Modeling Neural Networks as Kernel Chains

Chieh Wu* [1] **Aria Masoomi*** [1] **Jennifer Dy** [1]

Abstract

Although Multi-layer perceptrons (MLPs) are commonly used today for a wide range of tasks, their theoretical foundations are still an active area of research. In this paper, we propose a new kernel framework to model MLPs as a chain of transitions between two spaces. However, instead of modeling the entire network chain monolithically, we discovered that under certain assumptions, the network behavior becomes predictable by modeling just the transitions themselves. This greatly simplifies the model and enables the network to be solved greedily. Moreover, we discovered that by modeling the transitions to maximize the Cross-Covariance Norm (CCN) between the layer outputs and the labels, the resulting sequence of layer outputs converges to a solution that simultaneously minimizes the Mean Squared Error (MSE) and the Cross-Entropy (CE). Indeed, our proof directly links MSE and CE back to the two spaces of our model. The network under our model can be solved greedily and deterministically via an existing spectral method. In so doing, our model voids exploding/vanishing gradients and automatically discovers the initialization weight, width, and depth of the network.

1. Introduction

Since the seminal work by Rumelhart et al. [1], Multilayer Perceptrons (MLPs) have become a popular tool for classification. Yet, the theoretical understanding of its inner workings remains an active area of research. As a result, the training process necessitates careful choice of network width, depth, initialization weights, and activation functions; the number of initial settings often exceeds even the number of samples. The careful management of the network settings and weights are required because the community is still building its theoretical foundation.

Much work has been dedicated to understanding the inner working of MLPs. Works by Cybenko [2]; Hornik [3]; Lu et al. [4] have demonstrated MLP's ability to approximate continuous functions on compact subsets of $\mathcal{R}^n$. Montufar et al. [5] and Poole et al. [6] further investigated MLP's expressive power in relation to their depth where they conclude that deeper is better. To maximize the expressiveness with deeper networks, others have concentrated on the appropriate initialization of the weights [7; 8; 9]. Besides investigating specific components of MLPs, others attempt to understand MLPs from the geometric/graph perspective [10; 11; 12; 13; 14; 15].

Departing from these past works, we are interested in MLP's theoretical connection to kernel methods. While more research is required to elucidate their relationship, their suspected connection is not without evidence. Among the first evidences, Montavon et al. [16] used Kernel PCA (KPCA) with a Gaussian kernel to obtain the weights of a single layer. By iteratively feeding this output into another KPCA, they simulate the layers of a neural network to analyze the evolution of the samples. As the layers increase, they observed that fewer principal components are required to achieve a low least squares error (LSE) against the labels, thereby demonstrating how stacks of kernels and MLPs can exhibit similar predictive capability while compressing the data.

Recently, Belkin et al. [17] have compared MLP and kernel's ability to yield good generalization results despite overfitting. Their experiments suggest that this key property of MLPs is also manifested in kernel methods; which led them to conclude that to truly understand MLPs, we need to better understand the kernel discovery process. Although these observations lacked theoretical analyses, they still provide tantalizing clues into the kernel's relationship to MLPs. Following these evidences, we focus on developing a framework to model MLPs by leveraging the large body of analytic guarantees developed by the kernel community.

Recently, MLPs have also been modeled as a Gaussian process (GP) [18; 19; 20]. This kernel interpretation has inspired a significant amount of research [21; 22; 23; 24]. Extending from the GP perspective, the Neural Tangent Kernel (NTK) has been proposed to describe the dynamics of the network during training [25; 26], further elucidating the relationship between MLPs to kernels.

Our kernel perspective fundamentally deviates from the GP paradigm by reinterpreting the MLP classification process. Namely, we model MLPs as iterative cyclic transitions between two different spaces. However, instead of modeling the entire network monolithically as a long chain of tran-

sitions, we discovered that under certain assumptions, the network behavior becomes predictable by modeling just the transitions itself. This greatly simplifies the model and enables the network to be solved greedily. We further discovered that by modeling the transitions to maximize the Cross-Covariance Norm (CCN) between the layer outputs and the labels, the resulting sequence of layer outputs converges to a solution that simultaneously minimizes the Mean Squared Error (MSE) and the Cross-Entropy (CE). Indeed, our theoretical results directly link MSE and CE to the two spaces of our model.

While CCN is not an objective commonly used for classification on MLPs, this possibility has been recently studied [27; 28]. Motivated by Information Theory, their successful usage of CCN for classification and clustering yields convincing results that establish CCN's legitimacy as an MLP objective. Surprisingly, while our analyses originated from vastly different motivations, our work converges to support the usage of CCN for MLPs.

In leveraging CCN as the empirical risk, we identified three properties to interpret its effects on the samples at each layer. Namely, our proofs describe how CCN incrementally compress the data at each layer while discovering an optimal kernel. Lastly, our MLP model yields a formulation that can be solved deterministically via a spectral method without SGD. In so doing, our model voids exploding/vanishing gradients and automatically discovers the initialization weight, width, and depth of the network. The source code along with all experiments are readily reproducible and publicly available on `https://github.com/anonymous`.

2. Network Model

Let $X \in \mathbb{R}^{n\times d}$ be a dataset of n samples with d features and let $Y \in \mathbb{R}^{n\times c}$ be the corresponding one-hot encoded labels with c as the number of classes. The i^{th} sample and label of the dataset is written as x_i and y_i. The kernel matrices associated with X and Y are $K_X, K_Y \in \mathbb{R}^{n\times n}$ where a Gaussian and a linear kernel is used respectively. H is a centering matrix defined as $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$: I_n is the identity matrix of size n and $\mathbf{1}_n$ is a vector of 1s also of length n.

A general MLP structure uses L layers of stacked neurons to minimize a chosen empirical risk ($\mathcal{E}$). Without a loss of generality, we assume to use the same activation function Ψ at each layer. We further generalize Ψ beyond the traditional functions, i.e., instead of assuming that the input dimension q and output dimension m of Ψ are equal, we generalize Ψ to also include functions where $m \neq q$.

