Layer-wise Learning of Kernel Dependence Networks

Chieh Wu * **Aria Masoomi*** **Arthur Gretton** **Jennifer Dy**

Abstract

We propose a greedy strategy to train a deep network for multi-class classification, where each layer is defined as a composition of a linear projection and a nonlinear mapping. This nonlinear mapping is defined as the feature map of a Gaussian kernel, and the linear projection is learned by maximizing the dependence between the layer output and the labels, using the Hilbert Schmidt Independence Criterion (HSIC) as the dependence measure. Since each layer is trained greedily in sequence, all learning is local, and neither backpropagation nor even gradient descent is needed. The depth and width of the network are determined via natural guidelines, and the procedure regularizes its weights in the linear layer. As the key theoretical result, the function class represented by the network is proved to be sufficiently rich to learn any dataset labeling using a finite number of layers, in the sense of reaching minimum mean-squared error or cross-entropy, as long as no two data points with different labels coincide. Experiments demonstrate good generalization performance of the greedy approach across multiple benchmarks while showing a significant computational advantage against a multilayer perceptron of the same complexity trained globally by backpropagation.

1 Introduction

Since the seminal work by Rumelhart et al. [1], Multilayer Perceptrons (MLPs) have become a popular tool for classification. This success can in part be explained by the expressiveness of the resulting function class [2, 3, 4, 5]; in particular, a two-layer network can approximate any continuous function in a compact domain, to any desired accuracy, albeit with a network size exponential in input dimension. A helpful perspective on networks of large (or, in the limit, even infinite) width can be gained from kernel methods, for which the feature spaces can be infinite-dimensional by construction. For example, the Gaussian process (GP) has been used to understand the limiting behavior of wide networks [6, 7, 8]: in particular, deep GPs give a mechanism for constructing deep networks where each layer has infinitely many features [9, 10, 11, 12]. Following GPs, the Neural Tangent Kernel (NTK) was then proposed to describe the dynamics of the network during training [13, 14], further elucidating the relationship between MLPs and kernels. Moreover, Belkin et al. [15] has shown how MLPs and kernels both yield good generalization results despite overfitting, leading them to link kernel discovery to network training.

In this paper, we introduce a new function class for supervised classification, which builds a kernel neural network in a greedy layer-wise fashion: we call this the Kernel Dependence Network (*KNet*). Each layer is a composition of a linear subspace projection and an infinite-dimensional feature map. A key advantage is that the network can be trained greedily, layer by layer. *KNet* solves each layer by maximizing the dependence between the layer output and the labels, using the Hilbert Schmidt Independence Criterion (HSIC) as the dependence measure [16]. This can be achieved efficiently, even without Stochastic Gradient Descent (SGD), via the Iterative Spectral Method (ISM) [17, 18]. This local and greedy strategy is used in place of backpropagation (BP) to obviate the sharing of the gradient information throughout the network, thereby avoiding exploding/vanishing gradients. As a

*Signifies equal contribution.

Preprint. Under review.

consequence of our formulation, we demonstrate how the natural choice of the network width and depth also emerges.

A related work by Ma et al. [19] uses a chain of HSIC dependencies to simulate Information Bottleneck. They employ a standard network structure that is solved by SGD. In contrast, our kernel perspective replaces the conventional activation function with the feature map of a Gaussian kernel (GK), resulting in an infinitely wide network that is solved via the kernel trick. Additionally, our key contribution is the theoretical characterization of the richness of the function class described by our architecture. We prove a property related to that of *finite sample expressivity* [20, Section 4] for classical neural nets. Specifically, our network construction can achieve any desirable *training accuracy* given a minimum depth of 2 infinitely wide layers, with the implication of minimizing Mean Squared Error (MSE) and Cross-Entropy (CE) on the training data.

Similar to traditional MLPs, the richness of *KNet* promises to fit any training data with an overparameterized network, yet it also experimentally generalizes well on test data. To explain this observation for standard MLPs, the regularizing effects of architecture choice and optimization strategies have been used to explore how MLPs generalize [21, 22, 23]. As our second theoretical contribution, we demonstrate that our architecture and training procedure perform an implicit regularization, similar to the weight penalization arguments of Poggio et al. [24], and we describe the mechanisms by which this is achieved. We lastly verify every theoretical guarantee of our model experimentally, where *KNet* achieves comparable generalization results to MLPs of comparable complexity trained by BP.

2 Network Model

Let $X \in \mathbb{R}^{n\times d}$ be a dataset of n samples with d features and let $Y \in \mathbb{R}^{n\times \tau}$ be the corresponding one-hot encoded labels with τ number of classes. Let $\odot$ be the element-wise product. The i^{th} sample and label of the dataset is written as x_i and y_i. H is a centering matrix defined as $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ where I_n is the identity matrix of size $n \times n$ and $\mathbf{1}_n$ is a vector of 1s also of length n. Given H, we let $\Gamma = HYY^TH$ and let $K_{(\cdot)}$ be the Gaussian kernel (GK) matrix computed using a dataset matrix $(\cdot)$.

We denote the MLP weights as $W_1 \in \mathbb{R}^{d\times q}$ and $W_l \in \mathbb{R}^{m\times q}$ for the 1st layer and the l^{th} layer; assuming $l > 1$. The input and output at the l^{th} layer are $R_{l-1} \in \mathbb{R}^{n\times m}$ and $R_l \in \mathbb{R}^{n\times m}$, i.e., given $\psi : \mathbb{R}^{n\times q} \to \mathbb{R}^{n\times m}$ as the activation function, $R_l = \psi(R_{l-1}W_l)$. For each layer, the i^{th} row of its input R_{l-1} is $r_i \in \mathbb{R}^m$ and it represents the i^{th} input sample. We denote $\mathcal{W}_l$ as a function where $\mathcal{W}_l(R_{l-1}) = R_{l-1}W_l$; consequently, each layer is also a function $\phi_l = \psi \circ \mathcal{W}_l$. By stacking L layers together, the entire network itself becomes a function ϕ where $\phi = \phi_L \circ ... \circ \phi_1$. Given an empirical risk $(\mathcal{H})$ and a loss function $(\mathcal{L})$, our network model assumes an objective of

$$\min_{\phi} \quad \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\phi(x_i), y_i). \tag{1}$$

We propose to solve Eq. (1) greedily; this is equivalent to solving a sequence of *single-layered* networks where the previous network output becomes the current layer's input. At each layer, we find the W_l that maximizes the dependency between the layer output and the label via HSIC [16]:

$$\max_{W_l} \operatorname{Tr}\left(\Gamma \ \left[\psi(R_{l-1}W_l)\psi^T(R_{l-1}W_l)\right]\right) \quad \text{s. t. } W_l^TW_l = I. \tag{2}$$

Deviating from the traditional concept of activation functions, we use the GK feature map in their place, simulating an infinitely wide network. Yet, the kernel trick spares us the direct computation of $\psi(R_{l-1}W_l)\psi^T(R_{l-1}W_l)$; we instead compute the GK matrix given $\mathcal{K}(W_l^Tr_i, W_l^Tr_j) = \exp\{-||W_l^Tr_i - W_l^Tr_j||^2/2\sigma^2\}$. We also deviate from a standard MLP by restricting our solution space to a linear subspace. If the solution indeed lives on a linear subspace independent of their scale as suggested by [24, 25, 26], then we can exploit this prior knowledge to narrow the search space during optimization specific to the Stiefel Manifold, i.e., by adding the constraint of $W_l^TW_l = I$. This prior enables us to solve Eq. (2) by leveraging the iterative spectral method (ISM) proposed by Wu et al. [17, 18] to simultaneously avoid SGD and identify the network width. Applying ISM to our model, each layer's weight is initialized using the most dominant eigenvectors of

$$\mathcal{Q}_{l^0} = R_{l-1}^T(\Gamma - \text{Diag}(\Gamma 1_n))R_{l-1}, \tag{3}$$

where the Diag$(\cdot)$ function places the elements of a vector into the diagonal of a square matrix with zero elements. Once the initial weights W_{l^0} are set, ISM iteratively updates W_{l^i} to $W_{l^{i+1}}$ by setting $W_{l^{i+1}}$ to the most dominant eigenvectors of

$$\mathcal{Q}_{l^i} = R_{l-1}^T(\hat{\Gamma} - \text{Diag}(\hat{\Gamma}1_n))R_{l-1}, \tag{4}$$

where $\hat{\Gamma}$ is a function of W_{l^i} computed with $\hat{\Gamma} = \Gamma \odot K_{R_{l-1}W_{l^i}}$. This iterative weight-updating process stops when $\mathcal{Q}_{l^i} \approx \mathcal{Q}_{l^{i+1}}$, whereupon $\mathcal{Q}_{l^{i+1}}$ is set to $\mathcal{Q}_l^*$, and its most dominant eigenvectors W_l^* becomes the solution of Eq. (2).

ISM solves Eq. (2) directly on an infinitely wide network during training, obtaining W_l^*. However, during test, we approximate ψ with Random Fourier Features (RFF) [27], simulating a finite width network passing samples through W_l^* and ψ. Capitalizing on the spectral properties of ISM, the spectrum of $\mathcal{Q}_l^*$ completely determines the the width of the network $W_l^* \in \mathbb{R}^{m \times q}$, i.e., m is equal to the size of the RFF, and q is simply the rank of $\mathcal{Q}_l^*$. Furthermore, since Eq. (2) after normalization is upper bounded by 1, we can stop adding new layers when the HSIC value of the current layer approaches this theoretical bound, thereby prescribing a natural depth of the network. The resulting network ϕ after training will map samples of the same class into its own cluster, allowing the test samples to be classified by matching their network outputs to the nearest cluster center. We formally define the hyperparameter settings and provide Algorithm 1 in the Experimental section. The source code is also publicly available at `https://github.com/anonymous`.

3 Theoretical Origin of Kernel Dependence Networks

Background and Notations. Let $\mathcal{S}$ be a set of i, j sample pairs that belong to the same class. Its complement, $\mathcal{S}^c$ contains all sample pairs from different classes. We denote compositions of the first l layers as $\phi_{l^\circ} = \phi_l \circ ... \circ \phi_1$ where $l \leq L$. This notation enables us to connect the data directly to the layer output where $R_l = \phi_{l^\circ}(X)$. Since *KNet* is greedy, it solves MLPs by replacing ϕ in Eq. (1) incrementally with a sequence of functions $\{\phi_{l^\circ}\}_{l=1}^L$ where each layer relies on the weights of the previous layer. This implies that we are also solving a sequence of empirical risks $\{\mathcal{H}_l\}_{l=1}^L$, i.e., different versions of Eq. (1) given the current ϕ_{l°. We refer to $\{\phi_{l^\circ}\}_{l=1}^L$ and $\{\mathcal{H}_l\}_{l=1}^L$ as the *Kernel Sequence* and the *$\mathcal{H}$-Sequence*.

Classification Strategy. Classification tasks can be solved using objectives like MSE and CE to match the network output $\phi(X)$ to the label Y. While this approach achieves the desirable outcome, it also constrains the space of potential solutions where $\phi(X)$ must match Y. Yet, if ϕ maps X to the labels $\{0, 1\}$ instead of the true label $\{-1, 1\}$, $\phi(X)$ may not match Y, but the solution is the same. Therefore, enforcing $\phi(X) = Y$ ignores an entire space of solutions that are functionally equivalent. We posit that by relaxing this constraint and accept a larger space of potential global optima, it will be easier during optimization to collide with this space. This intuition motivates us to depart from the tradition of label matching, and instead seek out alternative objectives that focus on solving the underlying prerequisite of classification, i.e., learning a mapping of X where similar and different classes become distinguishable.

Since there are many notions of similarity, it is not always clear which is best for a particular situation. *KNet* overcomes this uncertainty by discovering the optimal similarity measure as a kernel function during training. To understand how, first realize that the i, j_{th} element of Γ, denoted as $\Gamma_{i,j}$, is a positive value for samples in $\mathcal{S}$ and negative for $\mathcal{S}^c$. By defining a kernel function $\mathcal{K}$ as a similarity measure between 2 samples, Eq. (2) becomes

$$\max_{W_l} \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \mathcal{K}_{W_l}(r_i, r_j) - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \mathcal{K}_{W_l}(r_i, r_j). \tag{5}$$

Notice that the objective uses the sign of $\Gamma_{i,j}$ as labels to guide the choice of W_l such that it increases $\mathcal{K}_{W_l}(r_i, r_j)$ when r_i, r_j belongs to $\mathcal{S}$ while decreasing $\mathcal{K}_{W_l}(r_i, r_j)$ otherwise. Therefore, by finding a W_l matrix that best parameterizes $\mathcal{K}$, HSIC discovers the optimal pair-wise relationship function $\mathcal{K}_{W_l}$ that separates samples into similar and dissimilar partitions. Given this strategy, we will formally demonstrate how learning $\mathcal{K}$ leads to classification in the following sections.

Optimization Strategy. MLP is traditionally solved with all the layers jointly via BP. We instead focus on greedily discovering a *Kernel Sequence* that compels the *$\mathcal{H}$-Sequence* to exhibit key behaviors that enable classification. Here, we discuss how these behaviors lead to an optimal solution.

First, the $\mathcal{H}$-*Sequence* must be convergent. We accomplish this by leveraging the Monotone Convergence Theorem (MCT) [28]: it states that a monotone sequence is guaranteed to have a limit if and only if the sequence is bounded. For *KNet*, since ψ is the feature map of GK, $\mathcal{K}_{W_l}(r_i, r_j)$ is naturally constrained between $[0, 1]$. Therefore, given our greedy strategy, $\mathcal{H}$-*Sequence* converges if we can achieve $\mathcal{H}_l \geq \mathcal{H}_{l-1}$ at every layer.

Second, while the MCT guarantees convergence, it does not guarantee the quality of its limit. Namely, the improvement at each layer could be so small such that the overall gain is trivial. To answer this question, we must investigate the potential contribution from each layer to identify the point of convergence. In the most ideal case, as $L \to \infty$ the network should converge towards an optimal kernel that achieves the theoretical upper limit of HSIC, or $\mathcal{H}^* = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j}$. Moreover, we should be able to achieve this within a finite number of layers.

We investigate these criteria using HSIC as the empirical risk of an MLP and prove that by using the feature map of a GK as ψ, there exists a sequence of weights $\{W_l\}_{l=1}^L$ where these considerations are simultaneously satisfied. We state this theorem below and provide its proof in App. A.

Theorem 1. *For any $\mathcal{H}_0$, there exists a Kernel Sequence $\{\phi_{l^\circ}\}_{l=1}^L$ parameterized by a set of weights W_l and a set of bandwidths σ_l such that:*

I. $\mathcal{H}_L$ can approach arbitrarily close to $\mathcal{H}^*$ such that for any $L > 1$ and $\delta > 0$ we can achieve

$$\mathcal{H}^* - \mathcal{H}_L \leq \delta, \tag{6}$$

II. as $L \to \infty$, the $\mathcal{H}$-Sequence converges to the global optimum, that is

$$\lim_{L \to \infty} \mathcal{H}_L = \mathcal{H}^*, \tag{7}$$

III. the convergence is strictly monotonic where

$$\mathcal{H}_l > \mathcal{H}_{l-1} \quad \forall l \geq 1. \tag{8}$$

To simplify the proof, Thm. 1 use the average directions of each class W_s at each layer when W_s is not guaranteed to be an optimal solution. In spite of this deficit, Thm. 1 proves that a global optimum is *attainable given a minimum of two layers*, i.e., for any $L > 1$. While this remarkable feat is theoretically possible, depending on the data, the σ required for the GK may be extremely small, leading to an undesirably sharp ϕ that is overfitting the noise. Fortunately, this issue is resolved by simply spreading the monotonic improvement across more layers. We have additionally observed experimentally that by replacing the suboptimal W_s with an optimal W_l^* from ISM (where $\partial \mathcal{H} / \partial W(W_l^*) = 0$), large σ values can be used to still achieve convergence with few layers.

