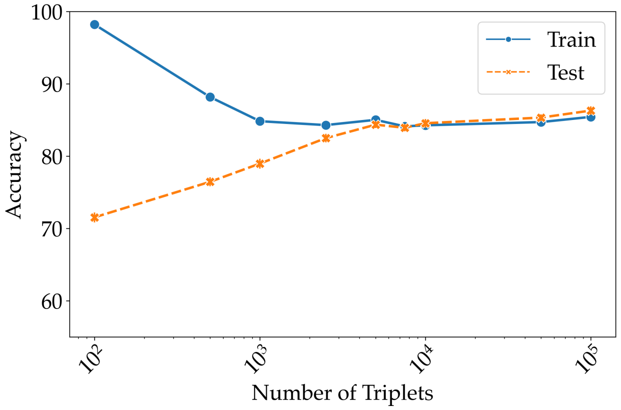

Figure 4: Train and test accuracy of Gaussian kernel. Here, we use $\sigma = 1$. For the link function f, we use $\rho = 1000$ to set the noise level around 0.05. We repeat each run 50 times.

The test accuracy increases as we have more triplets for training in Figure 4. We also observe that, as the number of triplets increases, the train and test accuracy gets close, consistent with our analysis in Theorems 1 and 2. Recall that excess risk decreases with more triplets according to Theorems 1 and 2.

5.2 EMPIRICAL EVALUATION: FOOD-100 DATASET

The Food-100 dataset [Wilber et al., 2014] consists of 100 food items and approximately 190,000 triplets based on human responses (See Figure 1 for example images from the dataset). We divide this dataset by items to ensure that the model does not encounter some items in the test and validation sets during the training phase. See Appendix for more information on how we split the dataset. We obtain embeddings for each item in Food-100 dataset using the embedding from the antepenultimate layer of AlexNet [Krizhevsky et al., 2012], pretrained on ImageNet [Deng et al., 2009]. We, then, project them to a 2D space using PaCMAP [Wang et al., 2021]. Figure 5 shows the performance of different kernels, among which the Gaussian kernel performs the best.

Theorems 1 and 2 provide bounds for excess risk. Therefore, our analysis allows us to bound the difference between the true risk and the empirical risk for any kernel choice. Experiments with different kernels demonstrate that train and test accuracies are close, indicating that the empirical risk approximates the true risk well. Choice of kernel has an effect on the true risk and therefore affects the risk achievable by the learned metric. This is reflected in the difference in test accuracies across different kernels. Since there is no way of knowing what the true risk is, cross-validation is an appropriate method for selecting the optimal kernel for the dataset at hand.

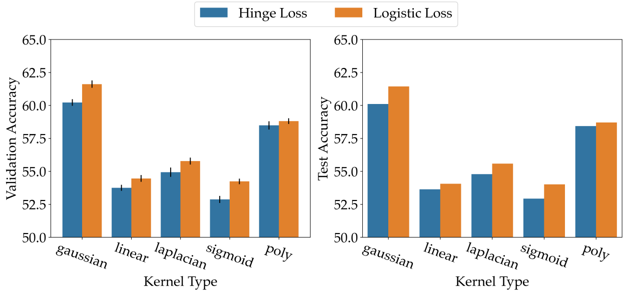

Figure 5: Performance of various kernels under the Food-100 dataset. For the Gaussian kernel, we use $\sigma = 2$; sigmoid kernel, $c = 1, \alpha = 0.01$; polynomial kernel, $c = 1, p = 2$, Laplacian kernel, $\alpha = 1$. We repeat the validation 20 times.

6 DISCUSSION

When undertaking the task of developing a theoretical understanding of triplet based nonlinear metric learning methods, the first natural setting to consider is kernelized metric learning. To the best of our knowledge, there are no generalization results analyzing the sample complexity for kernelized metric learning via triplet comparisons in the literature prior to our work. The theoretical foundations for metric learning via triplet queries are currently limited to linear settings, e.g., Mason et al. [2017], which provide generalization results for the linear setting when the set of items being queried is fixed and the number of items $n \gg d$ (See Appendix C for further explanation). Therefore, our work fills an important gap in the literature.

We provide a theoretical analysis for the kernelized metric learning problem. We provide novel generalization and sample complexity bounds. Developing an understanding of other nonlinear metric learning approaches, especially neural networks based approaches would be interesting for future research directions. That said, kernelized approaches are preferred in areas where interpretability and explainability are crucial, especially when they also perform nearly as well as other methods [Radhakrishnan et al., 2023]. Therefore, understanding kernelized settings is also of value beyond theoretical pursuit towards understanding a broader set of nonlinear approaches.

7 ACKNOWLEDGEMENTS

This work was partially supported by NSF grants NCS-FO 2219903 and NSF CAREER Award CCF 2238876.

References

Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1): 42–60, 2018.

MOSEK ApS. *The MOSEK optimization toolbox for Python*, 2024. URL `http://docs.mosek.com`.

Aurélien Bellet, Amaury Habrard, and Marc Sebban. *Metric learning*. Morgan & Claypool Publishers, 2015.

Gregory Canal, Blake Mason, Ramya Korlakai Vinayak, and Robert Nowak. One for all: Simultaneous metric and preference learning over multiple users. In *Advances in Neural Information Processing Systems*, volume 35, pages 4943–4956, 2022.

Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 2408–2415, 2013.

Ratthachat Chatpatanasiri, Teesid Korsrilabutr, Pasakorn Tangchanachaianan, and Boonserm Kijsirikul. A new kernelization framework for mahalanobis distance learning algorithms. *Neurocomputing*, 73(10-12):1570–1579, 2010.

Daiwei Chen, Yi Chen, Aniket Rege, and Ramya Korlakai Vinayak. Pal: Pluralistic alignment framework for learning from heterogeneous preferences. *arXiv preprint arXiv:2406.08469*, 2024.

Yifan Chen, Ethan N Epperly, Joel A Tropp, and Robert J Webber. Randomly pivoted cholesky: Practical approximation of a kernel matrix with few entry evaluations. *Communications on Pure and Applied Mathematics*, 78 (5):995–1041, 2025.

Clyde H Coombs. A theory of data, 1964.

Jason V Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pages 209–216, 2007.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning and ranking by pairwise comparison. In *Preference learning*, pages 65–82. Springer, 2010.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. Is that you? metric learning approaches for face identification. In *2009 IEEE 12th international conference on computer vision*, pages 498–505. IEEE, 2009.

Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 1763–1771, 2017.

Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *Similarity-based pattern recognition: third international workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*, pages 84–92. Springer, 2015.

Steven CH Hoi, Wei Liu, and Shih-Fu Chang. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 6(3):1–26, 2010.

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. Collaborative metric learning. In *Proceedings of the 26th international conference on world wide web*, pages 193–201, 2017.

Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Deep transfer metric learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 325–333, 2015.

Mengdi Huai, Hongfei Xue, Chenglin Miao, Liuyi Yao, Lu Su, Changyou Chen, and Aidong Zhang. Deep metric learning: The generalization analysis and an adaptive algorithm. In *IJCAI*, pages 2535–2541, 2019.

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

Matthäus Kleindessner and Ulrike von Luxburg. Kernel functions based on triplet comparisons. *Advances in neural information processing systems*, 30, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

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

Fanghui Liu, Xiaolin Huang, Yingyi Chen, and Johan Suykens. Fast learning in reproducing kernel krein spaces via signed measures. In *International Conference on Artificial Intelligence and Statistics*, pages 388–396. PMLR, 2021a.