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 respectively; assuming $l > 1$. The input and output at the l^{th} layer are $R_{l-1} \in \mathbb{R}^{n\times m}$ and $R_l \in \mathbb{R}^{n\times m}$ respectively, i.e., $R_l = \Psi(R_{l-1}W_l)$. For the network weights, we denote $\mathcal{W}_l$ as a function such that $\mathcal{W}_l(R_{l-1}) = R_{l-1}W_l$. Hence, each layer is itself a function $f_l : \mathbb{R}^m \to \mathbb{R}^m$ that operates on each sample individually, and is constructed by the composition of Ψ and $\mathcal{W}_l$, or $f_l = \Psi \circ \mathcal{W}_l$. Since the entire MLP stacks L layers together, the MLP itself is a function f consisting of the cumulative composition of all layers where $f = f_L \circ ... \circ f_1$. As a subproblem, we denote compositions of l functions as $f_{l\circ} = f_l \circ ... \circ f_1$ where $l \leq L$. This notation enables us to connect the data directly to the layer output where $R_l = f_{l\circ}(X)$. Finally, we learn f by minimizing empirical risk $\mathcal{E}$ with a loss function $\mathcal{L}$:

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

MLPs as an Iterative Cyclic Transition. We propose to model the MLP previously described as a cyclic transition between two spaces as shown in Fig. 1, i.e., Reproducing Kernel Hilbert Space (RKHS) and its linear subspace. Since the linear subspace is generated by applying q functions in the RKHS to all input samples, following the Riesz representation theorem, we refer to the linear subspace as the *images of the dual space* (IDS). The data is assumed to initially reside in RKHS. From there, each layer cycles from RKHS through W_l to IDS and Ψ back to RKHS. This process continues until the cycle stops at the last layer where its output is then used to compute $\mathcal{E}$. The idea of modeling MLPs as transitions of states is analogous to a Markov Chain (MC), i.e., the behavior of the entire network is encapsulated by the transition matrix and its initial state. Inspired by MC's ability to model a complex system of transitions, we discovered that the behavior of the entire MLP also becomes predictable if the transitions satisfy certain properties.

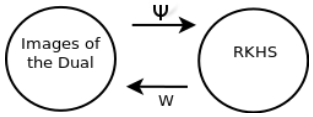

Figure 1. Flow of information

Since each cyclic transition is equivalent to a network layer, we can model the network at incremental depths as a sequence of functions $(f_{1\circ}, ..., f_{L\circ})$. Since each function in this sequence can be interpreted as a kernel feature map, we refer to this sequence as the *kernel sequence*. By setting each element of the *kernel sequence* as f in Eq. (1), it induces another sequence $(\mathcal{E}_1, ..., \mathcal{E}_L)$ of empirical risks which we will refer to as the *risk sequence*. By interpreting the transitions as a sequence, it models the entire MLP while simplifying its representation. From this perspective, our goal is to discover a transition model that generates *kernel*

sequences capable of solving classification problems. To accomplish this, we first identify the principal properties necessary to guide its design.

Properties of a Chain. Viewing the layers of MLPs as a sequence of functions enables the design of its behavior. This is because the pattern of the sequence itself can be manipulated by directing the transition model to exhibit certain properties. Specifically, we propose the following two properties which the *risk sequence* must satisfy.

Property 1. *The risk sequence must converge to a limit.*

Property 2. *The network weights $W_1, ..., W_L$ at convergence must satisfy the First Order and Second Order Necessary Conditions as defined by Bertsekas [29] for local optimality of the empirical risk.*

Instead of solving Eq. (1) directly via traditional approaches, we noticed that it can be alternatively optimized by satisfying Properties 1 and 2. This observation leads directly to the main proposal of our work which views each cyclic transition as a way to generate an element in the sequence. Specifically, we propose to solve Eq. (1) by applying the following proposition with its proof in App. A.

Proposition 1. *If each cyclic transition is greedily optimized to generate a monotonic risk sequence in a bounded space, then Properties 1 and 2 are satisfied.*

We note that Properties 1 and 2 can theoretically be used to satisfy any empirical risk. However, since our focus is on classification, we include an additional property in which the empirical risk must also satisfy. Namely, to perform classification, MLPs must also discover a mapping $\kappa\colon \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ where sample pairs of the *same* classes achieve a high score based on some notion of similarity. Alternatively, samples from *different* classes would yield a lower score. We refer to this requirement as the *affinity property* and constrain the potential empirical risks to a smaller subset where this property is satisfied. Formally, we state the following property:

Property 3. *The empirical risk must satisfy the affinity property defined as*

Definition 1. *Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (x_i, x_j) from a dataset X that belongs to the same and different classes respectively, let all $\mu_{i,j}$ values be non-negative scalars, and let κ be a similarity measure between any pairs of (x_i, x_j). An empirical risk satisfies the affinity property if there exists a κ such that its optimal solution f^* is also the optimal solution for the **affinity objective**, which we define here as*

$$\max_f \sum_{i,j\in\mathcal{S}} \mu_{i,j}\kappa(f(x_i), f(x_j)) - \sum_{i,j\in\mathcal{S}^c} \mu_{i,j}\kappa(f(x_i), f(x_j)).$$

While the first two properties place constraints on the transitional behavior of the *kernel/risk sequence*, property 3 is a constraint on the empirical risk. Since these three properties form the foundation of our work, we refer to them as the *KNet properties*. By extension, we refer to MLPs with these properties as Kernel Chain Networks or KNets. While these properties do not arise naturally, our work demonstrates that they can be induced by appropriately choosing the activation function, empirical risk, and optimization strategy.

Inducing a Converging *Risk Sequence*. Although there are many ways to generate converging sequences, here, we leverage the Monotone Convergence Theorem [30] where a monotone sequence is guaranteed to have a limit if and only if the sequence is bounded. Therefore, if each transition generates an $\mathcal{E}_l$ lower than $\mathcal{E}_{l-1}$ in a bounded space, the *risk sequence* is guaranteed to converge. Moreover, if the kernel sequence is constrained within the space of continuous functions, by leveraging Dini's Theorem [31], the convergence is not only pointwise but uniform.

In practice, both the function space for f and the solution space for $\mathcal{E}$ can be easily engineered. The solution space of $\mathcal{E}$s can be bounded by bounding all dependent variables. Inspired by the work on the geometry of the network solution landscape [14; 15], we bound the weights of each layer W_l on an orthogonal subspace, $W_l^T W_l = I$. Alternatively, the function space can be bounded by choosing a kernel whose RKHS consists of only continuous functions. Since the bounding of a function/solution space is easily realizable, the real challenge is identifying the conditions that guarantee a monotonic *risk sequence* as more layers are added to the network; more on this topic later.

Inducing an optimal *Risk Sequence* limit. Having a converging sequence is not sufficient unless its limit is an optimal $\mathcal{E}^*$. This can be accomplished by solving each layer greedily, i.e., instead of using the standard SGD with back-propagation to solve f_1 to f_L simultaneously, we can solve each layer using the output of the previous layer. Specifically, at the l^{th} layer we solve

$$\min_{f_l} \quad \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_{l^\circ}(x_i), y_i) \quad \text{s.t} \quad W_l^T W_l = I. \tag{2}$$

By solving $\mathcal{E}$ layer-wise, we ensure that every element of the *risk sequence* is generated at a local optimum, thereby ensuring an optimal $\mathcal{E}^*$ at the limit. This observation implies that by iteratively solving the transition model of Eq. (2), the entire MLP can be solved as suggested by Proposition 1.

Choosing the Activation Function and Empirical Risk. The ability of an MLP to satisfy the *KNet properties* is inextricably related to the choice of the activation function and the empirical risk. After evaluating several options, we discovered that each cyclic transition can be modeled

as a dependence maximization problem using the Cross-Covariance Norm (CCN) as $\mathcal{E}$. Indeed, we found that by maximizing each transition greedily via CCN, all 3 *KNet Properties* can be simultaneously satisfied.