Relating $\mathcal{H}^*$ to Classification. As $\mathcal{H}_l \to \mathcal{H}^*$, the maximization of Eq. (5) demonstrates how we learn the similarity function. However, it may be unclear why learning this function also induces an optimal classification. To understand this deeper, let us first clarify some key notations and concepts. We refer to the image of ψ and ϕ as the Reproducing Kernel Hilbert Space (RKHS) and the image of $\mathcal{W}$ as the Images of the RKHS Dual Space (IDS). This distinction is crucial because each space dictates the geometric orientations that lead to classification. Keeping this in mind, each layer's output can be measured by the within S_w^l and between S_b^l class scatter matrices defined as

$$S_w^l = \sum_{i,j \in \mathcal{S}} W_l^T (r_i - r_j)(r_i - r_j)^T W_l \quad \text{and} \quad S_b^l = \sum_{i,j \in \mathcal{S}^c} W_l^T (r_i - r_j)(r_i - r_j)^T W_l. \tag{9}$$

These matrices are historically important [29, 30] because their trace ratio ($\mathcal{T} = \operatorname{Tr}(S_w) / \operatorname{Tr}(S_b)$) measures class separability, i.e., a small $\mathcal{T}$ implies a tight grouping of the same class under Euclidean distance. Classification is consequently achieved by mapping different classes into these spatial separations. Crucially, by maximizing $\mathcal{H}$, $\mathcal{T}$ is minimized as a byproduct. In fact, we prove that as $\mathcal{H}_l \to \mathcal{H}^*$, $\mathcal{T}$ approaches 0, and ϕ will map samples of different classes into separated points.

Note that since $\mathcal{T}$ is computed with samples from the image of $\mathcal{W}$, this particular relationship resides in IDS. Concurrently, since the inner product defines similarity in RKHS, samples are simultaneously being partitioned via the angular distance. Indeed, our proof indicates that as $\mathcal{H}_l \to \mathcal{H}^*$, RKHS samples within $\mathcal{S}$ achieves perfect alignment while samples in $\mathcal{S}^c$ become orthogonal to each other, separated by a maximum angle of $\pi/2$. This dual relationship between IDS and RKHS produces a vastly different network output right before and immediately after the final activation function, where different notions of distance (Euclidean and angular) are employed to partition samples. We formally state these results in the following theorem with the proof in App. B.

Theorem 2. *As $l \to \infty$ and $\mathcal{H}_l \to \mathcal{H}^*$, the following properties are satisfied:*

I the scatter trace ratio $\mathcal{T}$ approaches 0 where

$$\lim_{l\to\infty} \frac{\mathrm{Tr}(S_w^l)}{\mathrm{Tr}(S_b^l)} = 0 \tag{10}$$

II the Kernel Sequence converges to the following kernel:

$$\lim_{l\to\infty} \mathcal{K}(x_i, x_j)^l = \mathcal{K}^*(x_i, x_j)^l = \begin{cases} 0 & \forall i, j \in \mathcal{S}^c \\ 1 & \forall i, j \in \mathcal{S} \end{cases} . \tag{11}$$

As corollaries to Theorem 2, the resulting partition of samples under Euclidean and angular distance implicitly satisfies different classification objectives in each space. In IDS, ***KNet*** will map a dataset of τ classes into τ distinct points. While these τ points may not match the original label, this difference is inconsequential. In contrast, samples in RKHS at convergence will reside along τ orthogonal axes on a unit sphere. By realigning these results to the standard bases, solutions that simulate the softmax are generated to solve CE. Therefore, as $\mathcal{H}_l \to \mathcal{H}^*$, the maximization of Eq. (5) minimizes MSE and CE in different spaces without matching the actual labels itself: instead, we match the underlying geometry of the network output. We state the two corollaries below with their proof in App. E.

Corollary 1. *Given $\mathcal{H}_l \to \mathcal{H}^*$, the network output in IDS solves MSE via a translation of labels.*
Corollary 2. *Given $\mathcal{H}_l \to \mathcal{H}^*$, the network output in RKHS solves CE via a change of bases.*

HSIC and Regularization. Overparameterized MLPs can generalize without any explicit regularizer [20]. This observation defies classical learning theory and has been a longstanding puzzle in the research community [31, 32, 33]. Overparameterized with infinite width, ***KNet*** experimentally exhibits a resembling generalization behavior. Moreover, Ma et al. [19] have experimentally observed that HSIC can generalize even without the $W^TW = I$ constraint; we seek to better understand this phenomenon theoretically. Recently, Poggio et al. [24] have proposed that traditional MLPs generalize because gradient methods implicitly regularize the normalized weights. We discovered a similar relationship with HSIC, i.e., the objective can be reformulated to isolate out n functions $[D_1(W_l), ..., D_n(W_l)]$ that act as a penalty term during optimization. Let $\mathcal{S}|i$ be the set of samples that belongs to the i_{th} class and let $\mathcal{S}^c|i$ be its complement, then each function $D_i(W_l)$ is defined as

$$D_i(W_l) = \frac{1}{\sigma^2} \sum_{j\in\mathcal{S}|i} \Gamma_{i,j} \mathcal{K}_{W_l}(r_i, r_j) - \frac{1}{\sigma^2} \sum_{j\in\mathcal{S}^c|i} |\Gamma_{i,j}| \mathcal{K}_{W_l}(r_i, r_j). \tag{12}$$

Notice that $D_i(W_l)$ is simply Eq. (5) for a single sample scaled by $\frac{1}{\sigma^2}$. Therefore, as we identify better solutions for W_l, this leads to an increase and decrease of $\mathcal{K}_{W_l}(r_i, r_j)$ associated with $\mathcal{S}|i$ and $\mathcal{S}^c|i$ in Eq. (12), thereby increasing the size of the penalty term $D_i(W_l)$. To appreciate how $D_i(W_l)$ penalizes $\mathcal{H}$, we propose an equivalent formulation in the theorem below with its derivation in App C.

Theorem 3. *Eq. (5) is equivalent to*

$$\max_{W_l} \sum_{i,j} \frac{\Gamma_{i,j}}{\sigma^2} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i^T W_l W_l^T r_j) - \sum_i D_i(W_l) ||W_l^T r_i||_2. \tag{13}$$

Based on Thm. 3, $D_i(W_l)$ adds a negative cost to the sample norm in IDS, $||W_l^T r_i||_2$, suggesting that ISM regularizes ***KNet*** regardless of $W_l^T W_l = I$. In fact, a better W_l imposes a heavier penalty on Eq. (13) where the overall $\mathcal{H}$ may actually decrease.

Complexity Analysis. The complexity analysis of a single ISM iteration as reported by Wu et al. [17] is $O(n^3)$. Since ISM is repeated with L layers, ***KNet*** complexity is simply $O(Ln^3)$. For memory, ***KNet*** suffers from the same $O(n^2)$ restriction which all kernel methods inherit.

Limitations of Layer-Wise Kernel Dependence Networks. Although our framework presents many theoretical advantages, we caution that much more research would be required for ***KNet*** to become practically viable. While ISM resolves many existing problems, it also limits the kernels to the ISM family [18]. Therefore, it currently cannot be extended to solve the traditional activation functions such as relu and sigmoid. The computational and memory complexity is another practical obstacle. Therefore, ***KNet*** at its current maturity is *intended for analysis* and is *not yet suitable for large datasets*. Although there are already existing solutions [34, 35] to overcome these challenges, these engineering questions are topics we purposely isolate away from the theory for future research.

4 Experiments

Datasets. This work focuses on verifying the theoretical guarantees of a greedily trained network against traditional MLPs of comparable complexity trained by BP. Specifically, we confirm the theoretical properties of *KNet* using three synthetic (Random, Adversarial and Spiral) and five popular UCI benchmark datasets: wine, cancer, car, divorce, and face [36]. They are included along with the source code in the supplementary, and their comprehensive download link and statistics are in App. F.

Evaluation Metrics. To evaluate the central claim that MLPs can be solved greedily, we report $\mathcal{H}^*$ at convergence along with the training/test accuracy for each dataset. Here, $\mathcal{H}^*$ is normalized to the range between 0 to 1 using the method proposed by Cortes et al. [37]. To corroborate Corollaries 1 and 2, we also record MSE and CE. To evaluate the sample geometry predicted by Eq. (10), we recorded the scatter trace ratio $\mathcal{T}$ to measure the compactness of samples within and between classes. The angular distance between samples in $\mathcal{S}$ and $\mathcal{S}^c$ as predicted by Eq. (11) is evaluated with the Cosine Similarity Ratio (C). The equations for $\mathcal{H}^*$ and C are

$$\mathcal{H}^* = \frac{\mathcal{H}(\phi(X), Y)}{\sqrt{\mathcal{H}(\phi(X), \phi(X))\mathcal{H}(Y, Y)}} \quad \text{and} \quad C = \frac{\sum_{i,j \in \mathcal{S}^c} \langle \phi(x_i), \phi(x_j) \rangle}{\sum_{i,j \in \mathcal{S}} \langle \phi(x_i), \phi(x_j) \rangle}. \tag{14}$$

Experiment Settings. The width of the network is set by ISM to keep 90% of the data variance. The RFF width is set to 300 for all datasets and the σ_l that maximizes $\mathcal{H}^*$ is chosen. The convergence threshold for $\mathcal{H}$-*Sequence* is set at $\mathcal{H}_l > 0.99$. The network structures discovered by ISM for every dataset are recorded and provided in App. G. The MLPs that use MSE and CE have weights initialized via the Kaiming method [38]. All datasets are centered to 0 and scaled to a standard deviation of 1. All sources are written in Python using Numpy, Sklearn and Pytorch [39, 40, 41]. All experiments were conducted on an Intel Xeon(R) CPU E5-2630 v3 @ 2.40GHz x 16 with 16 total cores.

Algorithm 1 *KNet* Algorithm

Input : Data $X \in \mathbb{R}^{n \times d}$, Label $Y \in \mathbb{R}^{n \times \tau}$
Output : Network weights $W_1, ..., W_L$
while $\mathcal{H}_l$ < *0.99* **do**
  Use the output of last layer as input
  Add a new layer
  Initialize layer weight with Eq. (3)
  while $\mathcal{Q}_{l^i} \not\approx \mathcal{Q}_{l^{i+1}}$ **do**
    Update $\mathcal{Q}_{l^i} \rightarrow \mathcal{Q}_{l^{i+1}}$ with Eq. (4)
  end
  W_l = Most dominant eigenvector of $\mathcal{Q}_l^*$
end

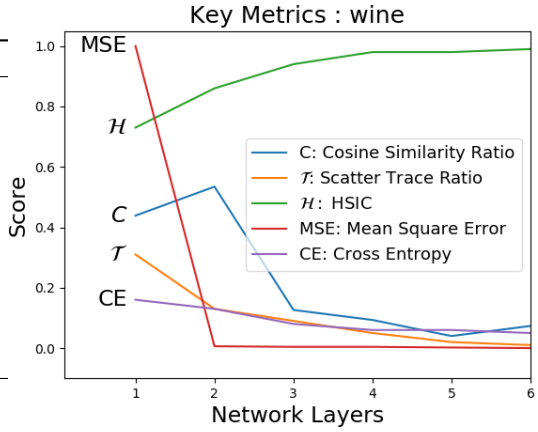

Figure 1: Key evaluation metrics at each layer.

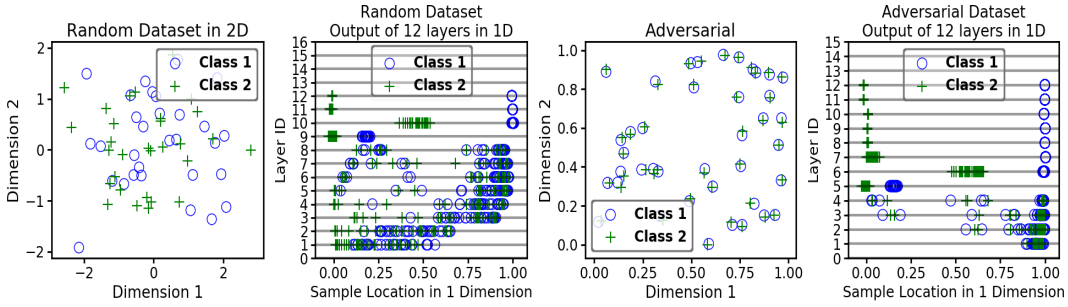

Figure 2: Simulation of Thm. 1 on Random and Adversarial datasets. The 2D representation is shown, and next to it, the 1D output of each layer is displayed over each line. Both datasets achieved the global optimum $\mathcal{H}^*$ at the 12_{th} layers. Refer to App. H for additional results.

Experimental Results. Since Thm. 1 guarantees an optimal convergence for *any* dataset with a suboptimal W_s, we designed an Adversarial dataset to trick the network, i.e., the samples pairs in $\mathcal{S}^c$ are significantly closer than samples pairs in $\mathcal{S}$. We next designed a Random dataset with completely random labels. We then simulated Thm. 1 in Python and plot out the sample behavior in Fig. 2. The original 2-dimensional data is shown next to its 1-dimensional IDS results: each line represents the 1D output at that layer. As predicted by the theorem, our network converged at the 12th layer and perfectly separated the samples based on labels. We emphasize that these achievements are acquired purely from the simulation of Thm. 1 without resorting to $\sigma \approx 0$ while using a *suboptimal* solution W_s. Namely, the smallest σ values used are 0.15 and 0.03 for Random and Adversarial respectively.

Using the optimal W^* from ISM, we next conduct 10-fold cross-validation across all 8 datasets and reported their mean and the standard deviation for all key metrics. The random and non-random datasets are visually separated. Once our model is trained and has learned its structure, we use the same depth and width to train 2 additional MLPs via SGD, where instead of HSIC, MSE and CE are used as the empirical risk. The results are listed in Table 1 with the best outcome in bold.