Qi Liu, Wenhan Li, Zhiyuan Chen, and Bin Hua. Deep metric learning for image retrieval in smart city development. *Sustainable Cities and Society*, 73:103067, 2021b.

Niki Martinel, Christian Micheloni, and Gian Luca Foresti. Kernelized saliency-based person re-identification through multiple metric learning. *IEEE Transactions on Image Processing*, 24(12):5645–5658, 2015.

Blake Mason, Lalit Jain, and Robert Nowak. Learning low-dimensional metrics. *Advances in neural information processing systems*, 30, 2017.

Namrata Nadagouda, Austin Xu, and Mark A Davenport. Active metric learning and classification using similarity queries. In *Uncertainty in Artificial Intelligence*, pages 1478–1488. PMLR, 2023.

Adityanarayanan Radhakrishnan, Max Ruiz Luyten, Neha Prasad, and Caroline Uhler. Transfer learning with kernel methods. *Nature Communications*, 14(1):5570, 2023.

Martina A Rau, Blake Mason, and Robert Nowak. How to model implicit knowledge? similarity learning methods to assess perceptions of visual representations. *International Educational Data Mining Society*, 2016.

Hans-Jürgen Reinhardt. *Analysis of approximation methods for differential and integral equations*, volume 57. Springer Science & Business Media, 2012.

Brett D Roads and Michael C Mozer. Obtaining psychological embeddings through joint kernel and metric learning. *Behavior research methods*, 51:2180–2193, 2019.

Matthew Schultz and Thorsten Joachims. Learning a distance metric from relative comparisons. *Advances in neural information processing systems*, 16, 2003.

Gokcan Tatli and Ramya Korlakai Vinayak. Metric clustering from triplet comparisons. In *2024 60th Annual Allerton Conference on Communication, Control, and Computing*, pages 01–08. IEEE, 2024.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Nakul Verma and Kristin Branson. Sample complexity of learning mahalanobis distance metrics. *Advances in neural information processing systems*, 28, 2015.

Jun Wang, Adam Woznica, Alexandros Kalousis, et al. Metric learning with multiple kernels. *Advances in neural information processing systems*, 24, 2011.

Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021. URL `http://jmlr.org/papers/v22/20-1061.html`.

Zhi Wang, Geelon So, and Ramya Korlakai Vinayak. Metric learning from limited pairwise preference comparisons. *arXiv preprint arXiv:2403.19629*, 2024.

Kilian Q Weinberger and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of machine learning research*, 10(2), 2009.

Michael Wilber, Iljung Kwak, and Serge Belongie. Cost-effective hits for relative similarity comparisons. In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, volume 2, pages 227–233, 2014.

Christopher Williams and Matthias Seeger. Using the nyström method to speed up kernel machines. *Advances in neural information processing systems*, 13, 2000.

Christopher KI Williams. Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning in graphical models*, pages 599–621. Springer, 1998.

Hao Wu, Qimin Zhou, Rencan Nie, and Jinde Cao. Effective metric learning with co-occurrence embedding for collaborative recommendations. *Neural Networks*, 124: 308–318, 2020.

Austin Xu and Mark Davenport. Simultaneous preference and metric learning from paired comparisons. *Advances in Neural Information Processing Systems*, 33:454–465, 2020.

Austin Xu, Andrew McRae, Jingyan Wang, Mark Davenport, and Ashwin Pananjady. Perceptual adjustment queries and an inverted measurement paradigm for low-rank metric learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Xingxu Yao, Dongyu She, Haiwei Zhang, Jufeng Yang, Ming-Ming Cheng, and Liang Wang. Adaptive deep metric learning for affective image retrieval and classification. *IEEE Transactions on Multimedia*, 23:1640–1653, 2020.

Han-Jia Ye, De-Chuan Zhan, and Yuan Jiang. Fast generalization rates for distance metric learning: Improved theoretical analysis for smooth strongly convex distance metric learning. *Machine Learning*, 108:267–295, 2019.

Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Jake An. Next item recommendation with self-attentive metric learning. In *Thirty-third AAAI conference on artificial intelligence*, volume 9, 2019.

Tong Zhang. Learning bounds for kernel regression using effective data dimensionality. *Neural computation*, 17(9): 2077–2098, 2005.

Junyu Zhou, Puyu Wang, and Ding-Xuan Zhou. Generalization analysis with deep relu networks for metric and similarity learning. *arXiv preprint arXiv:2405.06415*, 2024.

Metric Learning in an RKHS
(Supplementary Material)

Gokcan Tatli[1] **Yi Chen**[1] **Blake Mason**[2] **Robert Nowak**[1] **Ramya Korlakai Vinayak**[1]

[1]Dept. of Electrical and Computer Engineering, University of Wisconsin-Madison, Madison, Wisconsin, USA
[2]Amazon.com, USA

A LIMITATIONS AND BROADER IMPACTS

The results of this paper are primarily theoretical and ideally a suggestion towards what practical guarantees can and cannot be inferred from the output of metric learning algorithms. When attempting to apply guarantees to real world settings, special care must be taken to ensure that the assumptions of the theory match the practical situation to which they are being applied. Indeed, any assumptions taken in this work may be reasonably considered a limitation depending on the practitioner's desired application, though we hope that this work serves more as a guide for research and that its assumptions can be altered and its results can be reshown in new settings of practical interest.

Finally, we remark on the broader impacts of this work. While this paper is primarily theoretical, it is worth considering the impacts of the problems to which we are providing theory. Metric learning has recently been adopted as a model for facial recognition which may lead to negative sociopolitical externalities. Preference learning allows for algorithms to more adeptly specialize to users' preferences in recommendation and this may cause adverse effects owing to what content is shown to users.

B PROOFS

B.1 PROOF OF LEMMA 1

We first note that orthogonal projections are bounded linear operators. Therefore, $\mathcal{P}_{\mathcal{S}_\mathcal{X}}$'s are bounded and linear. One can easily show that compositions of bounded linear operators are also bounded and linear. Therefore, $L\mathcal{P}_{\mathcal{S}_\mathcal{X}}$ is bounded and linear for any orthogonal projection $\mathcal{P}_{\mathcal{S}_\mathcal{X}}$. Then, for $p \geq 1$, we have

$$\begin{aligned} \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger L^\dagger L \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_p} &\overset{a}{\leq} \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger\|_{S_\infty} \|L^\dagger L \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_p} \\ &\overset{b}{\leq} \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger\|_{S_\infty} \|L^\dagger L\|_{S_p} \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_\infty} \\ &= \|L^\dagger L\|_{S_p}, \end{aligned}$$

where (a) and (b) follow from Hölder's inequality. Note that $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_\infty}$ is the standard operator norm on $\mathcal{H}$, i.e., $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_\infty} = \max_{\|x\| \leq 1} \|\mathcal{P}_{\mathcal{S}_\mathcal{X}} x\|$. Since $\mathcal{P}_{\mathcal{S}_\mathcal{X}}$ is an orthogonal projection, we have $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_\infty} = \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger\|_{S_\infty} = 1$.