First, since our CCN is computed from the Reproducing Kernel Hilbert Space (RKHS), it can be rewritten directly into the *affinity objective* using the inner product. We formalize this relationship in the following theorem with its proof in App. D.

Theorem 1. *CCN is an affinity objective.*

Second, we discovered that given the appropriate activation function, CCN greedily generates a converging *risk sequence*. Since f_l is constrained by the structure $\Psi \circ W_l$, the choice of the activation function Ψ completely determines the flexibility of f_{l°. Using CCN as $\mathcal{E}$, we proved the existence of a set of weights that monotonically improves the *risk sequence* within a bounded space when the feature map of a Gaussian kernel is used as Ψ. We formally state this claim in the following theorem and provide the implication along with its proof in App. F.

Theorem 2. *There exist a set of weights $W_1, ..., W_L$ such that a CCN objective (with a Gaussian kernel) generates a uniformly converging risk sequence.*

We emphasize that the Gaussian kernel feature map deviates away from traditional concepts of an activation function, Ψ. If we let r_l be a single sample at the output of the l^{th} layer, then Ψ traditionally operates *separately* on each element of the vector $W_{l+1}^T r_l \in \mathbb{R}^m$ to produce another vector of $\mathbb{R}^m$. However, given a Gaussian kernel, Ψ operates on the entire vector to produce an output in $\mathbb{R}^\infty$. In so doing, each layer achieves the flexibility to approximate any function on a compact subset of $\mathbb{R}^n$ given the Universal Approximation theorem [32].

3. Desirable Properties of CCN

Using CCN to model each layer has many desirable properties beyond satisfying the *KNet properties*. Namely, it is positioned to leverage the theoretical guarantees previously developed by the kernel community. Since CCN in our context is computed from RKHS, CCN in RKHS is equivalent to the Hilbert Schmidt Independence Criterion (HSIC) as proven by Gretton et al. [33]. Hence, at each layer, Eq. (2) with the CCN objective can be significantly simplified into

$$\max_{W_l} \quad \text{Tr}(\Gamma K_{R_{l-1}W_l}) \quad \text{s.t} \quad W_l^T W_l = I, \tag{3}$$

where $\Gamma = HK_YH$ and $K_{R_{l-1}W_l}$ is the kernel matrix obtained using the samples after the linear transformation.

While CCN and HSIC are equivalent in RKHS, HSIC has previously received considerable attention in the kernel community. Therefore, it is equipped with a significant body of theoretical analysis and guarantees, of which include a spectral method that solves Eq. (3). Besides its theoretical advantages, HSIC provides interpretable insights into the inner workings of the network. In fact, three additional *KNet properties* are induced by the usage of HSIC to help interpret the sample behavior throughout the network. Due to these advantages, this paper will focus on the HSIC formulation while consigning CCN as the motivating objective.

Interpretation of an MLP via HSIC. While Gretton et al. [33] interpreted HSIC as CCN in RKHS, we investigated deeper into HSIC's effect on each sample within a transition. This allows us to study and interpret how layer-wise sample behaviors contribute to achieve classification. Is there a general pattern that characterizes the layer-to-layer interaction? Or, are the results of each layer randomly jumping until a local minimum is reached?

Unexpectedly, by using HSIC, we discovered that each transition is optimizing different notions of affinity in IDS and RKHS. Following Theorem 2 by using Gaussian/Linear kernels for the data/label respectively, we present the following two properties with their proofs in App. G and H.

Property 4. *The global optimum of Eq. (3) is achieved in IDS when sample pairs from the same and different classes are mapped into Euclidean distances of 0 and supremum.*

Property 5. *The global optimum of Eq. (3) is achieved in RKHS when sample pairs from the same and different classes are mapped into angular distances of 0 and $\frac{\pi}{2}$.*

Properties 4 and 5 suggest that as the network cycle through its layers, the HSIC objective iteratively improves the *affinity property* in both IDS and RKHS with different similarity measures. Although these properties are specific to the Gaussian kernel, the interpretation can be easily extended. In fact, the proof for Theorem 1 signifies that the kernel directly defines κ in the *affinity objective*. We propose the following properties given this observation.

Property 6. *Given HSIC as $\mathcal{E}$, its kernel function induces the similarity measure of the affinity objective.*

Since kernels define relationships between samples, it is not surprising in hindsight that they define the κ in the *affinity objective*. However, these properties jointly offer an interpretation on how samples behave at each transition. The network is using κ to define different similarities in different spaces; it then then compresses similar data into c discernible points with different notions of distance.

Relating HSIC Properties to Previous Work. Interestingly, these three additional properties provide a consistent narrative supporting the experimental observations made by Montavon et al. [16] as previously mentioned in the introduction. By applying Property 5, we see that as l increases in KNet, each f_{l° maps the original data into orientations that

are increasingly aligned or orthogonal, thereby compressing the data into fewer and fewer eigenvectors. Consequently, fewer dominant eigenvectors are required to minimize LSE as more layers are added to the network. Our finding on HSIC's ability to compress the data is also consistent with the observations made by Ma et al. [27] where they related HSIC to Information Bottlenecks. Although this interesting connection is trivial to demonstrate, a brief proof and further discussions are provided in App. I.

From a network-level's perspective, although there are many layers stacked together, the entire network can still be interpreted as having a single highly flexible layer. Therefore, the cyclic transitions of the network is a way of identifying the optimal feature map f^* at the limit of the *kernel sequence*. This observation begins to bridge the connection between MLPs and the kernel discovery process: an important and under-explored topic proposed by Belkin and Niyogi [34].

Relating HSIC to Traditional Empirical Risks. Perhaps the most surprising result of our investigation is the discovery of HSIC's relationship to traditional empirical risks such as MSE and CE. Note how Properties 4 and 5 described the transformation of samples between the IDS and RKHS space. Therefore, vastly different patterns can be induced by keeping or discarding the last activation function Ψ after training. Specifically, we found that IDS and RKHS induce different results capable of solving MSE or CE, i.e., the same weights that optimizes HSIC also minimizes MSE and CE in their respective space. Assuming that HSIC and the Gaussian kernel is used, we propose the following two theorems with their proofs in App. J and K.

Theorem 3. *Given a change of basis, the argmax of the HSIC objective is equivalent to the argmin of the CE objective in RKHS.*

Theorem 4. *The MSE objective is minimized by the argmax of the HSIC objective in IDS.*

Theorems 3 and 4 suggests that while HSIC is maximizing the dependence between R_l and Y to generate a converging *risk sequence*, it is simultaneously minimizing both MSE and CE in different spaces.

4. MLP Optimization with HSIC

While using HSIC as the empirical risk provides significant interpretation advantages, its formulation as Eq. (3) can also be solved without SGD, thereby avoiding the many existing challenges. Specifically, the Iterative Spectral Method (ISM) has recently been proposed by Wu et al. [35; 36] to effectively solve the objective.