Can $\mathcal{H}$-*Sequence* be optimized greedily? The $\mathcal{H}^*$ column in Table 1 consistently reports results that converge near its theoretical maximum value of 1, thereby corroborating with Thm. 1. As predicted by Thm. 2, we also report high training accuracies as $\mathcal{H}_l \to \mathcal{H}^*$. Given the overfitting results from Fig. 2, will our network generalize? Since smooth mappings are associated with better generalization, we also report the smallest σ value used for each network to highlight the smoothness of ϕ learned by ISM. Correspondingly, with the exception of the two random datasets, our test accuracy consistently performed well across all datasets. *KNet* further differentiates itself on a high dimension Face dataset where it was the only method that avoided overfitting. While we cannot definitively attribute the impressive test results to Thm. 3, the experimental evidence appears to be aligned with its implication.

	obj	$\sigma\uparrow$	$L\downarrow$	Train Acc $\uparrow$	Test Acc $\uparrow$	Time(s) $\downarrow$	$\mathcal{H}^*\uparrow$	MSE $\downarrow$	CE $\downarrow$	$C\downarrow$	$\mathcal{T}\downarrow$
random	$\mathcal{H}$	0.38	3.30 ± 0.64	**1.00 ± 0.00**	0.38 ± 0.21	**0.40 ± 0.37**	**1.00 ± 0.01**	**0.00 ± 0.01**	0.05 ± 0.00	**0.00 ± 0.06**	0.02 ± 0.0
	CE	-	3.30 ± 0.64	**1.00 ± 0.00**	0.48 ± 0.17	25.07 ± 5.55	**1.00 ± 0.00**	10.61 ± 11.52	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	3.30 ± 0.64	0.98 ± 0.04	**0.63 ± 0.21**	23.58 ± 8.38	0.93 ± 0.12	0.02 ± 0.04	0.74 ± 0.03	0.04 ± 0.04	0.08 ± 0.1
adver	$\mathcal{H}$	0.5	3.60 ± 0.92	**1.00 ± 0.00**	**0.38 ± 0.10**	**0.52 ± 0.51**	**1.00 ± 0.00**	**0.00 ± 0.00**	**0.04 ± 0.00**	**0.01 ± 0.08**	**0.01 ± 0.0**
	CE	-	3.60 ± 0.92	0.59 ± 0.04	0.29 ± 0.15	69.54 ± 24.14	0.10 ± 0.07	0.65 ± 0.16	0.63 ± 0.04	0.98 ± 0.03	0.92 ± 0.0
	MSE	-	3.60 ± 0.92	0.56 ± 0.02	0.32 ± 0.20	113.75 ± 21.71	0.02 ± 0.01	0.24 ± 0.01	0.70 ± 0.00	0.99 ± 0.02	0.95 ± 0.0
spiral	$\mathcal{H}$	0.46	5.10 ± 0.30	**1.00 ± 0.00**	**1.00 ± 0.00**	**0.87 ± 0.08**	0.98 ± 0.01	0.01 ± 0.00	0.02 ± 0.01	0.04 ± 0.03	0.02 ± 0.0
	CE	-	5.10 ± 0.30	**1.00 ± 0.00**	**1.00 ± 0.00**	11.59 ± 5.52	**1.00 ± 0.00**	57.08 ± 31.25	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	5.10 ± 0.30	**1.00 ± 0.00**	0.99 ± 0.01	456.77 ± 78.83	**1.00 ± 0.00**	**0.00 ± 0.00**	1.11 ± 0.04	0.40 ± 0.01	**0.00 ± 0.0**
wine	$\mathcal{H}$	0.47	6.10 ± 0.54	0.99 ± 0.00	**0.97 ± 0.05**	**0.28 ± 0.04**	0.98 ± 0.01	0.01 ± 0.00	0.07 ± 0.01	0.04 ± 0.03	0.02 ± 0.0
	CE	-	6.10 ± 0.54	**1.00 ± 0.00**	0.94 ± 0.06	3.30 ± 1.24	**1.00 ± 0.00**	40.33 ± 35.5	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	6.10 ± 0.54	**1.00 ± 0.00**	0.89 ± 0.17	77.45 ± 45.40	**1.00 ± 0.00**	**0.00 ± 0.00**	1.15 ± 0.07	0.49 ± 0.02	**0.00 ± 0.0**
cancer	$\mathcal{H}$	0.39	8.10 ± 0.83	0.99 ± 0.00	**0.97 ± 0.02**	**2.58 ± 1.07**	0.96 ± 0.01	0.02 ± 0.01	0.04 ± 0.01	0.02 ± 0.04	0.04 ± 0.0
	CE	-	8.10 ± 0.83	**1.00 ± 0.00**	**0.97 ± 0.01**	82.03 ± 35.15	**1.00 ± 0.00**	2330 ± 2915	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	8.10 ± 0.83	**1.00 ± 0.00**	**0.97 ± 0.03**	151.81 ± 27.27	**1.00 ± 0.00**	**0.00 ± 0.00**	0.66 ± 0.06	**0.00 ± 0.0**	**0.00 ± 0.0**
car	$\mathcal{H}$	0.23	4.90 ± 0.30	**1.00 ± 0.00**	**1.00 ± 0.01**	**1.51 ± 0.35**	0.99 ± 0.00	**0.00 ± 0.00**	0.01 ± 0.00	0.04 ± 0.03	0.01 ± 0.0
	CE	-	4.90 ± 0.30	**1.00 ± 0.00**	**1.00 ± 0.00**	25.79 ± 18.86	**1.00 ± 0.00**	225.11 ± 253	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	4.90 ± 0.30	**1.00 ± 0.00**	**1.00 ± 0.00**	503.96 ± 116.64	**1.00 ± 0.00**	**0.00 ± 0.00**	1.12 ± 0.07	0.40 ± 0.00	**0.00 ± 0.0**
face	$\mathcal{H}$	0.44	4.00 ± 0.00	**1.00 ± 0.00**	**0.99 ± 0.01**	**0.78 ± 0.08**	0.97 ± 0.00	**0.00 ± 0.00**	0.17 ± 0.00	0.01 ± 0.00	**0.00 ± 0.0**
	CE	-	4.00 ± 0.00	**1.00 ± 0.00**	0.79 ± 0.31	23.70 ± 8.85	**1.00 ± 0.00**	16099 ± 16330	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	4.00 ± 0.00	0.92 ± 0.10	0.52 ± 0.26	745.17 ± 281.56	0.94 ± 0.07	0.11 ± 0.12	3.50 ± 0.28	0.72 ± 0.01	**0.00 ± 0.0**
divorce	$\mathcal{H}$	0.41	4.10 ± 0.54	0.99 ± 0.01	0.98 ± 0.02	**0.71 ± 0.41**	0.99 ± 0.01	0.01 ± 0.01	0.03 ± 0.00	**0.00 ± 0.05**	0.02 ± 0.0
	CE	-	4.10 ± 0.54	**1.00 ± 0.00**	**0.99 ± 0.02**	2.62 ± 1.21	**1.00 ± 0.00**	14.11 ± 12.32	**0.00 ± 0.00**	**0.00 ± 0.00**	**0.00 ± 0.0**
	MSE	-	4.10 ± 0.54	**1.00 ± 0.00**	0.97 ± 0.03	47.89 ± 24.31	**1.00 ± 0.00**	**0.00 ± 0.00**	0.73 ± 0.07	**0.00 ± 0.01**	0.01 ± 0.0

Table 1: Each dataset contains 3 rows comparing the greedily trained *KNet* using $\mathcal{H}$ against traditional MLPs trained using MSE and CE via SGD given the same network width and depth. The best results are in bold with $\uparrow$ / $\downarrow$ indicating larger/smaller values preferred.

Since Thm. 1 also claims that we can achieve $\mathcal{H}^* - \mathcal{H}_l < \delta$ in finite number of layers, we include in Table 1 the average length of the $\mathcal{H}$-*Sequence* (L). The table suggests that the $\mathcal{H}$-*Sequence* converges quickly with 9 layers as the deepest network. Additionally, notice in Fig. 2 where 12 layers are required for the suboptimal W_s to convergence. In contrast, ISM used much smoother and larger σs (0.38 and 0.5 $\gg$ 0.15 and 0.03) and only requires 3 layers to achieve the same result.

The execution time for each objective is also recorded for reference in Table 1. Since *KNet* can be solved via a single forward pass while SGD requires many iterations of BP, *KNet* should be faster. The Time column of Table 1 confirmed this expectation by a wide margin. The biggest difference can be observed by comparing the face dataset, $\mathcal{H}$ finished with 0.78 seconds while MSE required

745 seconds; that is almost 1000 times difference. While the execution times reflect our expectation, techniques that vastly accelerate kernel computations [34, 35] would be required for larger datasets.

Predicted by Thm. 2, *KNet* induces low $\mathcal{T}$ and C as shown in Table 1, implying that samples in $\mathcal{S}$ and $\mathcal{S}^c$ are being pulled together and pushed apart based on Euclidean and angular distance in IDS and RKHS. Given this geometry, will its optimal arguments also induce a low MSE and CE? We evaluate these predictions from Corollaries 1 and 2 by keeping the same network weights while replacing the final objective with MSE and CE. Our corroborating results are highlighted in the columns of MSE and CE in Table 1. Interestingly, while using HSIC induces a low MSE and CE, training via BP using MSE or CE does not necessarily translate to good results for each other.

Within the network, Fig. 1 plots out all key metrics at each layer during training. Here, the $\mathcal{H}$-*Sequence* is clearly monotonic and converging towards a global optimal of 1. Moreover, the trends for $\mathcal{T}$ and C indicate an incremental clustering of samples into separate partitions. We are unsure why C consistently forms a hunchback pattern, this implies an initial expansion in the dimension of the data following compression. However, we note that this behavior is also observed with traditional networks by Ansuini et al. [42]. Corresponding to low $\mathcal{T}$ and C values, the low MSE and CE errors at convergence further reinforces the claims of Corollaries 1 and 2. Note that these identical patterns are consistent and repeatable across all datasets as shown in App. J.

We lastly highlight a visual pattern for the *Kernel Sequence* in Fig. 3. We rearrange the samples of the same class to be adjacent to each other. This allows us to quickly evaluate the kernel quality via its block diagonal structure. Since GK is restricted to values between 0 and 1, we let white and dark blue be 0 and 1 respectively where the gradients reflect values in between. Our proof predicts that the *Kernel Sequence* should converge to the optimal kernel $\mathcal{K}^*$, i.e., the *Kernel Sequence* should evolve from an uninformative kernel into a highly discriminating kernel of perfect block diagonal structures. Corresponding to the top row, the bottom row plots out the samples in IDS at each layer. As predicted by Thm. 2, the samples of the same class incrementally converge towards a single point in IDS. Again, this pattern is observable on all datasets, and the complete collection of the *kernel sequences* for each dataset can be found in App. I.

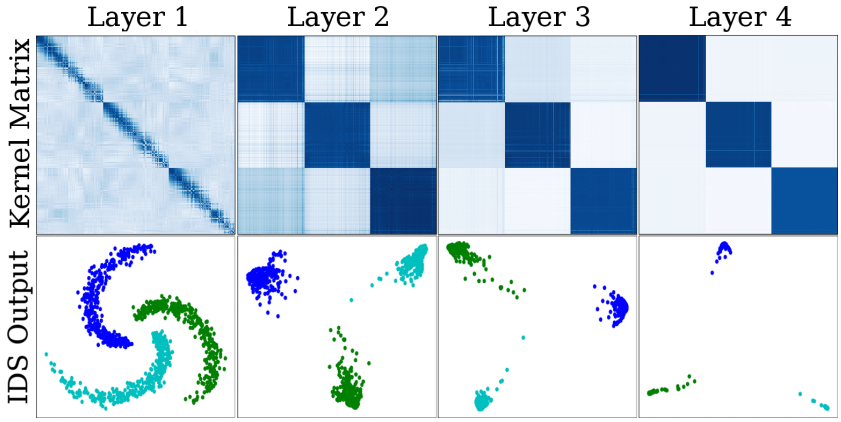

Figure 3: A visual confirmation of Thm. 2. The kernel matrices per layer produced by the *Kernel Sequence* are displayed in the top row with their corresponding outputs in IDS in the bottom row.

Conclusion. We have presented a new model of MLP for classification that bypasses BP and SGD. Our model, *KNet*, is guaranteed to reach the global optimum of HSIC in finite steps. The resulting geometric orientation of the samples minimizes the scatter ratio while produces an optimal kernel, $\mathcal{K}^*$. This results in a geometric orientation that consequently solves MSE and CE in different spaces. Indeed, these patterns are predictable by our theorems and experimentally reproducible. Therefore, *KNet* opens the door to a new perspective to analyze MLPs.

5 Broader Impact

In our attempt to explain the nature of Neural Networks, we discovered an alternative perspective to solve them, a kernel interpretation where all of its internal mechanisms are well understood. Although more research is required to make our method practical to larger datasets, we are confident that these obstacles will be solved in the near future. Additionally, a convolutional extension of our work is definitely feasible. Therefore, a logical direction of our research suggests that it will lead to significantly faster classifiers, paving the way for better medical applications, better self-driving cars, and better weather prediction.

While we cannot predict how others might use our work, however, we can make predictions based on the direction of our research. We hope that given this technology, medical researchers can train better models to predict cancer and other diseases. Environmental scientists might be able to better measure pollution, air quality, and water quality. We want better micro-lending algorithms that allow economically depressed regions of the world to thrive. We hope our research might even lead to better algorithms that discover vaccines to mitigate the impact of a pandemic.

As researchers, in our quest for the truth, we hope to remake the world a better place. Yet, by sharing our discoveries with the world, we lose control over its impact. Good intention alone cannot prevent the military or any terrorist organizations from leveraging the technology to create killer robots. Nor can it prevent totalitarian states from using it to suppress its own citizens. The same convolutional adaptation we hope to use to improve cancer prediction can be used by countries to identify citizens who criticize the government, locking them away without due process. Even in the best case where technology is developed without any malice, we see the potential of millions of jobs being automated. Worse yet, it is always those who are least able to adapt to new jobs, the most vulnerable, that are being replaced by our inventions.

None of these problems are easy to solve. Yet, as we see the mounting problems of the world, many of them caused by the advancement of our research, how much responsibility should we bear as individual researchers? Is the addition of the broader impact section sufficient? Or should we consider a version of the Hippocratic oath for our profession? For the betterment of the world, we must begin the dialog to answer these questions. As researchers and as citizens of the world, let us take on more responsibility to mitigate the worst impact of our work on society.

References

[1] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[2] Balázs Csanád Csáji. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24:48, 2001.

[3] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[4] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[5] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in neural information processing systems*, pages 6231–6239, 2017.

[6] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *ArXiv*, abs/1711.00165, 2018.

[7] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[8] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[9] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. *arXiv preprint arXiv:1902.06853*, 2019.

[10] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

[11] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

[12] David Duvenaud, Oren Rippel, Ryan Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, pages 202–210, 2014.

[13] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[14] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

[15] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *ICML*, 2018.

[16] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005.

[17] Chieh Wu, Stratis Ioannidis, Mario Sznaier, Xiangyu Li, David Kaeli, and Jennifer Dy. Iterative spectral method for alternative clustering. In *International Conference on Artificial Intelligence and Statistics*, pages 115–123, 2018.

[18] Chieh Wu, Jared Miller, Yale Chang, Mario Sznaier, and Jennifer G. Dy. Solving interpretable kernel dimensionality reduction. In *NeurIPS*, 2019.

[19] Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. The hsic bottleneck: Deep learning without back-propagation. *arXiv preprint arXiv:1908.01580*, 2019.

[20] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[21] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 233–242. JMLR. org, 2017.

[22] Giacomo De Palma, Bobak Kiani, and Seth Lloyd. Random deep neural networks are biased towards simple functions. In *Advances in Neural Information Processing Systems*, pages 1962–1974, 2019.

[23] Guillermo Valle-Pérez, Chico Q Camargo, and Ard A Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[24] Tomaso Poggio, Qianli Liao, and Andrzej Banburski. Complexity control by gradient descent in deep networks. *Nature Communications*, 11(1):1–5, 2020.

[25] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*, 2018.

[26] Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes. *arXiv preprint arXiv:1906.04724*, 2019.

[27] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[28] John Bibby. Axiomatisations of the average and a further generalisation of monotonic sequences. *Glasgow Mathematical Journal*, 15(1):63–65, 1974.

[29] Ronald A Fisher. The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188, 1936.

[30] Geoffrey J McLachlan. *Discriminant analysis and statistical pattern recognition*, volume 544. John Wiley & Sons, 2004.

[31] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10835–10845, 2019.

[32] Alon Brutzkus, Amir Globerson, Eran Malach, and Shai Shalev-Shwartz. Sgd learns over-parameterized networks that provably generalize on linearly separable data. *arXiv preprint arXiv:1710.10174*, 2017.

[33] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Advances in neural information processing systems*, pages 6155–6166, 2019.

[34] Ke Wang, Geoff Pleiss, Jacob Gardner, Stephen Tyree, Kilian Q Weinberger, and Andrew Gordon Wilson. Exact gaussian processes on a million data points. In *Advances in Neural Information Processing Systems*, pages 14622–14632, 2019.

[35] Alessandro Rudi, Luigi Carratino, and Lorenzo Rosasco. Falkon: An optimal large scale kernel method. In *Advances in Neural Information Processing Systems*, pages 3888–3898, 2017.

[36] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[37] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(Mar):795–828, 2012.

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[39] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. URL `http://www.scipy.org/`. [Online; accessed <today>].

[40] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[41] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[42] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. *arXiv preprint arXiv:1905.12784*, 2019.

[43] Grasgoire Montavon, Mikio L Braun, and Klaus-Robert Matller. Kernel analysis of deep networks. *Journal of Machine Learning Research*, 12(Sep):2563–2581, 2011.