B.2 PROOF OF PROPOSITION 1

We rewrite $\widehat{L}$ and $\widehat{L}_0$ together with (P2) and (P3) below for the ease of readability.

$$\widehat{L} := \arg\min_{L} \quad \widehat{R}_{\mathcal{S}}(L) \quad \text{s.t.} \quad \|L^\dagger L\|_{S_2} \leq \lambda_F \tag{P2}$$

$$\widehat{L}_0 := \arg\min_{L} \quad \widehat{R}_{\mathcal{S}}(L) \quad \text{s.t.} \quad \|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger L^\dagger L \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \leq \lambda_F \tag{P3}$$

We have $\|\widehat{L}^\dagger \widehat{L}\|_{S_2} \leq \lambda_F$ by definition. From Lemma 1, we also have

$$\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger \widehat{L}^\dagger \widehat{L} \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \leq \|\widehat{L}^\dagger \widehat{L}\|_{S_2}.$$

Thus, it holds that $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger \widehat{L}^\dagger \widehat{L} \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \leq \lambda_F$. Therefore, $\widehat{L}$ belongs to the feasible set of optimization problem (P3) and we can conclude that $\widehat{R}_\mathcal{S}(\widehat{L})$ is at least as small as $\widehat{R}_\mathcal{S}(\widehat{L}_0)$, i.e.,

$$\widehat{R}_\mathcal{S}(\widehat{L}_0) \leq \widehat{R}_\mathcal{S}(\widehat{L}). \tag{6}$$

For the reverse inequality, note that $\widehat{L}_0 = \widehat{L}_0 \mathcal{P}_{\mathcal{S}_\mathcal{X}}$. Therefore, $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger \widehat{L}_0^\dagger \widehat{L}_0 \mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} = \|\widehat{L}_0^\dagger \widehat{L}_0\|_{S_2}$ and $\widehat{L}_0$ belongs to the feasible set of (P2). Hence,

$$\widehat{R}_\mathcal{S}(\widehat{L}) \leq \widehat{R}_\mathcal{S}(\widehat{L}_0). \tag{7}$$

Based on (6) and (7), we find that

$$\widehat{R}_\mathcal{S}(\widehat{L}) = \widehat{R}_\mathcal{S}(\widehat{L}_0). \tag{8}$$

Furthermore, note that any L within the feasible set of (P2) also belongs to the feasible set of (P3). Provided that $L\mathcal{P}_{\mathcal{S}_\mathcal{X}} = \widehat{L}_0$, we also conclude that L is an optimal solution for (P2), since $\widehat{R}_\mathcal{S}(L\mathcal{P}_{\mathcal{S}_\mathcal{X}}) = \widehat{R}_\mathcal{S}(\widehat{L}_0) = \widehat{R}_\mathcal{S}(\widehat{L})$.

B.3 PROOF OF THEOREM 1

From Proposition 1, we have

$$\widehat{R}_\mathcal{S}(\widehat{L}_0) = \widehat{R}_\mathcal{S}(\widehat{L}). \tag{9}$$

Then, using standard Rademacher complexity bounding techniques, we can write following

$$\begin{aligned}
R(\widehat{L}_0) - R(L^*) \quad &= \quad R(\widehat{L}_0) - \widehat{R}_\mathcal{S}(\widehat{L}_0) + \widehat{R}_\mathcal{S}(\widehat{L}_0) - \widehat{R}_\mathcal{S}(L^*) + \widehat{R}_\mathcal{S}(L^*) - R(L^*) \\
&\overset{a}{=} \quad R(\widehat{L}_0) - \widehat{R}_\mathcal{S}(\widehat{L}_0) + \widehat{R}_\mathcal{S}(\widehat{L}) - \widehat{R}_\mathcal{S}(L^*) + \widehat{R}_\mathcal{S}(L^*) - R(L^*) \\
&\leq \quad 2 \sup_L |\widehat{R}_\mathcal{S}(L) - R(L)| \\
&\leq \quad 2\mathbb{E}_{\mathcal{S}\sim D}[\sup_L |\widehat{R}_\mathcal{S}(L) - R(L)|] + \beta\sqrt{\frac{2\ln 2/\delta}{|\mathcal{S}|}}
\end{aligned} \tag{10}$$

where $\beta := \sup|\ell(\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2) - \ell(\|L\phi_h' - L\phi_i'\|_\mathcal{H}^2 - \|L\phi_h' - L\phi_j'\|_\mathcal{H}^2)|$ and (a) is from (9). Note that $\beta \leq 12\alpha\lambda_F B^2$, since the difference of triplets is bounded by $6\lambda_F B^2$ (see Lemma 3) and the loss is $\alpha-$Lipschitz.

Now, using standard symmetrization and contraction lemmas, we may introduce $\epsilon_t \in \{-1, 1\}$'s, that are Rademacher random variables corresponding to each triplet t. Then, we have

$$\mathbb{E}_{\mathcal{S}\sim\mathcal{D}}[\sup_L |\widehat{R}_\mathcal{S}(L) - R(L)|] \leq \mathbb{E}_{\mathcal{S}\sim\mathcal{D},\epsilon\sim\{\pm 1\}^{|\mathcal{S}|}} \frac{2\alpha}{|\mathcal{S}|}\left[\sup_L \sum_{t\in\mathcal{S}} \epsilon_t(\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2))\right]$$

The expression inside the expectation on the right hand side can be considered as a function of random triplets in $\mathcal{S}$. We focus on the expectation on the right hand side:

$$\mathbb{E}_{\mathcal{S}\sim\mathcal{D},\epsilon\sim\{\pm 1\}^{|\mathcal{S}|}}\left[\sup_L \sum_{t\in\mathcal{S}} \epsilon_t(\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2))\right]. \tag{11}$$

Note that (11) is finite, since the difference of triplets is bounded. Therefore, we can apply Fubini's Theorem, and write it as

$$\mathbb{E}_\mathcal{S}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_L \sum_{t\in\mathcal{S}} \epsilon_t(\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2))\right]\right] \tag{12}$$

where $\mathbb{E}_{\epsilon|\mathcal{S}}$ is the conditional expectation given $\mathcal{S}$. In (11), we have a set of random triplets with corresponding random features $\phi_1, \ldots, \phi_n$ inside the expectation, where randomness is based on the triplet set $\mathcal{S}$. However, the conditional expectation $\mathbb{E}_{\epsilon|\mathcal{S}}$ in (12) is conditioned on $\mathcal{S}$. Note that the size of the Rademacher random vector ϵ is $|\mathcal{S}|$. We first focus on the conditional expectation:

$$\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_{L}\sum_{t\in\mathcal{S}}\epsilon_t(\|L\phi_h - L\phi_i\|^2_{\mathcal{H}} - \|L\phi_h - L\phi_j\|^2_{\mathcal{H}}))\right]. \tag{13}$$

Consider the span of features $\phi_1,\ldots,\phi_n$ and call it $\mathcal{S}_{\mathcal{X}}$. Using Riesz's Representation Theorem, we can write $L\phi$ for any ϕ as follows:

$$L\phi = \sum_{k=1}^{\infty}\langle\phi,\tau_k\rangle_{\mathcal{H}}\mathbf{e}_k$$

For the conditional expectation in (13), we can write each τ_k as the summation of τ_k' and $\tau_k^{\perp}$, where τ_k' represents the part lies in $\mathcal{S}_{\mathcal{X}}$ and $\tau_k^{\perp}$ is orthogonal to $\mathcal{S}_{\mathcal{X}}$.