ISM Algorithm. Besides the theoretical advantages of ISM, it is a fast algorithm that can be easily implemented. Based on ISM, there exists a family of kernels where each kernel is associated with its own Φ matrix. Once Φ is calculated, the solution of Eq. (3) is simply its q most dominant eigenvectors. For completeness, we include the equation for the Gaussian Φ matrix along with the ISM algorithm in App. L: the theoretical guarantees can be found in the original work by Wu et al. [35; 36].

Advantages of Using ISM. By combining the *KNet Properties* with ISM, MLPs can be solved without backpropagation, thereby avoiding the issues of vanishing/exploding gradients. Additionally, since ISM guarantees both the 1st and 2nd order conditions, saddle points can be avoided while ensuring the optimality of every element within the *risk sequence*. Moreover, ISM has a closed-form solution to initialize the network weights, thereby removing the need to initialize the network via random values. Even the width and depth of the network can be automatically determined via ISM. Note that the width of the network is determined by the dimension of W_l. Since ISM is a spectral method, the dimension of W_l can be guided by Φ's rank, i.e., the width at each layer can be dynamically determined by maintaining Φ's most dominant eigenvalues. As for the depth of the network, it is simply the length of the *kernel sequence*.

Solving MLP via ISM. By using ISM to obtain W_l at each layer for KNet, the width of the network can be determined by the rank of Φ. Once W_l is obtained, the layer can be simulated via $R_l = \Psi(R_{l-1}W_l)$. Of course, since the feature map, Ψ, of a Gaussian kernel has infinite dimensions, we use a low dimensional approximation of Ψ via the Random Fourier Feature (RFF) [37]. Hence, the output of RFF becomes the output of the layer; which is again fed into the next layer. This process is repeated until the *affinity objective* meets a predefined threshold.

For reasons we will clarify in a later section, the Silhouette Score $\tilde{S}$ [38] is used as a surrogate to evaluate the *affinity objective* in IDS. After training the model, the resulting cluster centers of each class can be used to label the test samples by matching them to the nearest center. Here, we summarized the KNet algorithm in Algorithm 1.

Algorithm 1 KNet Algorithm

Input : Data X, Label Y
Output : Network weights $W_1, ..., W_L$
while $\tilde{S} < 0.95$ **do**
  Use the output of last layer as input
  Add a new layer
  Solve W_l via ISM
end

Complexity Analysis. The complexity analysis of a single ISM iteration as reported by Wu et al. [35] is $O(n^2)$. Since ISM is repeated base on the number of layers, KNet's complexity is simply $O(Ln^2)$. In terms of memory, KNet suffers the same $O(n^2)$ restriction due to the kernel computation

that all kernel methods inherit.

Limitations to ISM. Although our framework presents many theoretical advantages, we caution that much more research would be required for KNet to become practically viable. Instead of replacing existing practices, we offer a theoretical path to analyze and model MLPs based on the IDS/RKHS transition model. It is from this model that leads to the proposal of the *KNet properties*. These properties are the central contributions, while our choice of using HSIC and ISM is simply a convenient solution.

While ISM resolves many existing problems, it also limits the kernel function to the ISM family. Therefore, it currently cannot solve the traditional activation functions such as relu and sigmoid. Although we believe to already have solutions to overcome these challenges, we leave this discussion for later work. Another challenge is KNet's requirement for computing the kernel matrix. Therefore, KNet at its current maturity is intended for analysis and is not suitable for large datasets. Although this restriction can be mitigated via existing approximation algorithms [37] or by solving each layer stochastically as suggested by Ma et al. [27], these engineering questions are topics we purposely isolate away from the theory for future research.

5. Experiments

Datasets. We emphasize that the focus of the paper is the development of a theoretical framework to model MLPs, and not an engineering exercise to ensure KNet's effectiveness on all possible dataset. Therefore, our experiments specifically avoided large datasets where the memory complexity of kernel methods is still an ongoing research topic. Instead, we concentrate on small UCI datasets to confirm the theoretical claims proposed by the paper. Specifically, we investigated the inner workings of a KNet using one synthetic (spiral) and 5 popular UCI datasets: wine, cancer, car, divorce, and face [39]. These datasets were chosen to include a mixture of continuous, discrete, and image data. The synthetic spiral dataset is designed to only have 2 dimensions for visualization purposes. Additional data statistics can be found in App. M.

Evaluation Metrics. To evaluate the central claim that MLPs can be solved greedily (Proposition 1), we record the empirical risk (the HSIC value) along with the training/test accuracy for each dataset. Here, HSIC is normalized to the range between 0 to 1 via the normalization method used by Cortes et al. [40]. To corroborate Theorems 3 and 4, we also record MSE and CE results at each layer as samples propagate throughout the network.

To examine Property 4, we use Silhouette Score ($\tilde{S} \in [0, 1]$) as a surrogate to measure the compactness of samples between and within classes. A maximum $\tilde{S}$ of 1 implies that samples of the same class are packed tightly together while samples of different classes are far apart. Due to its ability to summarize the compactness of clusters in Euclidean space, it allows us to visualize the physical locations of each sample at each layer. Due to these interpretable advantages, it is used as a convenient surrogate to measure the converging status of the *kernel sequence*.

Alternatively for Property 5, we use the Cosine Similarity Ratio (CS) to evaluate the angular distance between samples in RKHS. CS computes the average inner product between samples of different classes and divide that value by the average inner product between samples of the same class. Therefore, CS is a surrogate to evaluate the angular distance between samples of the same and different classes with an ideal value of 0. For the exact equations used to compute these metrics, refer to App. N.

Experiment Settings. The width for RFF at each layer is set to 300, chosen based on the proof of Theorem 2. The dimension of subspace q which determines the width of the network is set by ISM to keep 90% of the data variance. The depth of the network is the length of the *risk sequence*. The *kernel sequence* is considered as converged at $\tilde{S} > 0.95$. The KNet initialization weights are initialized via ISM. The network structure and dimensions discovered by ISM for every dataset are also recorded and viewable in App. O. The MLPs that use MSE and CE have weights initialized via the Kaiming method [8]. 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 [41; 42; 43]. All experiments were conducted on an Intel Xeon(R) CPU E5-2630 v3 @ 2.40GHz x 16 with 16 total cores. The σ value that maximizes the HSIC objective is used in the Gaussian kernel. Our additional theoretical and experimental work to justify the σ discovering algorithm can be found in App. P.

Experimental Setup. Using HSIC as $\mathcal{E}$, we conduct 10-fold cross-validation across 5 datasets and recorded the key evaluation metrics. For each metric, the mean and the standard deviation from the 10-folds are reported. Once KNet is trained and has learned its structure, we use the same depth and width to trained 2 separate MLPs, where instead of HSIC, MSE and CE are used as $\mathcal{E}$ and trained via standard SGD. The accumulative results of all 3 objectives are listed in Column 1 of Table 1 for each dataset.

Once KNet is trained, will its optimal arguments also induce a low MSE and CE? We evaluate this prediction by keeping the same network weights while replacing the final objective with MSE and CE as $\mathcal{E}$ in their respective spaces. Based on Theorems 3 and 4, a high HSIC should directly induce a low MSE and CE. These results are highlighted in the columns of HSIC, MSE, and CE from Table 1.