Appendix A Proof for Theorem 1

Theorem 1: *For any $\mathcal{H}_0$, there exists a Kernel Sequence $\{\phi_{l^\circ}\}_{l=1}^{L}$ parameterized by a set of weights W_l and a set of bandwidths σ_l such that:*

I. $\mathcal{H}_L$ can approach arbitrarily close to $\mathcal{H}^*$ such that for any $L > 1$ and $\delta > 0$ we can achieve

$$\mathcal{H}^* - \mathcal{H}_L \leq \delta, \tag{15}$$

II. as $L \to \infty$, the $\mathcal{H}$-*Sequence* converges to the global optimum, that is

$$\lim_{L\to\infty} \mathcal{H}_L = \mathcal{H}^*, \tag{16}$$

III. the convergence is monotonic where

$$\mathcal{H}_l > \mathcal{H}_{l-1} \quad \forall l. \tag{17}$$

Lemma 1. *Given σ_0 and σ_1 as the σ values from the last layer and the current layer, then there exists a lower bound for $\mathcal{H}_l$, denoted as $\mathscr{L}(\sigma_0, \sigma_1)$ such that*

$$\mathcal{H}_l \geq \mathscr{L}(\sigma_0, \sigma_1). \tag{18}$$

Basic Background, Assumptions, and Notations.

1. The simulation of this theorem for Adversarial and Random data is also publicly available on `https://github.com/anonymous`.
2. Here we show that this bound can be established given the last 2 layers.
3. σ_0 is the σ value of the previous layer
4. σ_1 is the σ value of the current layer
5. τ is the number of classes
6. n is total number of samples
7. n_i is number of samples in the i^{th} class
8. $\mathcal{S}$ is a set of all i, j sample pairs where r_i and r_j belong to the same class.
9. $\mathcal{S}^c$ is a set of all i, j sample pairs where r_i and r_j belong to different same classes.
10. $\mathcal{S}^\beta$ is a set of all i, j sample pairs that belongs to the same β^{th} classes.
11. $r_i^{(\alpha)}$ is the i^{th} sample in the α^{th} class among τ classes.
12. We assume no $r_i \neq r_j$ pair are equal $\forall i \neq j$.
13. Among all $r_i \neq r_j$ pairs, there exists an optimal r_i^*, r_j^* pair where $\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle$ $\forall r_i \neq r_i^*$ and $r_j \neq r_j^*$. We denote this maximum inner product as
$$u_{\sigma_0} = \langle r_i^*, r_j^* \rangle. \tag{19}$$
14. In *KNet*, each r_i sample is assumed to be a sample in the RKHS of the Gaussian kernel, therefore all inner products are bounded such that
$$0 \leq \langle r_i, r_j \rangle \leq u_{\sigma_0}. \tag{20}$$
15. We let W be
$$W_s = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & \dots & \sum_\iota r_\iota^{(\tau)} \end{bmatrix}. \tag{21}$$
Instead of using an optimal W^* defined as $W^* = \arg\max_W H_l(W)$, we use a suboptimal W_s where each dimension is simply the average direction of each class: $\frac{1}{\sqrt{\zeta}}$ is a normalizing constant $\zeta = ||W_s||_2^2$ that ensures $W_s^T W_s = I$. By using W_s, this implies that the $\mathcal{H}$ we obtain is already a lower bound compare $\mathcal{H}$ obtained by W^*. But, we will use this suboptimal W_s to identify an even lower bound. Note that based on the definition W^*, we have the property $\mathcal{H}(W^*) \geq \mathcal{H}(W)\, \forall W$.

16. We note that the objective $\mathcal{H}$ is

$$\mathcal{H} = \underbrace{\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma_1^2}}}_{\mathscr{W}} - \underbrace{\sum_{i,j\in\mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma_1^2}}}_{\mathscr{B}} \tag{22}$$

where we let $\mathscr{W}$ be the summation of terms associated with the within cluster pairs, and let $\mathscr{B}$ be the summation of terms associated with the between cluster pairs.

Proof.

The equation is divided into smaller parts organized into multiple sections.

For sample pairs in $\mathcal{S}$. The first portion of the function can be split into multiple classes where

$$\mathscr{W} = \underbrace{\sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{(r_i^{(1)} - r_j^{(1)})^T WW^T (r_i^{(1)} - r_j^{(1)})}{2\sigma_1^2}}}_{\mathscr{W}_1} + ... + \underbrace{\sum_{\mathcal{S}^\tau} \Gamma_{i,j} e^{-\frac{(r_i^{(\tau)} - r_j^{(\tau)})^T WW^T (r_i^{(\tau)} - r_j^{(\tau)})}{2\sigma_1^2}}}_{\mathscr{W}_\tau} \tag{23}$$

Realize that to find the lower bound, we need to determine the minimum possible value of each term which translates to **maximum** possible value of each exponent. Without of loss of generality we can find the lower bound for one term and generalize its results to other terms due to their similarity. Let us focus on the numerator of the exponent from $\mathscr{W}_1$. Given W_s as W, our goal is identify the **maximum** possible value for

$$\underbrace{(r_i^{(1)} - r_j^{(1)})^T W}_{\Pi_1} \underbrace{W^T (r_i^{(1)} - r_j^{(1)})}_{\Pi_2}. \tag{24}$$

Zoom in further by looking only at Π_1, we have the following relationships

$$\Pi_1 = \underbrace{r_i^{(1)^T} W}_{\xi_1} - \underbrace{r_j^{(1)^T} W}_{\xi_2} \tag{25}$$

$$\xi_1 = \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{26}$$

$$= \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{27}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} r_j^{(1)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{28}$$

$$= \frac{1}{\sqrt{\zeta}} r_j^{(1)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{29}$$

By knowing that the inner product is constrained between $[0, u_{\sigma_0}]$, we know the maximum possible value for ξ_1 and the minimum possible value for ξ_2 to be

$$\xi_1 = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} 1 + (n_1 - 1)u_{\sigma_0} & n_2 u_{\sigma_0} & n_3 u_{\sigma_0} & ... & n_\tau u_{\sigma_0} \end{bmatrix} \tag{30}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} 1 & 0 & 0 & ... & 0 \end{bmatrix}. \tag{31}$$

Which leads to

$$\Pi_1 = \frac{1}{\sqrt{\zeta}} (\xi_1 - \xi_2) = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} (n_1 - 1)u_{\sigma_0} & n_2 u_{\sigma_0} & n_3 u_{\sigma_0} & ... & n_\tau u_{\sigma_0} \end{bmatrix} \tag{32}$$

Since $\Pi_2^T = \Pi_1$ we have

$$\Pi_1 \Pi_2 = \frac{1}{\zeta} [(n_1 - 1)^2 u_{\sigma_0}^2 + n_2^2 u_{\sigma_0}^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2] \tag{33}$$

$$= \frac{1}{\zeta} [(n_1 - 1)^2 + n_2^2 + n_3^2 + ... + n_\tau^2] u_{\sigma_0}^2 \tag{34}$$

The lower bound for just the $\mathscr{W}_1$ term emerges as

$$\mathscr{W}_1 \geq \sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{[(n_1-1)^2+n_2^2+n_3^2+\ldots+n_\tau^2]u_{\sigma_0}^2}{2\zeta\sigma_1^2}}. \tag{35}$$

To further condense the notation, we define the following constant

$$\mathscr{N}_g = \frac{1}{2\zeta}[n_1^2 + n_2^2 + \ldots + (n_g - 1)^2 + \ldots + n_\tau^2]. \tag{36}$$

Therefore, the lower bound for $\mathscr{W}_1$ can be simplified as

$$\mathscr{W}_1 \geq \sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{\mathscr{N}_1 u_{\sigma_0}^2}{\sigma_1^2}} \tag{37}$$

and the general pattern for any $\mathscr{W}_g$ becomes

$$\mathscr{W}_g \geq \sum_{\mathcal{S}^i} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}. \tag{38}$$

The lower bound for the entire set of $\mathcal{S}$ then becomes

$$\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T WW^T(r_i-r_j)}{2\sigma_1^2}} = \mathscr{W}_1 + \ldots + \mathscr{W}_\tau \geq \underbrace{\sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}}_{\text{Lower bound}}. \tag{39}$$

For sample pairs in $\mathcal{S}^c$. To simplify the notation, we note that

$$-\mathscr{B}_{g_1,g_2} = -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{(r_i^{(g_1)}-r_j^{(g_2)})^T WW^T(r_i^{(g_1)}-r_j^{(g_2)})}{2\sigma_1^2}} \tag{40}$$

$$= -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T((r_i^{(g_1)}-r_j^{(g_1)})((r_i^{(g_1)}-r_j^{(g_2)}))^T W)}{2\sigma_1^2}} \tag{41}$$

$$= -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(g_1,g_2)} W)}{2\sigma_1^2}} \tag{42}$$

$$\tag{43}$$

We now derived the lower bound for the sample pairs in $\mathcal{S}^c$. We start by writing out the entire summation sequence for $\mathscr{B}$.

$$\begin{aligned}
\mathscr{B} = &\underbrace{-\sum_{i\in\mathcal{S}^1}\sum_{j\in\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(1,2)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{1,2}} - \underbrace{\ldots}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^1}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(1,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{1,\tau}} \\
&\underbrace{-\sum_{i\in\mathcal{S}^2}\sum_{j\in\mathcal{S}^1} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(2,1)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{2,1}} - \underbrace{\ldots}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^2}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(2,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{2,\tau}} \\
&\ldots \\
&\underbrace{-\sum_{i\in\mathcal{S}^\tau}\sum_{j\in\mathcal{S}^1} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(\tau,1)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{\tau,1}} - \underbrace{\ldots}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^{\tau-1}}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(\tau-1,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{\tau-1,\tau}}
\end{aligned} \tag{44}$$

Using a similar approach with the terms from $\mathscr{W}$, note that $\mathscr{B}$ is a negative value, so we need to maximize this term to obtain a lower bound. Consequently, the key is to determine the **minimal** possible values for each exponent term. Since every one of them will behave very similarly, we can simply look at the numerator of the exponent from $\mathscr{B}_{1,2}$ and then arrive to a more general conclusion. Given W_s as W, our goal is to identify the **minimal** possible value for

$$\underbrace{(r_i^{(1)} - r_j^{(2)})^T W}_{\Pi_1} \underbrace{W^T (r_i^{(1)} - r_j^{(2)})}_{\Pi_2}. \tag{45}$$

Zoom in further by looking only at Π_1, we have the following relationships

$$\Pi_1 = \underbrace{r_i^{(1)^T} W}_{\xi_1} - \underbrace{r_j^{(2)^T} W}_{\xi_2} \tag{46}$$

$$\xi_1 = \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{47}$$

$$= \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{48}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} r_j^{(2)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{49}$$

$$= \frac{1}{\sqrt{\zeta}} r_j^{(2)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{50}$$

By knowing that the inner product is constrained between $[0, u_{\sigma_0}]$, we know the **minimum** possible value for ξ_1 and the **maximum** possible value for ξ_2 to be

$$\xi_1 = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} 1 & 0 & 0 & ... & 0 \end{bmatrix} \tag{51}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} n_1 u_{\sigma_0} & 1 + (n_2 - 1) u_{\sigma_0} & n_3 u_{\sigma_0} & ... & n_\tau u_{\sigma_0} \end{bmatrix} \tag{52}$$

Which leads to

$$\Pi_1 = \frac{1}{\sqrt{\zeta}} (\xi_1 - \xi_2) = \frac{1}{\sqrt{\zeta}} \begin{bmatrix} 1 - n_1 u_{\sigma_0} & -(1 + (n_2 - 1) u_{\sigma_0}) & -n_3 u_{\sigma_0} & ... & -n_\tau u_{\sigma_0} \end{bmatrix} \tag{53}$$

Since $\Pi_2^T = \Pi_1$ we have

$$\Pi_1 \Pi_2 = \frac{1}{\zeta} [(1 - n_1 u_{\sigma_0})^2 + (1 + (n_2 - 1) u_{\sigma_0})^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2]. \tag{54}$$

The lower bound for just the $\mathscr{B}_{1,2}$ term emerges as

$$-\mathscr{B}_{1,2} \geq -\sum_{\mathcal{S}^1} \sum_{\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{(1 - n_1 u_{\sigma_0})^2 + (1 + (n_2 - 1) u_{\sigma_0})^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2}{2\zeta\sigma_1^2}}. \tag{55}$$

To further condense the notation, we define the following function

$$\begin{aligned} \mathscr{N}_{g_1, g_2}(u_{\sigma_0}) = \frac{1}{2\zeta} [&n_1^2 u_{\sigma_0}^2 + n_2^2 u_{\sigma_0}^2 + ... \\ &+ (1 - n_{g_1} u_{\sigma_0})^2 + ... + (1 + (n_{g_2} - 1) u_{\sigma_0})^2 \\ &+ ... + n_\tau^2 u_{\sigma_0}^2]. \end{aligned} \tag{56}$$

Note that while for $\mathcal{S}$, the u_{σ_0} term can be separated out. But here, we cannot, and therefore $\mathscr{N}$ here must be a function of u_{σ_0}. Therefore, the lower bound for $\mathscr{B}_{1,2}$ can be simplified into

$$-\mathscr{B}_{1,2} \geq -\sum_{\mathcal{S}^1} \sum_{\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{1,2}(u_{\sigma_0})}{\sigma_1^2}} \tag{57}$$

and the general pattern for any $\mathscr{B}_{g_1,g_2}$ becomes

$$-\mathscr{B}_{g_1,g_2} \geq -\sum_{\mathcal{S}^{g1}}\sum_{\mathcal{S}^{g2}} \Gamma_{i,j} e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}. \tag{58}$$

The lower bound for the entire set of $\mathcal{S}^c$ then becomes

$$-\sum_{i,j\in\mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i-r_j)^T W W^T (r_i-r_j)}{2\sigma_1^2}} = -\mathscr{B}_{1,2} - \mathscr{B}_{1,3} - ... - \mathscr{B}_{\tau-1,\tau} \tag{59}$$

$$\geq \underbrace{-\sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}}_{\text{Lower bound}}. \tag{60}$$

Putting $\mathcal{S}$ and $\mathcal{S}^c$ Together.

$$\mathcal{H} = \mathscr{W} + \mathscr{B} \tag{61}$$

$$\geq \underbrace{\sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}}_{\text{Lower bound of } \mathscr{W}} - \underbrace{\sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}}_{\text{Lower bound of } \mathscr{B}}. \tag{62}$$

Therefore, we have identified a lower bound that is a function of σ_0 and σ_1 where

$$\mathscr{L}(\sigma_0,\sigma_1) = \sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}} - \sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}. \tag{63}$$

From the lower bound, it is obvious why it is a function of σ_1. The lower bound is also a function of σ_0 because u_{σ_0} is actually a function of σ_0. To specifically clarify this point, we have the next lemma.

□

Lemma 2. *The u_{σ_0} used in Lemma 1 is a function of σ_0 where u_{σ_0} approaches to zero as σ_0 approaches to zero, i.e.*

$$\lim_{\sigma_0\to 0} u_{\sigma_0} = 0. \tag{64}$$

Assumptions and Notations.

1. We use Fig. 4 to help clarify the notations. We here only look at the last 2 layers.
2. We let $\mathcal{H}_0$ be the $\mathcal{H}$ of the last layer, and $\mathcal{H}_1$, the $\mathcal{H}$ of the current layer.
3. The input of the data is X with each sample as x_i, and the output of the previous layer are denoted as r_i. ψ_{σ_0} is the feature map of the previous layer using σ_0 and ψ_{σ_1} corresponds to the current layer.

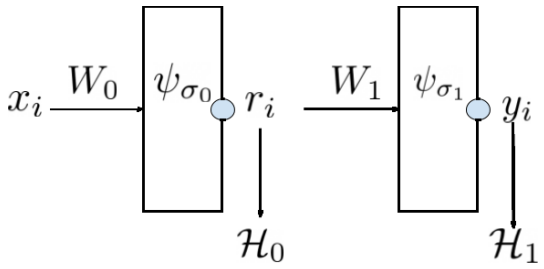

Figure 4: Figure of a 2 layer network.

4. As defined from Lemma 1, among all $r_i \neq r_j$ pairs, there exists an optimal r_i^*, r_j^* pair where $\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle \; \forall r_i \neq r_i^*$ and $r_j \neq r_j^*$. We denote this maximum inner product as

$$u_{\sigma_0} = \langle r_i^*, r_j^* \rangle. \tag{65}$$

Proof.