$$\tau_k = \tau_k' + \tau_k^{\perp}.$$

We can represent each τ_k' as $\sum_{j=1}^{n} v_{k,j}\psi_j$, where $\{\psi_1,\ldots,\psi_n\}$ is an orthonormal basis for the set $\{\phi_1,\ldots,\phi_n\}$ and $v_{k,j}\in\mathbb{R},\forall k,j$. Therefore, for any $\phi_i,\phi_j\in\mathcal{S}_{\mathcal{X}}$,

$$\begin{aligned}
\langle L\phi_i, L\phi_j\rangle_{\mathcal{H}} &= \sum_{k=1}^{\infty}\langle\phi_i,\tau_k\rangle_{\mathcal{H}}\langle\phi_j,\tau_k\rangle_{\mathcal{H}} \\
&= \sum_{a=1}^{n}\sum_{b=1}^{n}\left(\sum_{k=1}^{\infty}v_{k,a}v_{k,b}\right)\langle\phi_i,\psi_a\rangle_{\mathcal{H}}\langle\phi_j,\psi_b\rangle_{\mathcal{H}} \\
&= \varphi_i^T\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}\varphi_j
\end{aligned} \tag{14}$$

where $\varphi_i = [\langle\phi_i,\psi_1\rangle,\langle\phi_i,\psi_2\rangle,\ldots\langle\phi_i,\psi_n\rangle]^T$ and $\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}_{i,j} = \sum_{k=1}^{\infty}v_{k,j}v_{k,i}$. Note that $\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}$ and $\{\varphi_1,\ldots,\varphi_n\}$ are functions of $\mathcal{S}$. Based on (14), for $\phi_i,\phi_j\in\mathcal{S}$, we have

$$\begin{aligned}
& \|L\phi_h - L\phi_i\|^2_{\mathcal{H}} - \|L\phi_h - L\phi_j\|^2_{\mathcal{H}} \\
= \;& (\varphi_j-\varphi_i)^T\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(2\varphi_h-\varphi_i-\varphi_j) \\
= \;& \frac{1}{2}\left((\varphi_j-\varphi_i)^T\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(2\varphi_h-\varphi_i-\varphi_j) + (2\varphi_h-\varphi_i-\varphi_j)^T\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(\varphi_j-\varphi_i)\right) \\
= \;& \frac{1}{2}\mathrm{Tr}\left(\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(2\varphi_h-\varphi_i-\varphi_j)(\varphi_j-\varphi_i)^T + \mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(\varphi_j-\varphi_i)(2\varphi_h-\varphi_i-\varphi_j)^T\right) \\
= \;& \mathrm{Tr}\left(\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}(\varphi_h\varphi_j^T+\varphi_j\varphi_h^T-\varphi_h\varphi_i^T-\varphi_i\varphi_h^T+\varphi_i\varphi_i^T-\varphi_j\varphi_j^T)\right)
\end{aligned}$$

Suppose $\mathbf{K}_t = \varphi_h\varphi_j^T+\varphi_j\varphi_h^T-\varphi_h\varphi_i^T-\varphi_i\varphi_h^T+\varphi_i\varphi_i^T-\varphi_j\varphi_j^T$. Then, we have

$$\begin{aligned}
&\mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_{L}\sum_{t\in\mathcal{S}}\epsilon_t(\|L\phi_h - L\phi_i\|^2_{\mathcal{H}} - \|L\phi_h - L\phi_j\|^2_{\mathcal{H}})\right]\right] \\
&\qquad = \mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_{L}\mathrm{Tr}\left(\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right)\right]\right].
\end{aligned} \tag{15}$$

For the expression inside the expectations in (15), we have

$$\begin{aligned}
\sup_{L}\mathrm{Tr}\left(\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right) &\overset{a}{\leq} \sup_{L}\sum_{i=1}^{r}\sigma_i(\mathbf{M}^{\mathcal{S}_{\mathcal{X}}})\sigma_i\left(\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right) \\
&\overset{b}{\leq} \sup_{L}\left[\|\mathbf{M}^{\mathcal{S}_{\mathcal{X}}}\|_{\mathrm{F}}\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|_{\mathrm{F}}\right] \\
&\overset{c}{\leq} \lambda_F\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|_{\mathrm{F}} \\
&= \lambda_F\sqrt{\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|^2_{\mathrm{F}}}.
\end{aligned} \tag{16}$$

Here, (a) is from Von Neumann's trace inequality, (b) is the result of Cauchy–Schwarz Inequality and we recall that $\|\mathbf{M}^{\mathcal{S}_\mathcal{X}}\|_{\mathrm{F}} \le \lambda_F$. Inserting (16) into (15), we can write

$$\mathbb{E}_\mathcal{S}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_L \sum_{t\in\mathcal{S}} \epsilon_t(\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2)\right]\right] \le \lambda_F \mathbb{E}_\mathcal{S}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sqrt{\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|_\mathrm{F}^2}\right]\right].$$

Then, we have

$$\begin{aligned}
\mathbb{E}_\mathcal{S}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sqrt{\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|_\mathrm{F}^2}\right]\right] &\overset{a}{\le} \mathbb{E}_\mathcal{S}\left[\sqrt{\mathbb{E}_{\epsilon|\mathcal{S}}\left[\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|_\mathrm{F}^2\right]}\right] \\
&= \mathbb{E}_\mathcal{S}\left[\sqrt{\mathbb{E}_{\epsilon|\mathcal{S}}\left[\langle\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t, \sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\rangle\right]}\right] \\
&= \mathbb{E}_\mathcal{S}\left[\sqrt{\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sum_{t\in\mathcal{S}}\sum_{t'\in\mathcal{S}}\epsilon_t\epsilon_{t'}\langle\mathbf{K}_t, \mathbf{K}_{t'}\rangle\right]}\right] \\
&\overset{b}{=} \mathbb{E}_\mathcal{S}\left[\sqrt{\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sum_{t\in\mathcal{S}}\epsilon_t^2\langle\mathbf{K}_t, \mathbf{K}_t\rangle\right]}\right] \\
&= \mathbb{E}_\mathcal{S}\left[\sqrt{\sum_{t\in\mathcal{S}}\|\mathbf{K}_t\|_\mathrm{F}^2}\right] \\
&\le B^2\sqrt{|S|6} \qquad (17)
\end{aligned}$$

where (a) is from Jensen's inequality where the expectation is over the randomness in ϵ_t and (b) is due the fact that $\mathbb{E}(\epsilon_{t_1}\epsilon_{t_2}) = 0$ when $t_1 \ne t_2$. For the last step, recall that $\mathbf{K}_t = \varphi_h\varphi_j^T + \varphi_j\varphi_h^T - \varphi_h\varphi_i^T - \varphi_i\varphi_h^T + \varphi_i\varphi_i^T - \varphi_j\varphi_j^T$. Then, we have

$$\begin{aligned}
\|\mathbf{K}_t\|_\mathrm{F}^2 &\overset{a}{\le} 6\max_{i,j}\|\varphi_i\varphi_j^T\|_\mathrm{F}^2 \\
&\overset{b}{\le} 6B^4, \qquad (18)
\end{aligned}$$

where (a) is by triangle inequality and (b) follows from that fact that $\|\varphi\|_2 = \|\phi_i\|_\mathcal{H} \le B$. Note that $\|\varphi\|_2 = \|\phi_i\|_\mathcal{H}$ is by definition, where φ_i is defined via change of basis on the span $\mathcal{S}_\mathcal{X}$. Finally, from (10) and (17), we have

$$R(\widehat{L}_0) - R(L^*) \le 4\alpha B^2\lambda_F\sqrt{\frac{6}{|S|}} + 2\ell\sqrt{\frac{2\gamma^2\ln 2/\delta}{|\mathcal{S}|}},$$

which completes the proof of Theorem 1.