To confirm Theorem 2, we recorded the progression of the

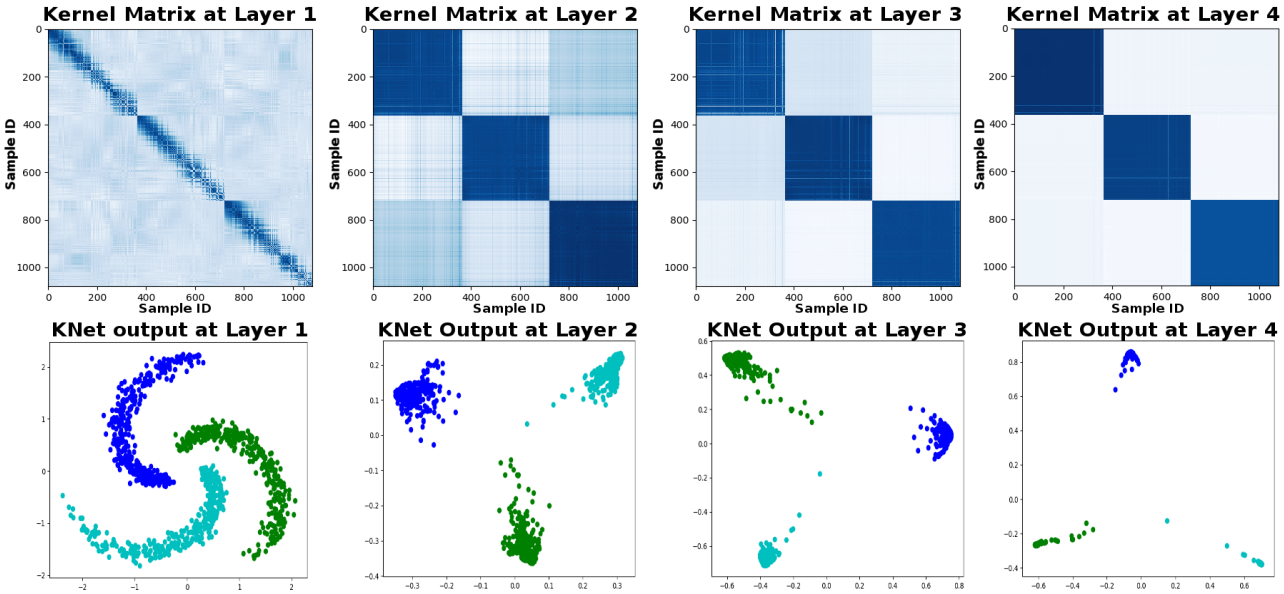

Figure 2. The top row is a visualization of the improving kernel matrices induced by the *kernel sequence* at each layer. The bottom row represents the output of each layer in IDS. The figures provide a visual confirmation of Theorem 4 and Property 4.

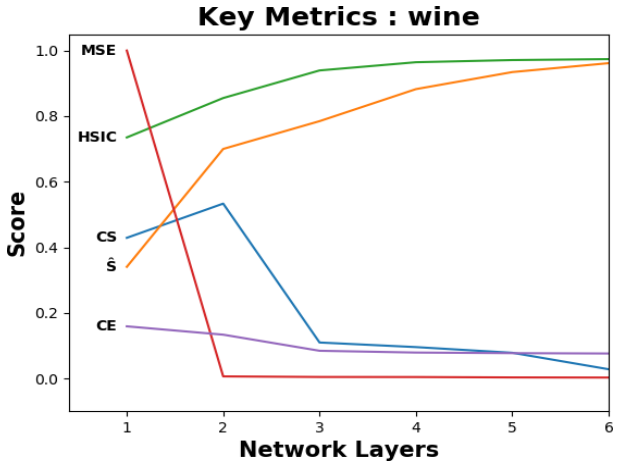

Figure 3. Key evaluation metrics of KNet recorded at each layer.

risk sequence along with several key metrics in Fig. 3. By evaluating the HSIC magnitude at each layer, we expect a uniform, monotonic, and converging pattern toward a global optimal of 1. Moreover, by also including $\tilde{S}$ and CS in the sample plot, it indicates if the samples are indeed being pulled together simultaneously in IDS and RKHS. Based on the proof of Properties 4 and 5, we expect an increasing $\tilde{S}$ and a decreasing CS. Lastly, the same plot is capable of validating Theorems 3 and 4 by including the MSE and CE errors at each layer.

Besides confirming the monotonicity of the *risk sequence*, we also wish to evaluate the quality of the *kernel sequence*. We accomplish this by plotting out the kernel matrix at each layer in the top row of Fig. 2. 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. Ideally, we wish to have a *kernel sequence* to evolve from an uninformative kernel into a highly discriminating kernel of perfect block structures. Corresponding to the top row, the bottom row plots the output of $f_{l^\circ}(X)$ at each layer in IDS to visually confirm Property 4.

Results. Since HSIC is used as the empirical risk, the high HSIC values (in blue) among the rows of the HSIC experiments in Table 1 corroborates with Proposition 1. Indeed, the empirical risk *can* be optimized by greedily solving each transition. In fact, as predicted by Theorem 1, a high HSIC value translates to a high training accuracy.

While Theorem 1 only provides guarantees on the training set, the accurate test results (in green) reflect KNet's ability for generalization. These results are also consistent with the observations made by Belkin et al. [17]. Indeed, while KNet is also trained using an infinitely wide network, it avoids overfitting despite the small sample sizes. Unexpectedly, KNet's ability for generalization is especially prominent on Face dataset (in **bold green**) where the input dimension consists of 960 features. While HSIC, MSE, and CE all achieved 100% accuracy on the training set, KNet was the only MLP structure that avoided overfitting on the test set.