Given Fig. 4, the equation for $\mathcal{H}_0$ is

$$\mathcal{H}_0 = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(x_i - x_j)^T W W^T (x_i - x_j)}{2\sigma_0^2}} - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(x_i - x_j)^T W W^T (x_i - x_j)}{2\sigma_0^2}} \tag{66}$$

$$= \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle \tag{67}$$

Notice that as $\sigma_0 \to 0$, we have

$$\lim_{\sigma_0 \to 0} \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle = \begin{cases} 0 & \forall i \neq j \\ 1 & \forall i = j \end{cases}. \tag{68}$$

In other words, as $\sigma_0 \to 0$, the samples r_i in the RKHS of a Gaussian kernel approaches orthogonal to all other samples. Given this fact, it also implies that the σ_0 controls the inner product magnitude in RKHS space of the maximum sample pair r_i^*, r_j^*. We define this maximum inner product as

$$\langle \psi_{\sigma_0}(x_i^*), \psi_{\sigma_0}(x_j^*) \rangle \geq \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle \tag{69}$$

or equivalently

$$\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle \tag{70}$$

Therefore, given a σ_0, it controls the upper bound of the inner product. Notice that as $\sigma_0 \to 0$, every sample in RKHS becomes orthogonal. Therefore, the upper bound of $\langle r_i, r_j \rangle$ also approaches 0 when $r_i \neq r_j$. From this, we see the relationship

$$\lim_{\sigma_0 \to 0} u_{\sigma_0} = \lim_{\sigma_0 \to 0} \exp -(|.|/\sigma_0^2) = 0 \tag{71}$$

, where |.| is bounded and has a minimum and maximum, because we have finite number of samples.

□

Lemma 3. *Given any fixed $\sigma_1 > 0$, the lower bound $\mathscr{L}(\sigma_0, \sigma_1)$ is a function with respect to σ_0 and as $\sigma_0 \to 0$, $\mathscr{L}(\sigma_0, \sigma_1)$ approaches the function*

$$\mathscr{L}(\sigma_1) = \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{1}{\zeta \sigma_1^2}}. \tag{72}$$

At this point, if we let $\sigma_1 \to 0$, we have

$$\lim_{\sigma_1 \to 0} \mathscr{L}(\sigma_1) = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \tag{73}$$

$$= \mathcal{H}^*. \tag{74}$$

Proof.

Given Lemma 2, we know that

$$\lim_{\sigma_0 \to 0} u_{\sigma_0} = 0. \tag{75}$$

Therefore, having $\sigma_0 \to 0$ is equivalent to having $u_{\sigma_0} \to 0$. Since Lemma 1 provide the equation of a lower bound that is a function of u_{σ_0}, this lemma is proven by simply evaluating $\mathscr{L}(\sigma_0, \sigma_1)$ as $u_{\sigma_0} \to 0$. Following these steps, we have

$$\mathscr{L}(\sigma_1) = \lim_{u_{\sigma_0} \to 0} \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}, \tag{76}$$

$$= \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{1}{\zeta \sigma_1^2}}. \tag{77}$$

At this point, as $\sigma_1 \to 0$, our lower bound reaches the global maximum

$$\lim_{\sigma_1 \to 0} \mathscr{L}(\sigma_1) = \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \tag{78}$$

$$= \mathcal{H}^*. \tag{79}$$

□

Lemma 4. *Given any $\mathcal{H}_{l-2}$, $\delta > 0$, there exists a $\sigma_0 > 0$ and $\sigma_1 > 0$ such that*

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta. \tag{80}$$

Proof.

Observation 1.

Note that the objective of $\mathcal{H}_l$ is

$$\begin{aligned} \mathcal{H}_l = \max_{W} \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i^{(\mathcal{S})} - r_j^{(\mathcal{S})})^T W W^T (r_i^{(\mathcal{S})} - r_j^{(\mathcal{S})})}{2\sigma_1^2}} \\ - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i^{(\mathcal{S}^c)} - r_j^{(\mathcal{S}^c)})^T W W^T (r_i^{(\mathcal{S}^c)} - r_j^{(\mathcal{S}^c)})}{2\sigma_1^2}}. \end{aligned} \tag{81}$$

Since the Gaussian kernel is bounded between 0 and 1, the theoretical maximum of $\mathcal{H}^*$ is when the kernel is 1 for $\mathcal{S}$ and 0 for $\mathcal{S}^c$ with the theoretical maximum as $\mathcal{H}^* = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j}$. Therefore Eq. (80) inequality is equivalent to

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} - \mathcal{H}_l \leq \delta. \tag{82}$$

Observation 2.

If we choose a σ_0 such that

$$\mathscr{L}^*(\sigma_1) - \mathscr{L}(\sigma_0, \sigma_1) \leq \frac{\delta}{2} \quad \text{and} \quad \mathcal{H}^* - \mathscr{L}^*(\sigma_1) \leq \frac{\delta}{2} \tag{83}$$

then we have identified the condition where $\sigma_0 > 0$ and $\sigma_1 > 0$ such that

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} - \mathscr{L}(\sigma_0, \sigma_1) \leq \delta. \tag{84}$$

Note that the $\mathscr{L}^*(\sigma_1)$ is a continuous function of σ_1. Therefore, a σ_1 exists such that $\mathscr{L}^*(\sigma_1)$ can be set arbitraty close to $\mathcal{H}^*$. Hence, we choose an σ_1 that has the following property:

$$\mathcal{H}^* - \mathscr{L}^*(\sigma_1) \leq \frac{\delta}{2}. \tag{85}$$

We next fix σ_1, we also know $\mathscr{L}(\sigma_0, \sigma_1)$ is a continuous function of σ_0, and it has a limit $\mathscr{L}^*(\sigma_1)$ as σ_0 approaches to 0, hence there exits a σ_0, where

$$\mathscr{L}^*(\sigma_1) - \mathscr{L}(\sigma_0, \sigma_1) \leq \frac{\delta}{2} \tag{86}$$

Then we have:

$$\mathscr{L}^*(\sigma_1) - \mathscr{L}(\sigma_0, \sigma_1) \leq \frac{\delta}{2} \quad \text{and} \quad \mathcal{H}^* - \mathscr{L}^*(\sigma_1) \leq \frac{\delta}{2}. \tag{87}$$

By adding the two $\frac{\delta}{2}$, we conclude the proof.

□

Lemma 5. *There exists a Kernel Sequence $\{\phi_{l^\circ}\}_{l=1}^{L}$ parameterized by a set of weights W_l and a set of bandwidths σ_l such that*

$$\lim_{l\to\infty} \mathcal{H}_l = \mathcal{H}^*, \quad \mathcal{H}_{l+1} > \mathcal{H}_l \quad \forall l \tag{88}$$

Before, the proof, we use the following figure, Fig. 5, to illustrate the relationship between *Kernel Sequence* $\{\phi_{l^\circ}\}_{l=1}^{L}$ that generates the $\mathcal{H}$*-Sequence* $\{\mathcal{H}_l\}_{l=1}^{L}$. By solving a network greedily, we separate the network into L separable problems. At each additional layer, we rely on the weights learned from the previous layer. At each network, we find σ_{l-1}, σ_l, and W_l for the next network. We also note that since we only need to prove the existence of a solution, this proof is done by ***Proof by Construction***, i.e, we only need to show an example of its existence. Therefore, this proof consists of us constructing a $\mathcal{H}$*-Sequence* which satisfies the lemma.

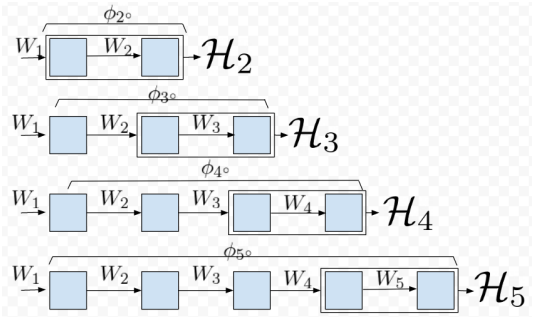

Figure 5: Relating *Kernel Sequence* to $\mathcal{H}$*-Sequence*.

Proof.

We first note that from Lemma 4, we have previously proven given any $\mathcal{H}_{l-2}$, $\delta > 0$, there exists a $\sigma_0 > 0$ and $\sigma_1 > 0$ such that

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta_l. \tag{89}$$

This implies that based on Fig. 5, at any given layer, we could reach arbitrarily close to $\mathcal{H}^*$. Given this, we list the 2 steps to build the $\mathcal{H}$*-Sequence*.

Step 1: Define $\{\mathcal{E}_n\}_{n=1}^{\infty}$ as a sequence of numbers $\mathcal{H}^* - \frac{\mathcal{H}^* - \mathcal{H}_0}{n}$ on the real line. We have the following properties for this sequence:

$$\lim_{n\to\infty} \mathcal{E}_n = \mathcal{H}^*, \quad \mathcal{E}_1 = \mathcal{H}_0. \tag{90}$$

Using these two properties, for any $\mathcal{H}_{l-1} \in [\mathcal{H}_0, \mathcal{H}^*]$ there exist an unique n, where

$$\mathcal{E}_n \leq \mathcal{H}_{l-1} < \mathcal{E}_{n+1}. \tag{91}$$

Step 2: For any given l, we choose δ_l to satisfies Eq. (89) by the following procedure, First find an n that satisfies

$$\mathcal{E}_n \leq \mathcal{H}_{l-1} < \mathcal{E}_{n+1}, \tag{92}$$

and second define δ_l to be

$$\delta_l = \mathcal{H}^* - \mathcal{E}_{n+1}. \tag{93}$$

To satisfy Eq. (89), the following must be true.

$$\mathcal{H}^* - \mathcal{H}_{l-1} \leq \delta_{l-1}. \tag{94}$$

and further we found n such that

$$\mathcal{E}_n \leq \mathcal{H}_{l-1} < \mathcal{E}_{n+1} \implies \mathcal{H}^* - \mathcal{E}_n \geq \mathcal{H}^* - \mathcal{H}_{l-1} > \mathcal{H}^* - \mathcal{E}_{n+1}. \tag{95}$$

Thus combining Eq. (93), Eq. (94), and Eq. (95) we have

$$\delta_{l-1} > \delta_l. \tag{96}$$

Therefore, $\{\delta_l\}$ is a decreasing sequence.

Step 3: Note that $\{\mathcal{E}_n\}$ is a converging sequence where

$$\lim_{n\to\infty} \mathcal{H}^* - \frac{\mathcal{H}^* - \mathcal{H}_0}{n} = \mathcal{H}^*. \tag{97}$$

Therefore, $\{\Delta_n\} = \mathcal{H}^* - \{\mathcal{E}_n\}$ is also a converging sequence where

$$\lim_{n\to\infty} \mathcal{H}^* - \mathcal{H}^* + \frac{\mathcal{H}^* - \mathcal{H}_0}{n} = 0 \tag{98}$$

and $\{\delta_l\}$ is a subsequence of $\{\Delta_l\}$. Since any subsequence of a converging sequence also converges to the same limit, we know that

$$\lim_{l\to\infty} \delta_l = 0. \tag{99}$$

Following this construction, if we always choose $\mathcal{H}_l$ such that

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta_l. \tag{100}$$

As $l \to \infty$, the inequality becomes

$$\mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l \leq \lim_{l\to\infty} \delta_l, \tag{101}$$

$$\leq 0. \tag{102}$$

Since we know that

$$\mathcal{H}^* - \mathcal{H}_l \geq 0 \,\forall l. \tag{103}$$

The condition of

$$0 \leq \mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l \leq 0 \tag{104}$$

is true only if

$$\mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l = 0. \tag{105}$$

This allows us to conclude

$$\mathcal{H}^* = \lim_{l\to\infty} \mathcal{H}_l. \tag{106}$$

Proof of the Monotonic Improvement.

Given Eq. (91) and Eq. (93), at each step we have the following:

$$\mathcal{H}_{l-1} < \mathcal{E}_{n+1} \tag{107}$$

$$\leq \mathcal{H}^* - \delta_l. \tag{108}$$

Rearranging this inequality, we have

$$\delta_l < \mathcal{H}^* - \mathcal{H}_{l-1}. \tag{109}$$

By combining the inequalities from Eq. (109) and Eq. (100), we have the following relationships.

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta_l < \mathcal{H}^* - \mathcal{H}_{l-1} \tag{110}$$

$$\mathcal{H}^* - \mathcal{H}_l < \mathcal{H}^* - \mathcal{H}_{l-1} \tag{111}$$

$$-\mathcal{H}_l < -\mathcal{H}_{l-1} \tag{112}$$

$$\mathcal{H}_l > \mathcal{H}_{l-1}, \tag{113}$$

$$\tag{114}$$

which concludes the proof of theorem.

□

Lemma 6. *Given $\frac{1}{\sqrt{\zeta}}$ as a normalizing constant for $W_s = \frac{1}{\sqrt{\zeta}} \sum_\alpha r_\alpha$ such that $W^T W = I$, then W_s is not guaranteed to be the optimal solution for the HSIC objective.*

Proof. We start with the Lagrangian

$$\mathcal{L} = -\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \mathrm{Tr}(\Lambda(W^T W - I)). \tag{115}$$

If we now take the derivative with respect to the Lagrange, we get

$$\nabla \mathcal{L} = \frac{1}{\sigma^2} \sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i - r_j)(r_i - r_j)^T W - 2W\Lambda. \tag{116}$$

By setting the gradient to 0, we have

$$\left[\frac{1}{2\sigma^2} \sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i - r_j)(r_i - r_j)^T \right] W = W\Lambda. \tag{117}$$

$$\mathcal{Q}_l W = W\Lambda. \tag{118}$$

From Eq. (118), we see that W is only the optimal solution when W is the eigenvector of Q_l. Therefore, by setting W to $W_s = \frac{1}{\sqrt{\zeta}} \sum_\alpha r_\alpha$, it is not guaranteed to yield an optimal.