Lemma 3. *Let $\phi(\boldsymbol{x})$ be a feature map from $\mathbb{R}^d$ to $\mathcal{H}$ with $\|\phi(\boldsymbol{x})\|_\mathcal{H} \le B$ for $\forall\boldsymbol{x}$, and L be a linear functional such that $L: \mathcal{H} \to \mathcal{H}$ and $\|L^\dagger L\|_{S_2} \le \lambda_F$. Then, for any $\boldsymbol{x}_h, \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$, we have*

$$\|L\phi_h - L\phi_i\|_\mathcal{H}^2 - \|L\phi_h - L\phi_j\|_\mathcal{H}^2 \le 6B^2\lambda_F$$

Proof of Lemma 3 First, note that

$$\begin{aligned}
\langle L\phi_h, L\phi_j\rangle_\mathcal{H} &= \langle\phi_h, L^\dagger L\phi_j\rangle_\mathcal{H} \\
&\overset{a}{\le} \|\phi_h\|_\mathcal{H}\|L^\dagger L\phi_j\|_\mathcal{H} \\
&\overset{b}{\le} \|\phi_h\|_\mathcal{H}\|L^\dagger L\|_{S_\infty}\|\phi_j\|_\mathcal{H} \\
&\le \|\phi_h\|_\mathcal{H}\|L^\dagger L\|_{S_2}\|\phi_j\|_\mathcal{H} \\
&\le B^2\lambda_F,
\end{aligned}$$

where (a) is from Cauchy-Schwarz Inequality and (b) is by definition of operator norm ($\|\cdot\|_{S_\infty}$). Then, we have

$$\begin{aligned}\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2 &= 2\langle L\phi_h, L\phi_j\rangle_{\mathcal{H}} - 2\langle L\phi_h, L\phi_i\rangle_{\mathcal{H}} + \langle L\phi_i, L\phi_i\rangle_{\mathcal{H}} - \langle L\phi_j, L\phi_j\rangle_{\mathcal{H}} \\ &\le 6B^2\lambda_F.\end{aligned}$$

B.4 PROOF OF THEOREM 2

Recall that the only difference between the setting in Theorem 1 and the setting in Theorem 2 is the constraint set. We replace the constraints $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger L^\dagger L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \le \lambda_F$ and $\|\mathbf{M}\|_F \le \lambda_F$ with $\|\mathcal{P}_{\mathcal{S}_\mathcal{X}}^\dagger L^\dagger L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_1} \le \lambda_*$ and $\|\mathbf{M}\|_* \le \lambda_*$ respectively. We update definitions accordingly. Then, the proof follows the same steps with the proof of Theorem 1 until (15), where we have

$$R(\widehat{L}_{n_0}) - R(L_n^*) \le \frac{4\alpha}{|\mathcal{S}|}\mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_L \operatorname{Tr}\left(\mathbf{M}^{\mathcal{S}_\mathcal{X}}\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right)\right]\right] + \beta\sqrt{\frac{2\ln 2/\delta}{|\mathcal{S}|}} \tag{19}$$

We focus on the expression inside the expectations and we can write

$$\begin{aligned}\sup_L \operatorname{Tr}\left(\mathbf{M}^{\mathcal{S}_\mathcal{X}}\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right) &\overset{a}{\le} \sup_L \|\mathbf{M}^{\mathcal{S}_\mathcal{X}}\|\left\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right\| \\ &\le \sup_L \|\mathbf{M}^{\mathcal{S}_\mathcal{X}}\|_*\left\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right\| \\ &\overset{b}{\le} \lambda_*\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\|\end{aligned} \tag{20}$$

Here, (a) is from Hölder's Ineqaulity for Schatten norms and we recall that $\|\mathbf{M}^{\mathcal{S}_\mathcal{X}}\|_* \le \lambda_*$ for (b). Inserting (20) into the expectations in (19), we can write

$$\mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\sup_L\sum_{t\in\mathcal{S}}\epsilon_t(\|L\phi_h - L\phi_i\|_{\mathcal{H}}^2 - \|L\phi_h - L\phi_j\|_{\mathcal{H}}^2)\right]\right] \le \lambda_*\mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\left\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right\|\right]\right].$$

Then, we have

$$\begin{aligned}\lambda_*\mathbb{E}_{\mathcal{S}}\left[\mathbb{E}_{\epsilon|\mathcal{S}}\left[\left\|\sum_{t\in\mathcal{S}}\epsilon_t\mathbf{K}_t\right\|\right]\right] &\overset{c}{\le} \lambda_*\mathbb{E}_{\mathcal{S}}\left[\sqrt{2\left\|\sum_{t=1}^{|\mathcal{S}|}\mathbb{E}_\epsilon\left[\mathbf{K}_t^2\right]\right\|\log 3|\mathcal{S}|} + 2\log 3|\mathcal{S}|\right] \\ &\overset{d}{\le} \lambda_*\mathbb{E}_{\mathcal{S}}\left[\sqrt{12B^4|\mathcal{S}|\log 3|\mathcal{S}|} + 2\log 3|\mathcal{S}|\right] \\ &= \lambda_*\left(\sqrt{12B^4|\mathcal{S}|\log 3|\mathcal{S}|} + 2\log 3|\mathcal{S}|\right)\end{aligned} \tag{21}$$

where we apply a matrix Bernstein bound to get (c) (see Theorem 6.6.1 in Tropp et al. [2015]) and (d) follows from (18). Lastly, from (19) and (21), we have

$$R(\widehat{L}_{n_0}) - R(L_n^*) \le 4\alpha\lambda_*\left(B^2\sqrt{12\frac{\log 3|\mathcal{S}|}{|\mathcal{S}|}} + \frac{2\log 3|\mathcal{S}|}{|\mathcal{S}|}\right) + 12\alpha B^2\lambda_*\sqrt{\frac{2\ln 2/\delta}{|\mathcal{S}|}},$$

which completes the proof of Theorem 2.

B.5 PROOF OF LEMMA 2

Let $\psi_1,\dots,\psi_n \in \mathcal{H}$ denote the KPCA directions which span $\mathcal{S}_\mathcal{X}$ such that $\langle\psi_i,\phi_j\rangle = (\varphi_j)_i \in \mathbb{R}$, where $(v)_i$ denotes the i^{th} entry of vector v. Furthermore, let $\mathbf{B}_i$, denote the ith row of a matrix $\mathbf{B}$. For any $L : \mathcal{H} \to \mathcal{H}$ and $\phi_x \in \mathcal{H}$ we may write $L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_x = \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\Psi_i \otimes \Psi_j\phi_x$, where $\Psi_i \otimes \Psi_j\phi_x = \langle\Psi_j,\phi_x\rangle_{\mathcal{H}}\Psi_i$. Let $\mathbf{W}$ be the matrix of w_{ij} weights. Lastly, let $a^T b$ denote the standard Euclidean inner product for $a, b \in \mathbb{R}^n$. Then, for $\phi_x, \phi_y \in \mathcal{H}$,