Data / $\mathcal{E}$ Objective	Train Acc	Test Acc	Time(s)	HSIC	MSE	CE	CSR	Silhouette
wine/HSIC	0.991 ± 0.003	0.972 ± 0.028	**0.313 ± 0.068**	0.972 ± 0.011	**0.015 ± 0.006**	**0.075 ± 0.015**	0.033 ± 0.042	0.960 ± 0.005
wine/CE	1.000 ± 0.000	0.938 ± 0.073	2.425 ± 0.864	1.000 ± 0.000	38.2 ± 33.2	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
wine/MSE	1.000 ± 0.000	0.915 ± 0.106	65.6 ± 22.5	1.000 ± 0.000	0.000 ± 0.000	1.106 ± 0.049	0.497 ± 0.009	0.999 ± 0.001
cancer/HSIC	0.988 ± 0.002	0.967 ± 0.022	**3.076 ± 2.220**	0.959 ± 0.004	**0.022 ± 0.003**	**0.041 ± 0.007**	0.003 ± 0.042	0.956 ± 0.004
cancer/CE	1.000 ± 0.000	0.958 ± 0.020	51.750 ± 19.055	1.000 ± 0.000	1545.8 ± 3466.7	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
cancer/MSE	1.000 ± 0.000	0.977 ± 0.013	167.5 ± 28.5	1.000 ± 0.000	0.000 ± 0.000	0.714 ± 0.082	0.001 ± 0.004	0.999 ± 0.000
car/HSIC	1.000 ± 0.000	0.998 ± 0.005	**1.593 ± 0.371**	0.996 ± 0.003	**0.002 ± 0.001**	**0.008 ± 0.002**	0.036 ± 0.054	0.977 ± 0.008
car/CE	1.000 ± 0.000	1.000 ± 0.000	19.839 ± 7.064	1.000 ± 0.000	166.3 ± 150.9	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
car/MSE	1.000 ± 0.000	1.000 ± 0.000	522.5 ± 192.8	1.000 ± 0.000	0.000 ± 0.000	1.128 ± 0.040	0.400 ± 0.003	0.999 ± 0.000
face/HSIC	1.000 ± 0.000	**0.992 ± 0.008**	**0.795 ± 0.088**	0.968 ± 0.004	**0.001 ± 0.000**	**0.175 ± 0.002**	0.009 ± 0.004	0.977 ± 0.002
face/CE	1.000 ± 0.000	0.697 ± 0.307	35.738 ± 19.761	1.000 ± 0.000	16279.2 ± 10636.20	10.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
face/MSE	0.923 ± 0.104	0.515 ± 0.153	789.4 ± 318.6	0.923 ± 0.073	0.392 ± 0.589	3.407 ± 0.335	0.722 ± 0.014	0.805 ± 0.134
divorce/HSIC	0.999 ± 0.004	0.976 ± 0.039	**0.607 ± 0.412**	0.988 ± 0.011	**0.010 ± 0.011**	**0.027 ± 0.003**	0.014 ± 0.035	0.975 ± 0.013
divorce/CE	1.000 ± 0.000	0.982 ± 0.027	3.025 ± 1.007	1.000 ± 0.000	31.8 ± 35.2	0.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000
divorce/MSE	1.000 ± 0.000	0.953 ± 0.073	60.8 ± 30.8	1.000 ± 0.000	0.000 ± 0.000	0.704 ± 0.082	0.000 ± 0.001	0.999 ± 0.000

Table 1. Comparing the training/test results of HSIC to MSE and CE across 5 datasets. HSIC (in **bold**) ran significantly faster than MSE and CE and is the only empirical risk that simultaneously optimizes all evaluation metrics while maintaining a low test error.

The execution time (in **bold**) for each objective is also recorded for reference in Table 1. Since KNet can be solved via a single forward pass while SGD required many iterations of backpropagation, we expected ISM to solve the empirical risk faster. The Time column of Table 1 reflects this expectation by a wide margin. The biggest difference can be observed by comparing the face datasets, HSIC solved $\mathcal{E}$ with 0.795 seconds while MSE required 789 seconds; it is almost 1000 times difference. While the execution time results reflect our expectation, it is important to note that none of the datasets are large enough for $O(n^2)$ memory requirement to becomes a significant issue.

Since a high $\tilde{S}$ and a low CS (in red) together implies a maximization of the *affinity objective* in both IDS and RKHS. The Silhouette and CS columns from Table 1 support the claims of Theorem 1 as well as Properties 4 and 5. In other words, the *kernel sequence* converged to a mapping where samples of the same/different classes are being pulled together and pushed apart respectively in different spaces.

By studying the rows in Table 1 that used MSE and CE as objective, note that the minimization of one objective fails to minimize the other objectives. However, as predicted by Theorem 3 and 4, a low MSE and CE (in **bold**) can be achieved simultaneously by using the arguments trained on the HSIC objective to compute MSE and CE.

The HSIC progression between the layers is shown in Fig. 3 to validate Proposition 1 and Theorem 2. Since a monotonic *risk sequence* is a central requirement, the uniformly converging HSIC pattern toward 1 confirms both claims. Simultaneously, we also recorded other metric progressions and found repeatable consistency across all datasets that reinforce our theorems; the complete set of figures for all datasets can be found in App. R.

Notably, while all metrics exhibit a monotonic trend, CS often produces a "hunchback" pattern; a term coined by Ansuini et al. [44]. While studying AlexNet, VGG, and ResNet, they notice that as the data progress through the layers, their intrinsic dimension consistently expands prior to compression. Interestingly, we observed a similar pattern with our measure of dimension, the CS. Moreover, the hunchback pattern also agrees with Montavon's experiments. Since KPCA was used for each layer, the weights only come after Ψ. Therefore, they skipped the initial expansion layer and observed only the compression portion of the network. To corroborate with their results, we also recorded the number of eigenvectors used at each layer and found a consistent pattern of compression. Since these interesting results are not central to the claims of this paper, we leave the additional results and further discussion to App. O.

We lastly produce a visual evidence on the quality of the *kernel sequence* in Fig. 2. The figure shows an incrementally improving block structure as the sequence converges, thereby discovering an optimal kernel at convergence. The bottom row represents the output of each function in the sequence. As predicted by Property 4, 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. Q.

Conclusion. We have presented a novel approach to interpret, model, and predict the behavior of MLPs for classification. The central contribution of our work is the proposal of the IDS/RKHS transition model which leads to the *KNet properties* as conditions required for analysis. To satisfy these conditions, we discovered that HSIC using a Gaussian kernel satisfies these properties while simultaneously solves MSE/CE in IDE/RKHS. Moreover, HSIC provides a convenient interpretation of MLP's effect on each sample. Indeed, these observations are predictable by our theorems and experimentally reproducible. Therefore, KNets open the door to a new method to analyze MLPs.

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

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

[4] 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.

[5] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[6] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368, 2016.

[7] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[8] 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.

[9] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

[10] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[11] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[12] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[13] Na Lei, Zhongxuan Luo, Shing-Tung Yau, and David Xianfeng Gu. Geometric understanding of deep learning. *arXiv preprint arXiv:1805.10451*, 2018.

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

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

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

[17] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *arXiv preprint arXiv:1802.01396*, 2018.

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

[19] 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.

[20] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] Chieh Wu, Zulqarnain Khan, Yale Chang, Stratis Ioannidis, and Jennifer Dy. Deep kernel learning for clustering, 2019.

[29] Dimitri P Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3):334–334, 1997.

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

[31] Robert Gardner Bartle and Donald R Sherbert. *Introduction to real analysis*, volume 2. Wiley New York, 1992.

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

[33] 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.

[34] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[35] 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.

[36] Chieh Wu, Jared Miller, Yale Chang, Mario Sznaier, and Jennifer Dy. Solving interpretable kernel dimension reduction. *arXiv preprint arXiv:1909.03093*, 2019.

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

[38] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

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

[40] 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.

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

[42] 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.

[43] 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.

[44] 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.

[45] Bharath K Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Gert Lanckriet, and Bernhard Schölkopf. Injective hilbert space embeddings of probability measures. In *21st Annual Conference on Learning Theory (COLT 2008)*, pages 111–122. Omnipress, 2008.