□

Appendix B Proof for Theorem 2

Theorem 2: *As $l \to \infty$ and $\mathcal{H}_l \to \mathcal{H}^*$, the following properties are satisfied:*

I the scatter ratio approaches 0 where

$$\lim_{l \to \infty} \frac{\mathrm{Tr}(S_w^l)}{\mathrm{Tr}(S_b^l)} = 0 \tag{119}$$

II the *Kernel Sequence* converges to the following kernel:

$$\lim_{l \to \infty} \mathcal{K}(x_i, x_j)^l = \mathcal{K}^* = \begin{cases} 0 & \forall i, j \in \mathcal{S}^c \\ 1 & \forall i, j \in \mathcal{S} \end{cases} . \tag{120}$$

Proof. We start by proving condition II starting from the $\mathcal{H}$ objective using a GK

$$\max_W \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \mathcal{K}_W(r_i, r_j) - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \mathcal{K}_W(r_i, r_j) \tag{121}$$

$$\max_W \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} \tag{122}$$

Given that $\mathcal{H}_l \to \mathcal{H}^*$, and the fact that $0 \leq \mathcal{K}_W \leq 1$, this implies that the following condition must be true:

$$\mathcal{H}^* = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j}(1) - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}|(0). \tag{123}$$

Based on Eq. (89), our construction at each layer ensures to satisfy

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta_l. \tag{124}$$

Substituting the definition of $\mathcal{H}^*$ and $\mathcal{H}_l$, we have

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j}(1) - \left[\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \mathcal{K}_W(r_i, r_j) - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \mathcal{K}_W(r_i, r_j) \right] \leq \delta_l \tag{125}$$

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j}(1 - \mathcal{K}_W(r_i, r_j)) + \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \mathcal{K}_W(r_i, r_j) \leq \delta_l. \tag{126}$$

Since every term within the summation in Eq. (126) is positive, this implies

$$1 - \mathcal{K}_W(r_i, r_j) \leq \delta_l \quad i, j \in \mathcal{S} \tag{127}$$
$$\mathcal{K}_W(r_i, r_j) \leq \delta_l \quad i, j \in \mathcal{S}^c. \tag{128}$$

So as $l \to \infty$ and $\delta_l \to 0$, every component getting closer to limit Kernel, i.e, taking the limit from both sides and using the fact that is proven is theorem 1 $\lim_{l\to\infty} \delta_l = 0$ leads to

$$\lim_{l\to\infty} 1 \leq \mathcal{K}_W(r_i, r_j) \quad i, j \in \mathcal{S} \tag{129}$$
$$\lim_{l\to\infty} \mathcal{K}_W(r_i, r_j) \leq 0 \quad i, j \in \mathcal{S}^c \tag{130}$$

both terms must instead be strictly equality. Therefore, we see that at the limit point $\mathcal{K}_W$ would have the form

$$\mathcal{K}^* = \begin{cases} 0 & \forall i, j \in \mathcal{S}^c \\ 1 & \forall i, j \in \mathcal{S} \end{cases} . \tag{131}$$

First Property:

Using Eq. (127) and Eq. (128) we have:

$$1 - \delta_l \leq e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} \quad i, j \in \mathcal{S} \tag{132}$$
$$e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} \leq \delta_l \quad i, j \in \mathcal{S}^c. \tag{133}$$

As $\lim_{l\to\infty} \delta_l = 0$, taking the limit from both side leads to:

$$\begin{cases} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} = 1 & \forall i, j \in \mathcal{S} \\ e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} = 0 & \forall i, j \in \mathcal{S}^c \end{cases} . \tag{134}$$

If we take the log of the conditions, we get

$$\begin{cases} \frac{1}{2\sigma^2}(r_i - r_j)^T W W^T (r_i - r_j) = 0 & \forall i, j \in \mathcal{S} \\ \frac{1}{2\sigma^2}(r_i - r_j)^T W W^T (r_i - r_j) = \infty & \forall i, j \in \mathcal{S}^c \end{cases} . \tag{135}$$

This implies that as $l \to \infty$ we have

$$\lim_{l\to\infty} \sum_{i,j\in\mathcal{S}} \frac{1}{2\sigma^2}(r_i - r_j)^T W W^T (r_i - r_j) = \lim_{l\to\infty} \operatorname{Tr}(S_w) = 0. \tag{136}$$

$$\lim_{l\to\infty} \sum_{i,j\in\mathcal{S}^c} \frac{1}{2\sigma^2}(r_i - r_j)^T W W^T (r_i - r_j) = \lim_{l\to\infty} \operatorname{Tr}(S_b) = \infty, \tag{137}$$

This yields the ratio

$$\lim_{\mathcal{H}_l \to \mathcal{H}^*} \frac{\operatorname{Tr}(S_w)}{\operatorname{Tr}(S_b)} = \frac{0}{\infty} = 0. \tag{138}$$

□

Appendix C Proof for Theorem 3

Theorem 3: *Eq. (5) objective is equivalent to*

$$\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i^T W W^T r_j) - \sum_i D_i(W) ||W^T r_i||_2. \tag{139}$$

Proof. Let $A_{i,j} = (r_i - r_j)(r_i - r_j)^T$. Given the Lagranian of the HSIC objective as

$$\mathcal{L} = -\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \text{Tr}[\Lambda(W^T W - I)]. \tag{140}$$

Our layer wise HSIC objective becomes

$$\min_W -\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \text{Tr}[\Lambda(W^T W - I)]. \tag{141}$$

We take the derivative of the Lagrangian, the expression becomes

$$\nabla_W \mathcal{L}(W, \Lambda) = \sum_{i,j} \frac{\Gamma_{i,j}}{\sigma^2} e^{-\frac{\text{Tr}(W^T A_{i,j} W)}{2\sigma^2}} A_{i,j} W - 2W\Lambda. \tag{142}$$

Setting the gradient to 0, and consolidate some scalar values into $\hat{\Gamma}_{i,j}$, we get the expression

$$\left[\sum_{i,j} \frac{\Gamma_{i,j}}{2\sigma^2} e^{-\frac{\text{Tr}(W^T A_{i,j} W)}{2\sigma^2}} A_{i,j}\right] W = W\Lambda \tag{143}$$

$$\left[\frac{1}{2}\sum_{i,j} \hat{\Gamma}_{i,j} A_{i,j}\right] W = W\Lambda \tag{144}$$

$$\mathcal{Q} W = W\Lambda. \tag{145}$$

From here, we see that the optimal solution is an eigenvector of $\mathcal{Q}$. Based on ISM, it further proved that the optimal solution is not just any eigenvector, but the eigenvectors associated with the smallest values of $\mathcal{Q}$. From this logic, ISM solves objective (141) with a surrogate objective

$$\min_W \quad \text{Tr}\left(W^T \left[\frac{1}{2}\sum_{i,j} \hat{\Gamma}_{i,j} A_{i,j}\right] W\right) \quad \text{s.t.}\, W^T W = I. \tag{146}$$

Given $D_{\hat{\Gamma}}$ as the degree matrix of $\hat{\Gamma}$ and $R = [r_1, r_2, ...]^T$, ISM further shows that Eq. (146) can be written into

$$\min_W \quad \text{Tr}\left(W^T R^T \left[D_{\hat{\Gamma}} - \hat{\Gamma}\right] R W\right) \quad \text{s.t.}\, W^T W = I \tag{147}$$

$$\max_W \quad \text{Tr}\left(W^T R^T \left[\hat{\Gamma} - D_{\hat{\Gamma}}\right] R W\right) \quad \text{s.t.}\, W^T W = I \tag{148}$$

$$\max_W \quad \text{Tr}\left(W^T R^T \hat{\Gamma} R W\right) - \text{Tr}\left(W^T R^T D_{\hat{\Gamma}} R W\right) \quad \text{s.t.}\, W^T W = I \tag{149}$$

$$\max_W \quad \text{Tr}\left(\hat{\Gamma} R W W^T R^T\right) - \text{Tr}\left(D_{\hat{\Gamma}} R W W^T R^T\right) \quad \text{s.t.}\, W^T W = I \tag{150}$$

$$\max_W \quad \sum_{i,j} \hat{\Gamma}_{i,j} [R W W^T R^T]_{i,j} - \sum_{i,j} D_{\hat{\Gamma}_{i,j}} [R W W^T R^T]_{i,j} \quad \text{s.t.}\, W^T W = I. \tag{151}$$

Since the jump from Eq. (146) can be intimidating for those not familiar with the literature, we included a more detailed derivation in App. D.

Note that the degree matrix $D_{\hat{\Gamma}}$ only have non-zero diagonal elements, all of its off diagonal are 0. Given $[R W W^T R^T]_{i,j} = (r_i^T W W^T r_j)$, the objective becomes

$$\max_W \quad \sum_{i,j} \hat{\Gamma}_{i,j} (r_i^T W W^T r_j) - \sum_i D_i(W) ||W^T r_i||_2 \quad \text{s.t.}\, W^T W = I. \tag{152}$$

Here, we treat D_i as a penalty weight on the norm of the $W^T r_i$ for every sample. □

To better understand the behavior of $D_i(W)$, note that $\hat{\Gamma}$ matrix looks like

$$\hat{\Gamma} = \frac{1}{\sigma^2}\begin{bmatrix} \left[\Gamma_{\mathcal{S}} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}}\right] & \left[-|\Gamma_{\mathcal{S}^c}| e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}}\right] & \cdots \\ \left[-|\Gamma_{\mathcal{S}^c}| e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}}\right] & \left[\Gamma_{\mathcal{S}} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}}\right] & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix}. \tag{153}$$

The diagonal block matrix all $\Gamma_{i,j}$ elements that belong to $\mathcal{S}$ and the off diagonal are elements that belongs to $\mathcal{S}^c$. Each penalty term is the summation of its corresponding row. Hence, we can write out the penalty term as

$$D_i(W_l) = \frac{1}{\sigma^2} \sum_{j \in \mathcal{S}|i} \Gamma_{i,j} \mathcal{K}_{W_l}(r_i, r_j) - \frac{1}{\sigma^2} \sum_{j \in \mathcal{S}^c|i} |\Gamma_{i,j}| \mathcal{K}_{W_l}(r_i, r_j). \tag{154}$$

From this, it shows that as W improve the objective, the penalty term is also increased. In fact, at its extreme as $\mathcal{H}_l \to \mathcal{H}^*$, all the negative terms are gone and all of its positive terms are maximized and this matrix approaches

$$\hat{\Gamma}^* = \frac{1}{\sigma^2} \begin{bmatrix} [\Gamma_{\mathcal{S}}] & [0] & ... \\ [0] & [\Gamma_{\mathcal{S}}] & ... \\ ... & ... & ... \end{bmatrix}. \tag{155}$$

From the matrix $\hat{\Gamma}^*$ and the definition of $D_i(W_l)$, we see that as $\mathcal{K}_W$ from $\mathcal{S}$ increase,

Since $D_i(W)$ is the degree matrix of $\hat{\Gamma}$, we see that as $\mathcal{H}_l \to \mathcal{H}^*$, we have

$$D_i^*(W) > D_i(W). \tag{156}$$

Appendix D Derivation for $\sum_{i,j} \Psi_{i,j}(x_i - x_j)(x_i - x_j)^T = 2X^T(D_\Psi - \Psi)X$

Since Ψ is a symmetric matrix, and $A_{i,j} = (x_i - x_j)(x_i - x_j)^T$, we can rewrite the expression into

$$\begin{aligned} \textstyle\sum_{i,j} \Psi_{i,j} A_{i,j} &= \textstyle\sum_{i,j} \Psi_{i,j}(x_i - x_j)(x_i - x_j)^T \\ &= \textstyle\sum_{i,j} \Psi_{i,j}(x_i x_i^T - x_j x_i^T - x_i x_j^T + x_j x_j^T) \\ &= 2\textstyle\sum_{i,j} \Psi_{i,j}(x_i x_i^T - x_j x_i^T) \\ &= \left[2\textstyle\sum_{i,j} \Psi_{i,j}(x_i x_i^T)\right] - \left[2\textstyle\sum_{i,j} \Psi_{i,j}(x_i x_j^T)\right]. \end{aligned}$$

If we expand the 1st term, we get

$$2\sum_i^n \sum_j^n \Psi_{i,j}(x_i x_i^T) = 2\sum_i \Psi_{i,1}(x_i x_i^T) + \ldots + \Psi_{i,n}(x_i x_i^T) \tag{157}$$

$$= 2\sum_i^n [\Psi_{1,1} + \Psi_{1,2} + ...] x_i x_i^T \tag{158}$$

$$= 2\sum_i^n d_i x_i x_i^T \tag{159}$$

$$= 2X^T D_\Psi X \tag{160}$$

Given Ψ_i as the ith row, next we look at the 2nd term

$$2\sum_i \sum_j \Psi_{i,j} x_i x_j^T = 2\sum_i \Psi_{i,1} x_i x_1^T + \Psi_{i,2} x_i x_2^T + \Psi_{i,3} x_i x_3^T + ... \tag{161}$$

$$= 2\sum_i x_i(\Psi_{i,1} x_1^T) + x_i(\Psi_{i,2} x_2^T) + x_i(\Psi_{i,3} x_3^T) + ... \tag{162}$$

$$= 2\sum_i x_i \left[(\Psi_{i,1} x_1^T) + (\Psi_{i,2} x_2^T) + (\Psi_{i,3} x_3^T) + ...\right] \tag{163}$$

$$= 2\sum_i x_i \left[X^T \Psi_i^T\right]^T \tag{164}$$

$$= 2\sum_i x_i \left[\Psi_i X\right] \tag{165}$$

$$= 2\left[x_1 \Psi_1 X + x_2 \Psi_2 X + x_3 \Psi_3 X + ...\right] \tag{166}$$

$$= 2\left[x_1 \Psi_1 + x_2 \Psi_2 + x_3 \Psi_3 + ...\right] X \tag{167}$$

$$= 2X^T \Psi X \tag{168}$$

$$\tag{169}$$

Putting both terms together, we get

$$\sum_{i,j} \Psi_{i,j} A_{i,j} = 2X^T D_\Psi X - 2X^T \Psi X a \tag{170}$$

$$= 2X^T [D_\Psi - \Psi] X \tag{171}$$

$$\tag{172}$$

Appendix E Proof for Corollary 1 and 2

Corollary 1: *Given $\mathcal{H}_l \to \mathcal{H}^*$, the network output in IDS solves MSE via a translation of labels.*

Proof.

As $\mathcal{H}_l \to \mathcal{H}^*$, Thm. 2 shows that sample of the same class are mapped into the same point. Assuming that ϕ has mapped the sample into c points $\alpha = [\alpha_1, ..., \alpha_c]$ that's different from the truth label $\xi = [\xi_1, ..., \xi_c]$. Then the MSE objective is minimized by translating the ϕ output by

$$\xi - \alpha. \tag{173}$$

□

Corollary 2: *Given $\mathcal{H}_l \to \mathcal{H}^*$, the network output in RKHS solves CE via a change of bases.*

Assumptions, and Notations.

1. n is the number of samples.
2. τ is the number of classes.
3. $y_i \in \mathbb{R}^\tau$ is the ground truth label for the i^{th} sample. It is one-hot encoded where only the j^{th} element is 1 if x_i belongs to the j^{th} class, all other elements would be 0.
4. We denote ϕ as the network, and $\hat{y}_i \in \mathbb{R}^\tau$ as the network output where $\hat{y}_i = \phi(x_i)$. We also assume that $\hat{y}_i$ is constrained on a probability simplex where $1 = \hat{y}_i^T \mathbf{1}_n$.
5. We denote the j^{th} element of y_i, and $\hat{y}_i$ as $y_{i,j}$ and $\hat{y}_{i,j}$ respectively.
6. We define

   Orthogonality Condition: A set of samples $\{\hat{y}_1, ..., \hat{y}_n\}$ satisfies the orthogonality condition if

   $$\begin{cases} \langle \hat{y_i}, \hat{y_j} \rangle = 1 & \forall \quad i,j \text{ same class} \\ \langle \hat{y_i}, \hat{y_j} \rangle = 0 & \forall \quad i,j \text{ not in the same class} \end{cases}. \tag{174}$$

7. We define the Cross-Entropy objective as

   $$\arg\min_\phi - \sum_{i=1}^{n} \sum_{j=1}^{\tau} y_{i,j} \log(\phi(x_i)_{i,j}). \tag{175}$$

Proof.

From Thm. 2, we know that the network ϕ output, $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$, satisfy the orthogonality condition at $\mathcal{H}^*$. Then there exists a set of orthogonal bases represented by $\Xi = [\xi_1, \xi_2, ..., \xi_c]$ that maps $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_n\}$ to simulate the output of a softmax layer. Let $\xi_i = \hat{y}_j, j \in \mathcal{S}^i$, i.e., for the i_{th} class we arbitrary choose one of the samples from this class and assigns ξ_i of that class to be equal to the sample's output. Realize in our problem we have $< \hat{y}_i, \hat{y}_i >= 1$, so if $< \hat{y}_i, \hat{y}_j >= 1$, then subtracting these two would lead to $< \hat{y}_i, \hat{y}_i - \hat{y}_j >= 0$, which is the same as $\hat{y}_i = \hat{y}_j$. So this representation is well-defined and its independent of choices of the sample from each group if they satisfy orthogonality condition. Now we define transformed labels, Y as:

$$Y = \hat{Y}\Xi. \tag{176}$$

Note that $Y = [y_1, y_2, ..., y_n]^T$ which each y_i is a one hot vector representing the class membership of i sample in c classes. Since given Ξ as the change of basis, we can match $\hat{Y}$ to Y exactly, CE is minimized.

□

Appendix F Dataset Details

No samples were excludes from any of the dataset.

Wine. This dataset has 13 features, 178 samples, and 3 classes. The features are continuous and heavily unbalanced in magnitude. The dataset can be downloaded at `https://archive.ics.uci.edu/ml/datasets/wine`.

Divorce. This dataset has 54 features, 170 samples, and 2 classes. The features are discrete and balanced in magnitude. The dataset can be downloaded at `https://archive.ics.uci.edu/ml/datasets/Divorce+Predictors+data+set`.