$$\begin{aligned}
\|L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_x - L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_y\|^2_{\mathcal{H}} &= \langle L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_x - L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_y, L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_x - L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_y\rangle \qquad (22)\\
&= \left\langle \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\Psi_i \otimes \Psi_j(\phi_x - \phi_y), \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\Psi_i \otimes \Psi_j(\phi_x - \phi_y)\right\rangle_{\mathcal{H}}\\
&= \left\langle \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\langle\Psi_j, \phi_x - \phi_y\rangle_{\mathcal{H}}\Psi_i, \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\langle\Psi_j, \phi_x - \phi_y\rangle_{\mathcal{H}}\Psi_i\right\rangle_{\mathcal{H}}\\
&= \left\langle \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\left((\varphi_{\boldsymbol{x}})_j - (\varphi_{\mathbf{y}})_j\right)\Psi_i, \sum_{i=1}^n\sum_{j=1}^n w_{i,j}\left((\varphi_{\boldsymbol{x}})_j - (\varphi_{\mathbf{y}})_j\right)\Psi_i\right\rangle_{\mathcal{H}}\\
&= \left\langle \sum_{i=1}^n \mathbf{W}_i^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\Psi_i, \sum_{i=1}^n \mathbf{W}_i^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\Psi_i\right\rangle_{\mathcal{H}}\\
&= \sum_{i=1}^n \mathbf{W}_i^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\left\langle \Psi_i, \sum_{j=1}^n \mathbf{W}_j^T(\varphi_{\mathbf{c}} - \varphi_{\mathbf{y}})\Psi_j\right\rangle_{\mathcal{H}}\\
&= \sum_{i=1}^n \left(\mathbf{W}_i^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\right)^2\langle\Psi_i, \Psi_i\rangle_{\mathcal{H}}\\
&= \sum_{i=1}^n (\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})^T\mathbf{W}_i\mathbf{W}_i^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\\
&= (\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})^T\mathbf{W}\mathbf{W}^T(\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}})\\
&= \|\varphi_{\boldsymbol{x}} - \varphi_{\mathbf{y}}\|^2_{\mathbf{M}} \qquad (23)
\end{aligned}$$

where in the final step we have defined $\mathbf{M} := \mathbf{W}\mathbf{W}^T$. Then the eigenvalues of $\mathbf{M}$ are equal to the square of the singular values of $\mathbf{W}$. In general we note that the eigenvalues of $(L\mathcal{P}_{\mathcal{S}_\mathcal{X}})^\dagger L\mathcal{P}_{\mathcal{S}_\mathcal{X}}$ are equal to the eigenvalues of $\mathbf{M}$ where $L^\dagger$ denote the adjoint of L. Note that $\|L\phi_x - L\phi_y\|^2_{\mathcal{H}} = \|L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_x - L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\phi_y\|^2_{\mathcal{H}}$ for $\phi_x, \phi_y \in \mathcal{S}_\mathcal{X}$. Hence, we have $\|L\phi_{\boldsymbol{x}}\|_{\mathcal{H}} = \|\varphi_{\boldsymbol{x}}\|_{\mathbf{M}}$ for any $\phi_x \in \mathcal{S}_\mathcal{X}$ from (23).

B.6 PROOF OF PROPOSITION 3

From Lemma 2, we have $\|\mathcal{P}^\dagger_{\mathcal{S}_\mathcal{X}}L^\dagger L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_p} = \|\mathbf{M}\|_p\ \forall p$. Similarly, from the fact that $\|L\phi_{\boldsymbol{x}}\|_{\mathcal{H}} = \|\varphi_{\boldsymbol{x}}\|_{\mathbf{M}}$ (see Lemma 2), we have $|\|L\phi_h - L\phi_i\|^2 - \|L\phi_h - L\phi_j\|^2| = |\|\varphi_h - \varphi_i\|^2_{\mathbf{M}} - \|\varphi_h - \varphi_j\|^2_{\mathbf{M}}|$ within $\mathcal{S}_\mathcal{X}$. Then, from Proposition 2, we conclude that (P3) and $(P4)$ have the same optimal value. Therefore, we have that

$$\begin{aligned}
\min_{L}\quad & \widehat{R}_{\mathcal{S}}(L)\\
\text{s.t.}\quad & \|\mathcal{P}^\dagger_{\mathcal{S}_\mathcal{X}}L^\dagger L\mathcal{P}_{\mathcal{S}_\mathcal{X}}\|_{S_2} \le \lambda_F
\end{aligned} \qquad \text{(P3)}$$

is equal to

$$\begin{aligned}
\min_{\mathbf{M}}\quad & \widehat{\widehat{R}}_{\mathcal{S}}(\mathbf{M})\\
\text{s.t.}\quad & \|\mathbf{M}\|_F \le \lambda_F\\
& \mathbf{M} \succeq 0,
\end{aligned} \qquad \text{(P4)}$$

By definition, $\widehat{\mathbf{L}}_0$ is an optimal solution for (P3) and $\widehat{\mathbf{M}}$ is the optimal solution for (P4). Recall that there exists a psd matrix $\mathbf{M}$ for each pair of L and $\mathcal{S}_\mathcal{X}$ from Lemma 2.

For the construction of $\widehat{L}_0$ from $\widehat{\mathbf{M}}$, we follow similar lines with the proof of Lemma 2. $\widehat{L}_0$ is defined reversing equalities in Section B.5 for $\mathbf{M} = \widehat{\mathbf{M}}$, from (22) to (23). Therefore, we observe that $\widehat{\mathbf{M}}$ is the corresponding psd matrix for the pair $(\widehat{L}_0, \mathcal{S}_\mathcal{X})$. This is actually true for any L provided that $L\mathcal{P}_{\mathcal{S}_\mathcal{X}} = \widehat{L}_0$.

C DISCUSSION

Our results extend the linear metric setting of Mason et al. [2017] in two key ways: First, our main results provide generalization error and sample complexity bounds for the kernelized metric learning from triplet comparisons. Second, the linear metric learning analysis of Mason et al. [2017] requires that the number of items, n, be larger than the dimensionality, d, which limits its applicability. In contrast, our analysis, which also considers linear kernels, offers a more general framework, even for linear metric learning from triplet comparisons.

Mason et al. [2017] consider a fixed set of items in $\mathbb{R}^d$ and derive generalization bounds based on selecting triplets uniformly from those that can be generated from the fixed item set. Their analysis exploits the fact that the item set is fixed and requires that the number of items n is larger than the dimensionality d, which limits its applicability. Also note that, the true risk of Mason et al. [2017] is defined with respect to a discrete uniform distribution over $n\binom{n-1}{2}$ triplets possible from the fixed set of n items.

Our setting differs significantly from Mason et al. [2017] in the following aspects: We do not assume a fixed set of items, which would otherwise restrict generalization only to the triplets drawn from this fixed set. Instead, in our setting, each triplet query involves items drawn iid from an unknown distribution $\mathcal{D}$. Our true risk is defined over this unknown distribution and the generalization bounds hold for triplets chosen from this distribution. Thus, our analysis also extends the generalization results even for the linear kernel case in high dimensions (large d) apart from generalizing to infinite-dimensional RKHS.

Given the difference in settings, the proof technique we use differs from Mason et al. [2017]). To derive our sample complexity results, we turn our attention to the metric and exploit the fact that the true metric L^* has a bounded Schatten$-p$ norm, which constrains how L interacts with any random data. We use this constraint in conjunction with the Riesz Representation Theorem to further refine our analysis.

D ADDITIONAL SIMULATIONS AND EXPERIMENTAL DETAILS

In the practical implementation of kernelized metric learning problem, our target is to solve convex program (P4). Solving (P4), we learn a finite metric $\widehat{\mathbf{M}}$.