[46] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

A. Proof for Proposition 1

Theorem 1 : *If each cyclic transition is greedily optimized to generate a monotonic sequence in a bounded space, then Properties 1 and 2 are satisfied.*

Proof. To begin our proof, we first evoke the Monotone Convergence Theorem [30] which states that a monotone sequence is guaranteed to have a limit if and only if the sequence is bounded. Since this condition is assumed to be true in Theorem 1, then it follows that the sequence generated by the cyclic transitions must also converge to a limit. Hence, Property 1 is satisfied.

As for Property 2, the limit of the sequence must satisfy the 2nd order condition. By assuming that the MLP is solved greedily, this implies that the output of the previous layer is used as input for the next layer. Therefore, each cycle is its own optimization problem. Assuming that each cyclic transition is solved to satisfy the 2nd order condition, then every element of the *risk sequence* must also satisfy the 2nd order condition. Therefore guaranteeing the limit of the sequence to satisfy the 2nd order condition and Property 2. □

B. 2nd Order Necessary Conditions

Lemma 1 (Bertsekas, Proposition 3.1.1 [29])**.** *(Lagrange Multiplier Theorem - Necessary Conditions) Consider the optimization problem:* $\min_{W:h(W)=0} f(W)$*, where* $f : \mathbb{R}^{d\times q} \to \mathbb{R}$ *and* $h : \mathcal{R}^{d\times q} \to \mathbb{R}^{q\times q}$ *are twice continuously differentiable. Let* $\mathcal{L}$ *be the Lagrangian and* $h(W)$ *its equality constraint. Then, a local minimum must satisfy the following conditions:*

$$\nabla_W \mathcal{L}(W^*, \Lambda^*) = 0, \tag{4a}$$

$$\nabla_\Lambda \mathcal{L}(W^*, \Lambda^*) = 0, \tag{4b}$$

$$\begin{aligned} &\mathrm{Tr}(Z^T \nabla^2_{WW} \mathcal{L}(W^*, \Lambda^*) Z) \geq 0 \\ &\quad \text{for all } Z \neq 0, \text{with } \nabla h(W^*)^T Z = 0. \end{aligned} \tag{4c}$$

C. Lemma 2

C.1. Assumptions and Notations of the Proof

Since the positive and negative properties of the Γ matrix are used throughout several proofs in the paper, we prove its properties as a lemma here. Following the convention of the paper, we are given a dataset $X \in \mathbb{R}^{N\times d}$ of C classes where N denotes the number of samples and d the dimensions of the data. Given C classes, we let n_c be the number of samples in the cth class. Let its classification labels use one-hot encoding and denoted as $Y \in \mathbb{R}^{N\times C}$. Also let H be the centering matrix defined as $H = I - \frac{1}{N}\mathbf{1}_{N\times N}$ where $\mathbf{1}_{N\times N} \in \mathbb{R}^{N\times N}$ represents a matrix of all 1s. Since H is a centering matrix, it has the property $H = H^T$. Lastly, let $\Gamma \in \mathbb{R}^{N\times N}$ matrix be defined as $\Gamma = HYY^TH$.

C.2. Lemmas of the Proof

Lemma 2. *If the size of any single class is less than the size of the union of any other two classes, then every* (x_i, x_j) *pair of samples of the same class has a corresponding positive* $\Gamma_{i,j}$ *and every* (x_i, x_j) *pair of samples of different classes has a corresponding negative* $\Gamma_{i,j}$*.*

Proof. Given C classes, the total number of samples N is the summation of the number of samples, n_c, for each class where

$$N = \sum_{c=1} n_c. \tag{5}$$

It can be easily verified that the linear kernel matrix $K = YY^T$ has (i, j)th element as 1 and 0 if the (x_i, x_j) pair belong to the same and different classes respectively. It can also be easily verified that $K = K^T$. We denote the constant η here as

$$\eta = \mathbf{1}_N^T K \mathbf{1}_N \tag{6}$$

where $\mathbf{1}_N$ is a vector of 1s of length N. In other words, η represents the cardinality of all (x_i, x_j) pairs that belong to the same classes and can also be computed as

$$\eta = \sum_{c=1} n_c^2. \tag{7}$$

We next define the degree vector as

$$d = [d_1, d_2, ...d_N]^T = K\mathbf{1}_N \tag{8}$$

and the stacked degree matrix $D \in \mathbb{R}^{N\times N}$ as

$$D = [d, d, ...]. \tag{9}$$

To clarify, the columns of D consists of repeated vectors of d. Given these relationship, we first apply the centering matrix to K and obtain

$$HK = (I - \frac{1}{N}\mathbf{1}_{N\times N})K = K - \frac{1}{N}D^T. \tag{10}$$

Next, we apply the centering matrix on the right hand side and obtain

$$\begin{aligned}
\Gamma = HKH &= ((HKH)^T)^T \\
&= (H(HK)^T)^T \\
&= (H(K - \frac{1}{N}D^T)^T)^T \\
&= ((I - \frac{1}{N}\mathbf{1}_{N\times N})(K^T - \frac{1}{N}D))^T \\
&= K - \frac{1}{N}D^T - \frac{1}{N}D + \frac{\eta}{N^2}\mathbf{1}_{N\times N}.
\end{aligned}$$

Therefore, at each element of Γ we get

$$\Gamma_{i,j} = K_{i,j} - \frac{1}{N}(d_i + d_j) + \frac{\eta}{N^2}. \tag{11}$$

First, we assume that the (x_i, x_j) pair belong to the same class of size n_κ, and therefore $K_{i,j} = 1$, we obtain

$$\begin{aligned}
\Gamma_{i,j} &= 1 - \frac{2n_\kappa}{N} + \frac{\eta}{N^2} \\
&= \frac{N^2}{N^2} - \frac{2n_\kappa N}{N^2} + \frac{\eta}{N^2}.
\end{aligned}$$

Since we want to show that this value is always positive, the division by N^2 can be ignored. Hence, we obtain

$$\begin{aligned}
\frac{N^2}{N^2} - \frac{2n_\kappa N}{N^2} + \frac{\eta}{N^2} &> 0 \\
N^2 - 2n_\kappa N + \eta &> 0 \\
\left[\sum_{c=1} n_c\right]\left[\sum_{c=1} n_c\right] - 2n_\kappa \sum_{c=1} n_c + \sum_{c=1} n_c^2 &> 0
\end{aligned}$$

Since $\sum_{c=1} n_c^2$ is always positive, it is always greater than 0. For now we ignore this term and focus on proving the following inequality.

$$\begin{aligned}
\left[\sum_{c=1} n_c\right]\left[\sum_{c=1} n_c\right] &> 2n_\kappa \left[\sum_{c=1} n_c\right] \\
\left[\sum_{c=1} n_c\right] &> 2n_\kappa \\
n_1 + n_2 + ... + n_k + ... + n_C &> 2n_k \\
n_1 + n_2 + ... + 0 + ... + n_C &> n_k
\end{aligned}$$

Since n_k by assumption cannot be greater than the summation of any 2 classes, the above inequality is always true.