Car. This dataset has 6 features, 1728 samples and 2 classes. The features are discrete and balanced in magnitude. The dataset can be downloaded at `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`.

Cancer. This dataset has 9 features, 683 samples, and 2 classes. The features are discrete and unbalanced in magnitude. The dataset can be downloaded at `https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)`.

Face. This dataset consists of images of 20 people in various poses. The 624 images are vectorized into 960 features. The dataset can be downloaded at `https://archive.ics.uci.edu/ml/datasets/CMU+Face+Images`.

Random. This dataset has 2 features, 80 samples and 2 classes. It is generate with a gaussian distribution where half of the samples are randomly labeled as 1 or 0.

Adversarial. This dataset has 2 features, 80 samples and 2 classes. It is generate with the following code:

```
#!/usr/bin/env python

n = 40
X1 = np.random.rand(n,2)
X2 = X1 + 0.01*np.random.randn(n,2)

X = np.vstack((X1,X2))
Y = np.vstack(( np.zeros((n,1)), np.ones((n,1)) ))
```

Appendix G W_l Dimensions for each 10 Fold of each Dataset

We report the input and output dimensions of each W_l for every layer of each dataset in the form of (α, β); the corresponding dimension becomes $W_l \in \mathbb{R}^{\alpha \times \beta}$. Since each dataset consists of 10-folds, the network structure for each fold is reported. We note that the input of the 1st layer is the dimension of the original data. However, after the first layer, the width of the RFF becomes the output of each layer; here we use 300.

The β value is chosen during the ISM algorithm. By keeping only the most dominant eigenvector of the Φ matrix, the output dimension of each layer corresponds with the rank of Φ. It can be seen from each dataset that the first layer significantly expands the rank. The expansion is generally followed by a compression of fewer and fewer eigenvalues. These results conform with the observations made by Montavon et al. [43] and Ansuini et al. [42].

Data	Layer 1	Layer 2	Layer 3	Layer 4
adversarial 1	(2, 2)	(300, 61)	(300, 35)	
adversarial 2	(2, 2)	(300, 61)	(300, 35)	
adversarial 3	(2, 2)	(300, 61)	(300, 8)	(300, 4)
adversarial 4	(2, 2)	(300, 61)	(300, 29)	
adversarial 5	(2, 2)	(300, 61)	(300, 29)	
adversarial 6	(2, 2)	(300, 61)	(300, 7)	(300, 4)
adversarial 7	(2, 2)	(300, 61)	(300, 34)	
adversarial 8	(2, 2)	(300, 12)	(300, 61)	(300, 30)
adversarial 9	(2, 2)	(300, 61)	(300, 33)	
adversarial 10	(2, 2)	(300, 61)	(300, 33)	

Data	Layer 1	Layer 2	Layer 3
Random 1	(3, 3)	(300, 47)	(300, 25)
Random 2	(3, 3)	(300, 46)	(300, 25)
Random 3	(3, 3)	(300, 46)	(300, 25)
Random 4	(3, 3)	(300, 47)	(300, 4)
Random 5	(3, 3)	(300, 47)	(300, 25)
Random 6	(3, 3)	(300, 45)	(300, 23)
Random 7	(3, 3)	(300, 45)	(300, 25)
Random 8	(3, 3)	(300, 45)	(300, 21)
Random 9	(3, 3)	(300, 45)	(300, 26)
Random 10	(3, 3)	(300, 47)	(300, 25)

Data	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
spiral 1	(2, 2)	(300, 15)	(300, 6)	(300, 7)	(300, 6)	
spiral 2	(2, 2)	(300, 13)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
spiral 3	(2, 2)	(300, 12)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
spiral 4	(2, 2)	(300, 13)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
spiral 5	(2, 2)	(300, 13)	(300, 6)	(300, 7)	(300, 6)	
spiral 6	(2, 2)	(300, 14)	(300, 6)	(300, 7)	(300, 6)	
spiral 7	(2, 2)	(300, 14)	(300, 6)	(300, 7)	(300, 6)	
spiral 8	(2, 2)	(300, 14)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
spiral 9	(2, 2)	(300, 13)	(300, 6)	(300, 7)	(300, 6)	
spiral 10	(2, 2)	(300, 14)	(300, 6)	(300, 7)	(300, 6)	

Data	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
wine 1	(13, 11)	(300, 76)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
wine 2	(13, 11)	(300, 76)	(300, 6)	(300, 6)	(300, 6)	(300, 6)
wine 3	(13, 11)	(300, 75)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
wine 4	(13, 11)	(300, 76)	(300, 6)	(300, 6)	(300, 6)	(300, 6)
wine 5	(13, 11)	(300, 74)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
wine 6	(13, 11)	(300, 74)	(300, 6)	(300, 6)	(300, 6)	(300, 6)
wine 7	(13, 11)	(300, 74)	(300, 6)	(300, 6)	(300, 6)	(300, 6)
wine 8	(13, 11)	(300, 75)	(300, 6)	(300, 7)	(300, 6)	(300, 6)
wine 9	(13, 11)	(300, 75)	(300, 6)	(300, 8)	(300, 6)	(300, 6)
wine 10	(13, 11)	(300, 76)	(300, 6)	(300, 7)	(300, 6)	(300, 6)

Data	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6
car 1	(6, 6)	(300, 96)	(300, 6)	(300, 8)	(300, 6)	
car 2	(6, 6)	(300, 96)	(300, 6)	(300, 8)	(300, 6)	
car 3	(6, 6)	(300, 91)	(300, 6)	(300, 8)	(300, 6)	
car 4	(6, 6)	(300, 88)	(300, 6)	(300, 8)	(300, 6)	(300, 6)
car 5	(6, 6)	(300, 94)	(300, 6)	(300, 8)	(300, 6)	
car 6	(6, 6)	(300, 93)	(300, 6)	(300, 7)		
car 7	(6, 6)	(300, 92)	(300, 6)	(300, 8)	(300, 6)	
car 8	(6, 6)	(300, 95)	(300, 6)	(300, 7)	(300, 6)	
car 9	(6, 6)	(300, 96)	(300, 6)	(300, 9)	(300, 6)	
car 10	(6, 6)	(300, 99)	(300, 6)	(300, 8)	(300, 6)	

Data	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
divorce 1	(54, 35)	(300, 44)	(300, 5)	(300, 5)	
divorce 2	(54, 35)	(300, 45)	(300, 4)	(300, 4)	
divorce 3	(54, 36)	(300, 49)	(300, 6)	(300, 6)	
divorce 4	(54, 36)	(300, 47)	(300, 7)	(300, 6)	
divorce 5	(54, 35)	(300, 45)	(300, 6)	(300, 6)	
divorce 6	(54, 36)	(300, 47)	(300, 6)	(300, 6)	
divorce 7	(54, 35)	(300, 45)	(300, 6)	(300, 6)	(300, 4)
divorce 8	(54, 36)	(300, 47)	(300, 6)	(300, 7)	(300, 4)
divorce 9	(54, 36)	(300, 47)	(300, 5)	(300, 5)	
divorce 10	(54, 36)	(300, 47)	(300, 6)	(300, 6)	

Data	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5	Layer 6	Layer 7	Layer 8	Layer 9	Layer 10
cancer 1	(9, 8)	(300, 90)	(300, 5)	(300, 6)	(300, 6)	(300, 5)	(300, 4)	(300, 5)	(300, 6)	(300, 6)
cancer 2	(9, 8)	(300, 90)	(300, 6)	(300, 7)	(300, 8)	(300, 11)	(300, 8)	(300, 4)		
cancer 3	(9, 8)	(300, 88)	(300, 5)	(300, 6)	(300, 7)	(300, 7)	(300, 6)	(300, 4)		
cancer 4	(9, 8)	(300, 93)	(300, 6)	(300, 7)	(300, 9)	(300, 11)	(300, 8)			
cancer 5	(9, 8)	(300, 93)	(300, 9)	(300, 10)	(300, 10)	(300, 11)	(300, 9)	(300, 7)		
cancer 6	(9, 8)	(300, 92)	(300, 7)	(300, 8)	(300, 8)	(300, 7)	(300, 7)			
cancer 7	(9, 8)	(300, 90)	(300, 4)	(300, 4)	(300, 5)	(300, 6)	(300, 6)	(300, 6)	(300, 6)	
cancer 8	(9, 8)	(300, 88)	(300, 5)	(300, 6)	(300, 7)	(300, 8)	(300, 7)	(300, 6)		
cancer 9	(9, 8)	(300, 88)	(300, 5)	(300, 7)	(300, 7)	(300, 7)	(300, 7)			
cancer 10	(9, 8)	(300, 97)	(300, 9)	(300, 11)	(300, 12)	(300, 13)	(300, 6)			

Data	Layer 1	Layer 2	Layer 3	Layer 4
face 1	(960, 233)	(300, 74)	(300, 73)	(300, 46)
face 2	(960, 231)	(300, 75)	(300, 73)	(300, 43)
face 3	(960, 231)	(300, 76)	(300, 73)	(300, 44)
face 4	(960, 232)	(300, 76)	(300, 74)	(300, 44)
face 5	(960, 231)	(300, 77)	(300, 73)	(300, 43)
face 6	(960, 232)	(300, 74)	(300, 72)	(300, 47)
face 7	(960, 232)	(300, 76)	(300, 73)	(300, 45)
face 8	(960, 230)	(300, 74)	(300, 74)	(300, 44)
face 9	(960, 233)	(300, 76)	(300, 76)	(300, 45)
face 10	(960, 231)	(300, 76)	(300, 70)	(300, 43)

Appendix H Sigma Values used for Random and Adversarial Simulation

The simulation of Thm. 1 as shown in Fig. 2 spread the improvement across multiple layers. The σ_l and $\mathcal{H}_l$ values are recorded here. We note that σ_l are reasonably large and not approaching 0 and the improvement of $\mathcal{H}_l$ is monotonic.

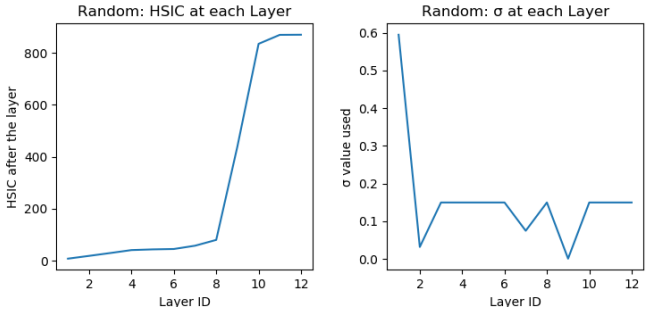

Figure 6

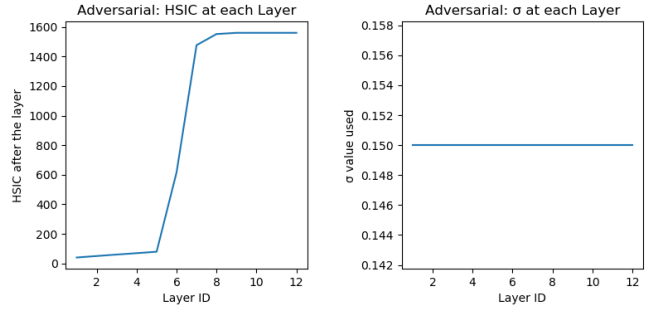

Figure 7

Given a sufficiently small σ_0 and σ_1, Thm. 1 claims that it can come arbitrarily close to the global optimal using a minimum of 2 layers. We here simulate 2 layers using a relatively small σ values ($\sigma_0 = 10^{-5}$) on the Random (left) and Adversarial (right) data and display the results of the 2 layers below. Notice that given 2 layer, it generated a clearly separable clusters that are pushed far apart.

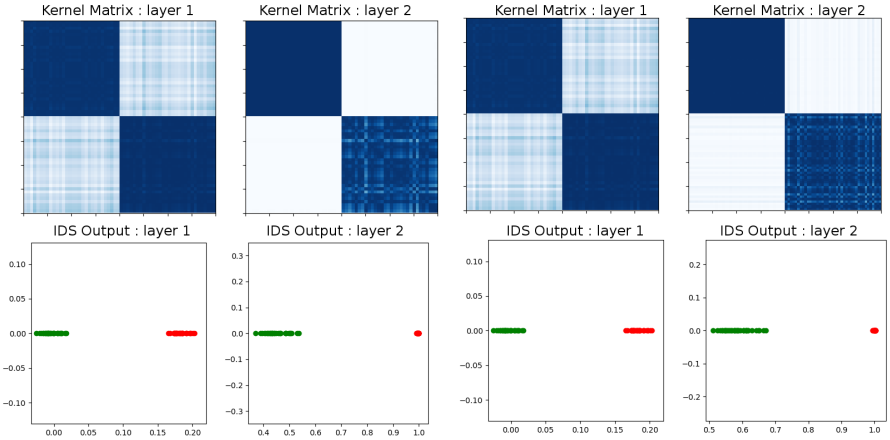

Figure 8: Random Dataset with 2 layers and $\sigma = 10^{-5}$

Figure 9: Adversarial Dataset with 2 layers and $\sigma = 10^{-5}$

Appendix I Graphs of Kernel Sequences

A representation of the *Kernel Sequence* are displayed in the figures below for each dataset. The samples of the kernel matrix are previously organized to form a block structure by placing samples of the same class adjacent to each other. Since the Gaussian kernel is restricted to values between 0 and 1, we let white and dark blue be 0 and 1 respectively where the gradients reflect values in between. Our theorems predict that the *Kernel Sequence* will evolve from an uninformative kernel into a highly discriminating kernel of perfect block structures.

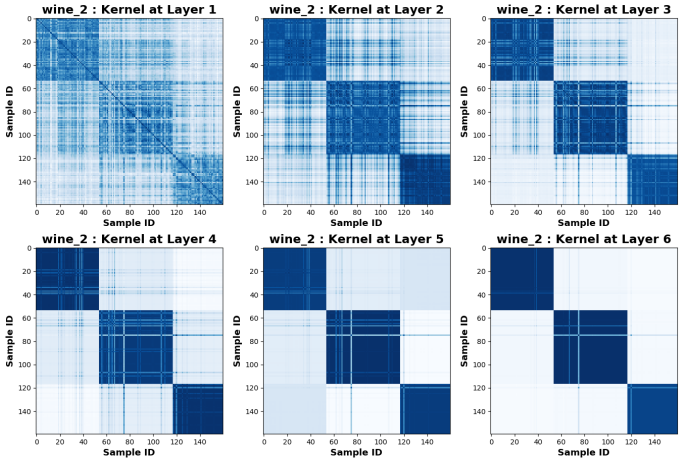

Figure 10: The kernel sequence for the wine dataset.

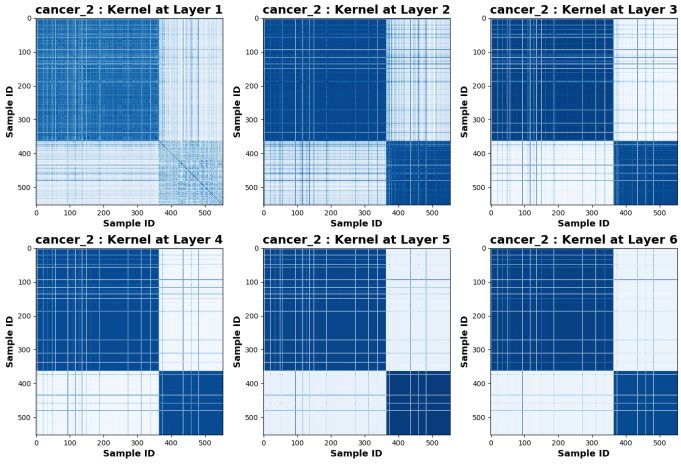

Figure 11: The kernel sequence for the cancer dataset.