Unseen Triplets: To evaluate the performance of $\widehat{\mathbf{M}}$ for unseen triplets, we, first find $\varphi_{n+1} = \mathbf{A}^T[k(\boldsymbol{x}_{n+1}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}_{n+1}, \boldsymbol{x}_n)]^T$ for each new point $\boldsymbol{x}_{n+1}$ seen in the test set using kernel function $k(x, y)$, where $\mathbf{A}$ is from KPCA procedure (see Section 4.1). This corresponds to finding the projections of new points to the span of $\phi_1 \dots \phi_n$. Then, we can estimate the label for an unseen triplet using new (finite) representations φ_{n+1}'s and $\widehat{\mathbf{M}}$.

Computing Infrastructure: Our code is designed to run on a personal laptop. The experiment and simulations reported in this paper were conducted on a MacBook Pro with M3 Max CPU with 48GB of RAM.

We will open-source our code for reproducibility upon acceptance of this work.

D.1 SPIRAL WITH GEODESIC DISTANCE:

We present the performance of different kernels in Figure 2 for the task of metric learning on a 2D spiral, where the true distance is the geodesic distance. Table 1 shows parameters of the kernel functions used for this task, which are as follows: $\sigma = 1, c = 1, \alpha = 1, p = 2$.

D.2 GAUSSIAN KERNEL MAP

Preliminary: We want to generate a linear functional $L^* : \mathcal{H} \to \mathcal{H}$ that lies on an $r-$dimensional manifold. First, note that Riesz's Representation Theorem allows us to represent the linear functional L^* as follows:

$$L^*\phi = \sum_{k=1}^{\infty} \langle \phi, \tau_k \rangle_{\mathcal{H}} \mathbf{e}_k.$$

Given that L^* lies on an $r-$dimensional manifold, each τ_k can be written as $\sum_{j=1}^{r} v_{k,j}\psi_j$, where $\{\psi_1, \dots, \psi_r\}$ is a set of features that span an $r-$dimensional manifold. Therefore, for any ϕ_i, ϕ_j,

$$\begin{aligned} \langle L\phi_i, L\phi_j \rangle_{\mathcal{H}} &= \sum_{k=1}^{\infty} \langle \phi_i, \tau_k \rangle_{\mathcal{H}} \langle \phi_j, \tau_k \rangle_{\mathcal{H}} \\ &= \sum_{a=1}^{r} \sum_{b=1}^{r} \left(\sum_{k=1}^{\infty} v_{k,a} v_{k,b} \right) \langle \phi_i, \psi_a \rangle_{\mathcal{H}} \langle \phi_j, \psi_b \rangle_{\mathcal{H}}, \qquad (24) \end{aligned}$$

where $\mathbf{G}_{a,b} = (\sum_{k=1}^{\infty} v_{k,a} v_{k,b})$. Each entry of $\mathbf{G}$ is an inner product in ℓ_2, so $\mathbf{G}$ is a positive semidefinite matrix. Our target is to sample a set of features in $\mathcal{H}$ that spans an r_0-dimensional manifold, where $r_0 = \max(r)$ and generate a random psd matrix $\mathbf{G}$ to define L^*. Inspiring from the simulation setup of Mason et al. [2017] for linear metric learning problem, we define $\mathbf{G}$ as $\mathbf{G} = \frac{r_0}{\sqrt{r}} \mathbf{U}\mathbf{U}^T$ to make average magnitude of entries constant independent from r and r_0, where $\mathbf{U} \in \mathbb{R}^{r_0 \times r}$ is a random orthogonal matrix. This procedure provides a linear functional L^* lying on an $r-$dimensional manifold.

Linear Functional L^*: We sample a set $\{z_1 \dots z_r\}$, where each $z_i \sim \mathcal{N}(\mathbf{0}_d, \frac{1}{d} I_d)$. Then, consider a kernel map $\phi(\cdot)$ such that $\langle \phi(z_i), \phi(z_j) \rangle = k(z_i, z_j)$. We generate corresponding features using this kernel map, where the set of features $\{\phi(z_1) \dots \phi(z_r)\}$ span an $r-$dimensional manifold in $\mathcal{H}$ and call $\psi_i = \phi(z_i)$. We also generate a random psd matrix $\mathbf{G}_{r \times r}$. Finally, we have an explicit formula for L^* based on (24). Now, we can express inner product $\langle L\phi_i, L\phi_j \rangle_{\mathcal{H}}$ in terms of known parameters:

$$\langle L\phi_i, L\phi_j \rangle_{\mathcal{H}} = [k(\boldsymbol{x}_i, z_1), \dots, k(\boldsymbol{x}_i, z_r)] \mathbf{G} [k(\boldsymbol{x}_j, z_1), \dots, k(\boldsymbol{x}_j, z_r)]^T, \tag{25}$$

where $\langle \phi_i, \psi_a \rangle_{\mathcal{H}} = k(\boldsymbol{x}_i, z_a)$ and $\phi_i = \phi(\boldsymbol{x}_i)$. We can easily find the difference of distances for triplet comparisons based on (25), since we have

$$\|L\phi(\boldsymbol{x}_h) - L\phi(\boldsymbol{x}_i)\|_{\mathcal{H}}^2 = \langle L\phi_h, L\phi_h \rangle_{\mathcal{H}} - 2\langle L\phi_h, L\phi_i \rangle_{\mathcal{H}} + \langle L\phi_i, L\phi_i \rangle_{\mathcal{H}}.$$

Triplet Generation: We randomly sample triples $\{\boldsymbol{x}_h, \boldsymbol{x}_i, \boldsymbol{x}_j\}$ where $\boldsymbol{x}_i \sim \mathcal{N}(\mathbf{0}_d, \frac{1}{d} I_d)$. Then, we can numerically find the difference of distances using (25) and generate noisy answers for triplets with a link function as mentioned in Section 5.1.

Accuracy: We generate another set of random triplets. We can numerically find the true label corresponding to each triplet using L^*. Finally, we compare true labels with estimated labels to find accuracy.

Below, we provide more extensive simulations with a Gaussian kernel, where $\sigma = 1$.

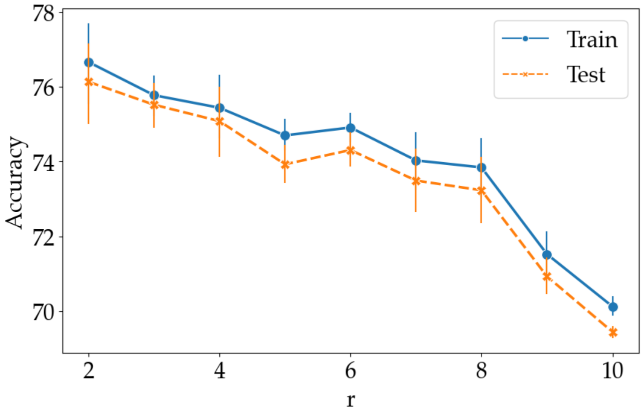

Figure 6: Train and test accuracy for noiseless setting with 50 repetitions for each run. We fix the number of triplets to 5000.

From Figure 6, we observe that given a set number of triplets, the accuracy one can obtain decreases as the rank r increases, as captured by our analysis, where L^* lies on an $r-$dimensional manifold. The task of learning a kernelized metric becomes more complex as r increases.

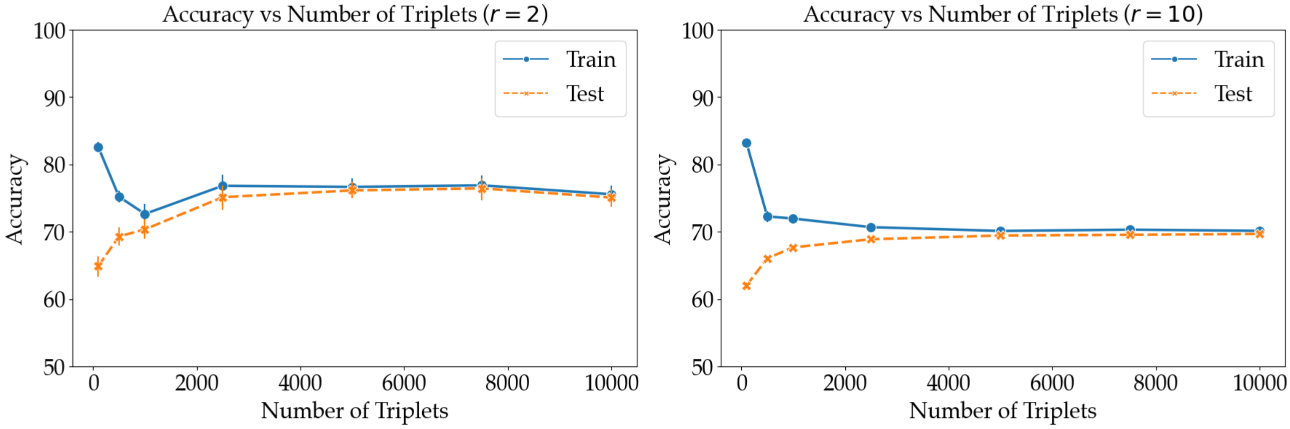

Figure 7: Train and test accuracy for noiseless setting with 50 repetitions varying number of triplets (100, 500, 1000, 2500, 5000, 10000), where $r = 2$ (left) and $r = 10$ (right).

Figure 7 shows that test accuracy increases when the triplet set gets larger. As a result, the learned metric generalizes better. For example, we observe that, to obtain the same accuracy of 70%, $\sim$ 1000 triplets are sufficient when rank is 2, whereas the triplets needed when rank is 10 is $\sim$ 5000.

Next, we provide simulation results with noisy responses. From Figure 8, we observe that accuracy is lower for larger r values even with a significant amount of noise on responses. Finally, Figure 9 shows accuracy for varying numbers of triplets at different noise levels of 5% and 10%.

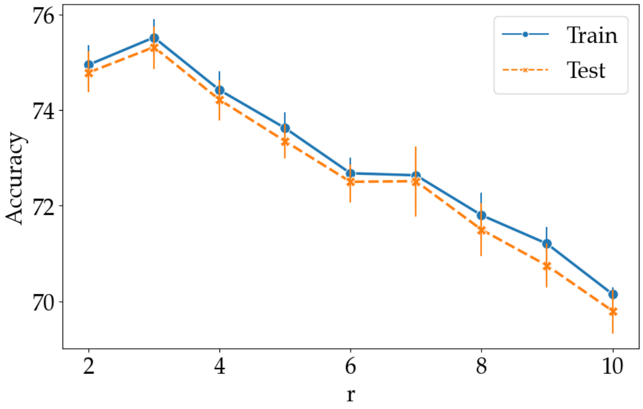

Figure 8: Train and test accuracy for noisy setting with 50 repetitions for each run. We fix number of triplets to 10000 and the ratio of noisy responses is approximately 5%.

D.3 EMPIRICAL EVALUATION: FOOD-100 DATASET

We provide a brief description for the Food-100 dataset (More details can be found in the work of Wilber et al. [2014]). The Food-100 dataset consists of carefully selected 100 food items, where each image has only one food. Answers to 190,376 triplets are collected from Amazon Mechanical Turk workers. Let $\mathcal{T}$ be the set of all triplets.

For each iteration, we randomly select 20 items and call them $\mathcal{X}_{\text{unseen}}$. Then, we define a triplet set $\mathcal{T}_{\text{unseen}}$ from $\mathcal{X}_{\text{unseen}}$ as follows:

$$\mathcal{T}_{\text{unseen}} := \{\{x_h, x_i, x_j\} : x_h \in \mathcal{X}_{\text{unseen}} \text{ or } x_i \in \mathcal{X}_{\text{unseen}} \text{ or } x_j \in \mathcal{X}_{\text{unseen}}\}.$$

Next, we uniformly sample triplets for the training set $\mathcal{T}_{\text{train}}$ from the set $\mathcal{T} \setminus \mathcal{T}_{\text{unseen}}$ to guarantee that there exist unseen items in $\mathcal{T}_{\text{train}}$. Finally, we uniformly sample triplets for the test set $\mathcal{T}_{\text{test}}$ from the set of all triplets $\mathcal{T}$. We apply the same

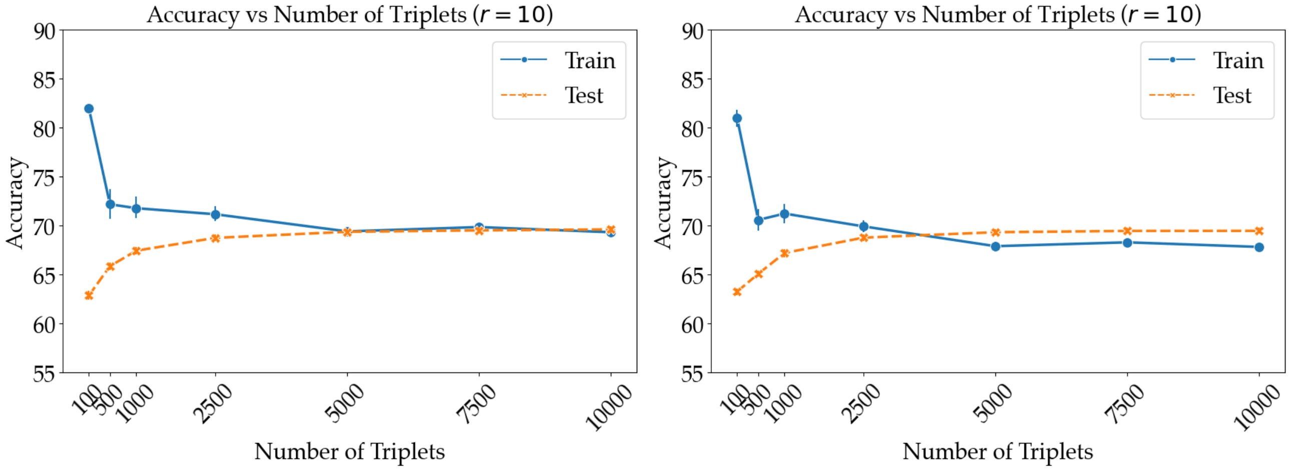

Figure 9: Train and test accuracy for noisy setting and $r = 10$ with 20 repetitions varying number of triplets, where the ratio of noisy responses is approximately 5% (left) and 10% (right).

splitting strategy on the $\mathcal{T}_{\text{train}}$ set to further split it to different training and validation part 20 times. We report the mean and standard deviation of the validation accuracies on these 20 validation parts.

Choice of Parameters for Kernel Function: We conducted a parameter search on the validation set in the following range:

- $\sigma : 0.01, 0.1, 1, 10$
- $\alpha : 0.01, 0.1, 1$
- $p : 2, 5, 7, 10$

Our results show the best test accuracy values based on this search.