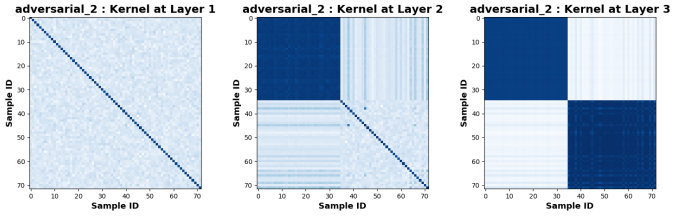

Figure 12: The kernel sequence for the Adversarial dataset.

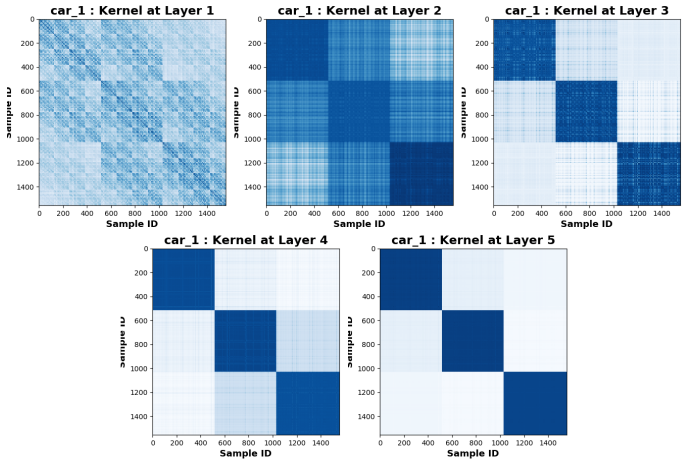

Figure 13: The kernel sequence for the car dataset.

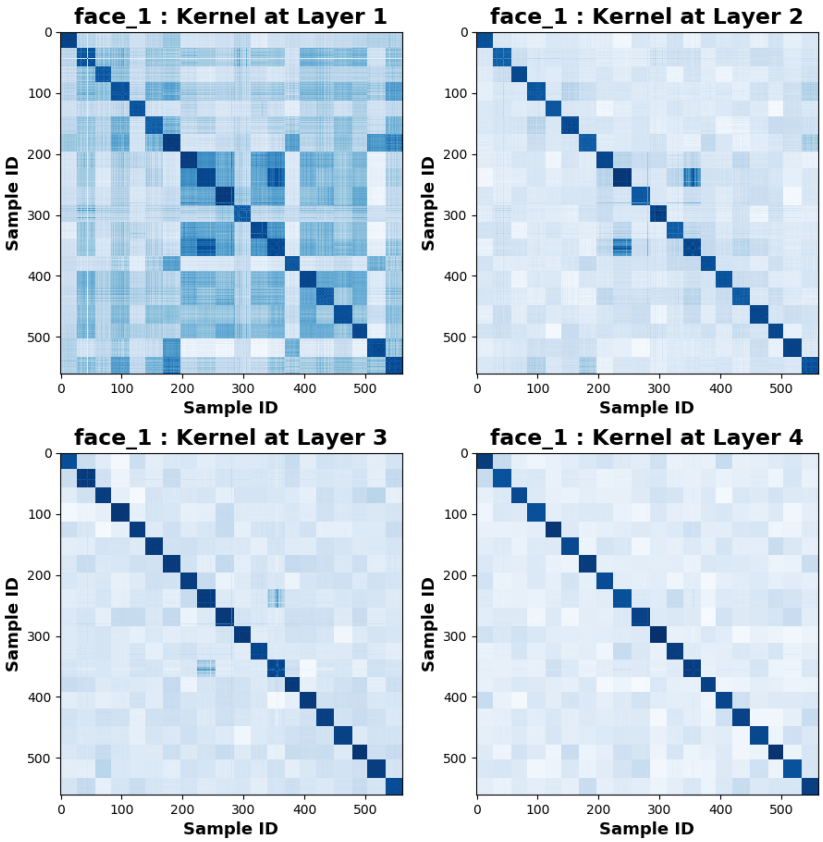

Figure 14: The kernel sequence for the face dataset.

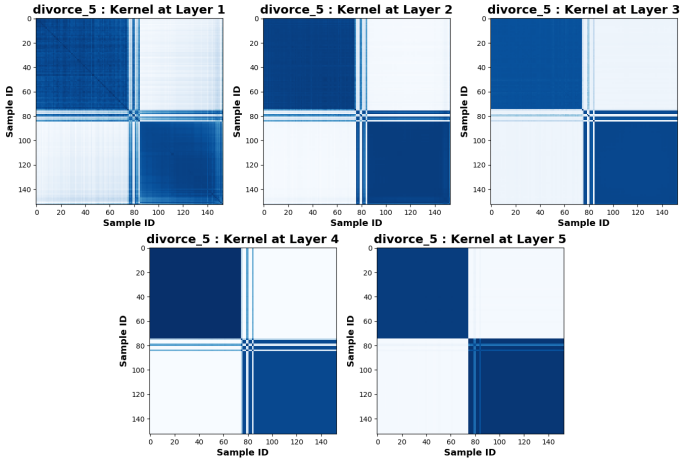

Figure 15: The kernel sequence for the divorce dataset.

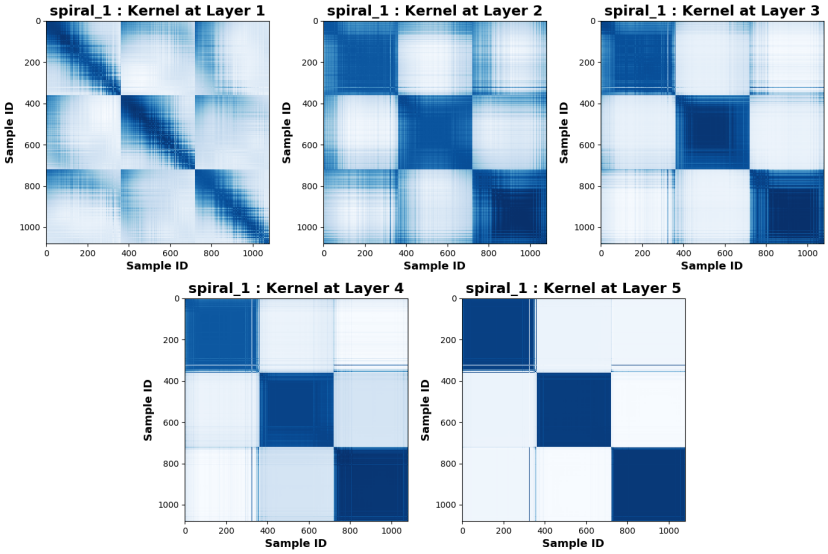

Figure 16: The kernel sequence for the spiral dataset.

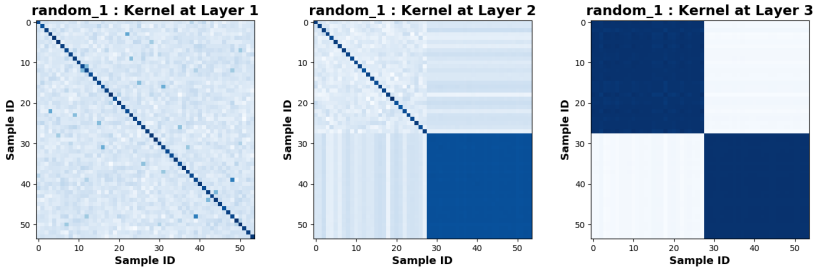

Figure 17: The kernel sequence for the Random dataset.

Appendix J Evaluation Metrics Graphs

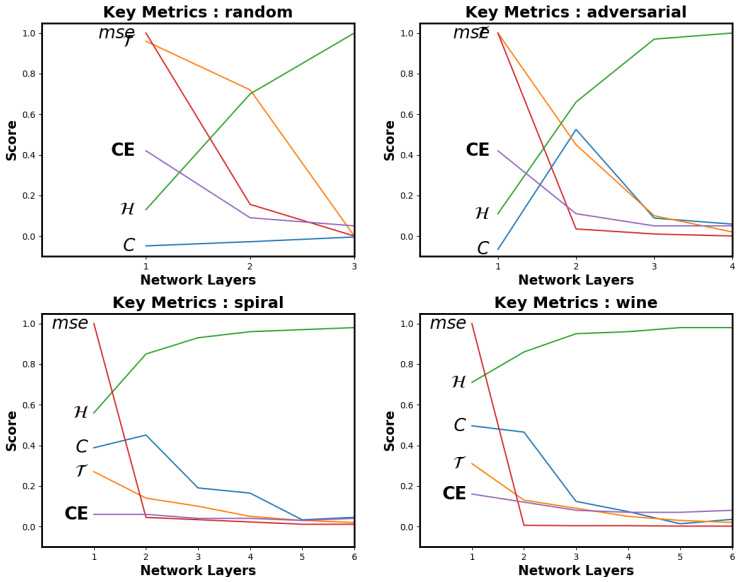

Figure 18

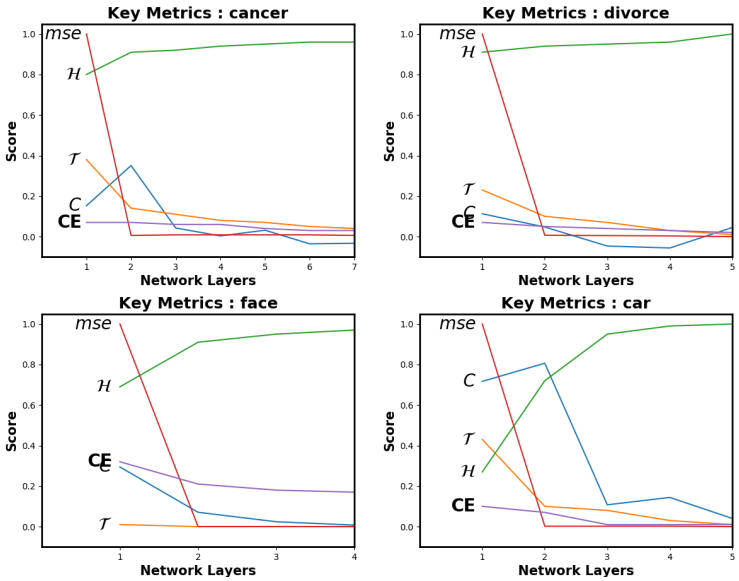

Figure 19: Figures of key metrics for all datasets as samples progress through the network. It is important to notice the uniformly and monotonically increasing $\mathcal{H}$-*Sequence* for each plot since this guarantees a converging kernel/risk sequence. As the $\mathcal{T}$ approach 0, samples of the same/difference classes in IDS are being pulled into a single point or pushed maximally apart respectively. As C approach 0, samples of the same/difference classes in RKHS are being pulled into 0 or $\frac{\pi}{2}$ cosine similarity respectively.

Appendix K Optimal Gaussian σ for Maximum Kernel Separation

Although the Gaussian kernel is the most common kernel choice for kernel methods, its σ value is a hyperparameter that must be tuned for each dataset. This work proposes to set the σ value based on the maximum kernel separation. The source code is made publicly available on `https://github.com/anonamous`.

Let $X \in \mathbb{R}^{n \times d}$ be a dataset of n samples with d features and let $Y \in \mathbb{R}^{n \times \tau}$ be the corresponding one-hot encoded labels where τ denotes the number of classes. Given $\kappa_X(\cdot, \cdot)$ and $\kappa_Y(\cdot, \cdot)$ as two kernel functions that applies respectively to X and Y to construct kernel matrices $K_X \in \mathbb{R}^{n \times n}$ and $K_Y \in \mathbb{R}^{n \times n}$. Given a set $\mathcal{S}$, we denote $|\mathcal{S}|$ as the number of elements within the set. Also let $\mathcal{S}$ and $\mathcal{S}^c$ be sets of all pairs of samples of (x_i, x_j) from a dataset X that belongs to the same and different classes respectively, then the average kernel value for all (x_i, x_j) pairs with the same class is

$$d_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{177}$$

and the average kernel value for all (x_i, x_j) pairs between different classes is

$$d_{\mathcal{S}^c} = \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{178}$$

We propose to find the σ that maximizes the difference between $d_{\mathcal{S}}$ and $d_{\mathcal{S}^c}$ or

$$\max_{\sigma} \quad \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} - \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{179}$$

It turns out that is expression can be computed efficiently. Let $g = \frac{1}{|\mathcal{S}|}$ and $\bar{g} = \frac{1}{|\mathcal{S}^c|}$, and let $\mathbf{1}_{n \times n} \in \mathbb{R}^{n \times n}$ be a matrix of 1s, then we can define Q as

$$Q = -gK_Y + \bar{g}(\mathbf{1}_{n \times n} - K_Y). \tag{180}$$

Or Q can be written more compactly as

$$Q = \bar{g}\mathbf{1}_{n \times n} - (g + \bar{g})K_Y. \tag{181}$$

Given Q, Eq. (179) becomes

$$\min_{\sigma} \quad \mathrm{Tr}(K_X Q). \tag{182}$$

This objective can be efficiently solved with BFGS.

Below in Fig. 20, we plot out the average within cluster kernel and the between cluster kernel values as we vary σ. From the plot, we can see that the maximum separation is discovered via BFGS.

Relation to HSIC. From Eq. (182), we can see that the σ that causes maximum kernel separation is directly related to HSIC. Given that the HSIC objective is normally written as

$$\min_{\sigma} \quad \mathrm{Tr}(K_X H K_Y H), \tag{183}$$

by setting $Q = HK_YH$, we can see how the two formulations are related. While the maximum kernel separation places the weight of each sample pair equally, HSIC weights the pair differently. We also notice that the $Q_{i,j}$ element is positive/negative for (x_i, x_j) pairs that are with/between classes respectively. Therefore, the argument for the global optimum should be relatively close for both objectives. Below in Figure 21, we show a figure of HSIC values as we vary σ. Notice how the optimal σ is almost equivalent to the solution from maximum kernel separation. For the purpose of *KNet*, we use σ that maximizes the HSIC value.

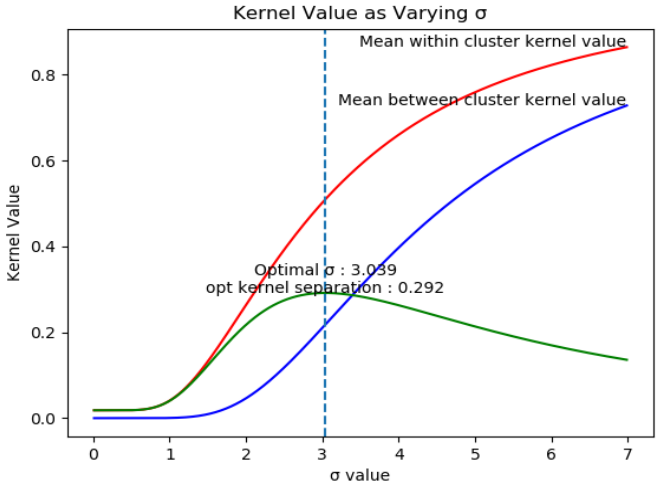

Figure 20: Maximum Kernel separation.

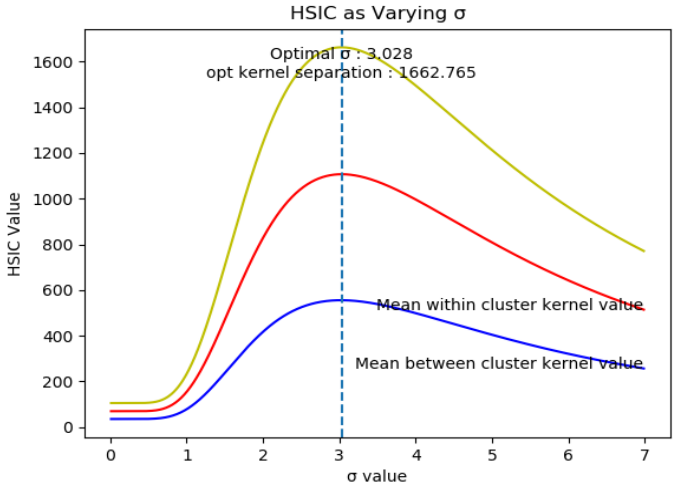

Figure 21: Maximal HSIC.