Next, we show that if (x_i, x_j) are not in the same class, the $\Gamma_{i,j}$ value is always negative. In this case, $K_{i,j} = 0$ and therefore, we get

$$\begin{aligned}\Gamma_{i,j} &= 0 - \frac{1}{N}(d_i + d_j) + \frac{\eta}{N^2} \\ &= \frac{\eta}{N^2} - \frac{N}{N^2}(d_i + d_j).\end{aligned}$$

Again, we can ignore the denominator. We let samples x_i and x_j come from two distinct class group of sizes n_α and n_β. To obtain a negative value for $\Gamma_{i,j}$ we need to satisfy the inequality

$$\begin{aligned}\eta - N(d_i + d_j) &< 0 \\ \sum_{c=1} n_c^2 - (n_\alpha + n_\beta)(\sum_{c=1} n_c) &< 0 \\ (n_\alpha^2 + n_\beta^2) - (n_\alpha^2 + n_\beta^2) - 2(n_\alpha n_\beta) - \sum_{c=1} n_c(n_c - n_\alpha - n_\beta) &< 0 \\ -2(n_\alpha n_\beta) - \sum_{c=1} n_c(n_c - n_\alpha - n_\beta) &< 0\end{aligned}$$

Since $-2(n_\alpha n_\beta)$ is always negative, we can ignore this term. In fact, since n_c is always positive, we can also remove it from the 2nd term and simply need to show that for all c

$$\begin{aligned}n_c - n_\alpha - n_\beta &< 0, \\ n_c &< n_\alpha + n_\beta.\end{aligned}$$

Since the size of any class is always smaller than the size of the union any two classes, the inequality must be true. Hence, its corresponding $\Gamma_{i,j}$ will also always be negative. □

It is important to note that our proof is simply a theoretical exercise to match the formulation with HSIC. In practice, since the labels are known, $\Gamma_{i,j}$ can always be set to 1 and -1 based on the label. This is equivalent to changing the definition of the one-hot encoded labels. Therefore, the assumption that any class cannot be larger than the combination of any other two classes can be easily circumvented if the data is highly unbalanced. In general, this proof recommends to set $\Gamma_{i,j}$ directly to positive and negative 1 instead of relying on the centering matrix.

D. Proof for Theorem 1

Theorem 1: *CCN is an affinity objective.*

Proof. Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (x_i, x_j) from a dataset X with labels Y that belongs to the same and different classes respectively, let $\Gamma_{i,j}$ be a set of scalar values, and let κ be a similarity measure between any pairs of (x_i, x_j).

To prove Theorem 1, we first leverage the result from Gretton et al. [33] and rewrite the CCN objective as

$$\max_W \quad \mathrm{Tr}(K_{XW} H K_Y H) \qquad \text{s.t} \qquad W^T W = I. \tag{12}$$

If we let $\Gamma = HK_YH$ and convert the trace into a sum, then Eq. (12) becomes

$$\max_W \quad \sum_{i,j} \Gamma_{i,j} K_{(XW)_{i,j}} \qquad \text{s.t} \qquad W^T W = I. \tag{13}$$

By the definition of kernel matrices, $K_{(XW)_{i,j}}$ is the inner product of $\Psi(W^T x_i)$ and $\Psi(W^T x_j)$, or $\langle \Psi(W^T x_i), \Psi(W^T x_j) \rangle$. By applying Lemma 2 in Appendix C, Eq. (13) can be split into positive and negative $\Gamma_{i,j}$s where

$$\max_W \quad \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \langle \Psi(W^T x_i), \Psi(W^T x_j) \rangle - \sum_{i,j \in \mathcal{S}^c} \Gamma_{i,j} \langle \Psi(W^T x_i), \Psi(W^T x_j) \rangle \;\; \text{s.t} \;\; W^T W = I. \tag{14}$$

Since the inner product between unit vectors is a similarity measure of the angular distance, by setting $\kappa(.,.)$ as the inner product and letting $f = \Psi \circ W$ the theorem is proven. □

E. Proof for Theorem 2

Given:

$$\min_{W_l} \quad -\text{HSIC}(R_{l-1}W_l, Y) \quad \text{s. t: } W_l^T W_l = I. \tag{15}$$

Theorem 2: *Given a **risk sequence** of globally optimized empirical risks $(\mathcal{E}_1^*, ..., \mathcal{E}_L^*)$ that is individually generated via the Greedy method with Eq. (15) as proposed by Algorithm 1. If a RBF kernel is used, the **risk sequence** monotonically decreases and converges.*

E.1. Assumptions and Notations of the Proof

Here, we let R_{l-1} be the input to the lth layer where r_i represents the ith sample of R_{l-1}. Given W_l as the weight of the lth layer and let the non-linear activation function at the lth layer is denoted as ϕ_l. Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (r_i, r_j) from a dataset R_{l-1} that belongs to the same and different classes respectively. Corresponding to the data matrix R_{l-1} is the label matrix $Y \in \mathbb{R}^{n \times c}$ where c denotes the number of classes. Each label $y_i \in \mathbb{R}^c$ is in the one-hot encoding format.

We also here note the definition of an injective mapping, i.e., ϕ is injective if

$$x_i \neq x_j \rightarrow \phi(x_i) \neq \phi(x_j). \tag{16}$$

Although there are many ways to generate converging sequences, here, we leverage the Monotone Convergence Theorem [30] where a monotone sequence is guaranteed to have a limit if and only if the sequence is bounded. Therefore, if each transition generates an $\mathcal{E}_l$ lower than $\mathcal{E}_{l-1}$ in a bounded space, the *risk sequence* is guaranteed to converge, i.e., the monotonic decrease of the *risk sequence* directly implies a convergence.

If we let $\Gamma = HK_YH$, then Eq. (15) can be reformulated into

$$\max_{W_l} \sum_{i,j} \Gamma_{i,j} K_{R_{l-1}W_{l_{i,j}}} \quad \text{s. t: } W_l^T W_l = I. \tag{17}$$

Since $\Gamma_{i,j}$ came directly from the label, it is a bounded value, i.e, the elements of th Γ matrix is bounded. In addition, since the kernel matrix is the inner product of the elements in RKHS, and each function is bounded in L_2, the kernel matrix itself is also bounded. Together, the empirical risk $\mathcal{E} = \sum_{i,j} \Gamma_{i,j} K_{R_{l-1}W_{l_{i,j}}}$ must also exist in a bounded space. Since the space for $\mathcal{E}$ is bounded, the key to prove Thm. 2 is to demonstrate the monotonicity of *risk sequence* by confirming the inequality

$$\mathcal{E}_l^* \geq \mathcal{E}_{l+1}^*. \tag{18}$$

Since the empirical risks at local minimums can vary wildly, for theoretical analysis purpose, we assume that the sequence is generated only at the global minimum at each stage. In addition, we assume that an algorithm capable of obtaining the global is given. Specifically, we assume to be given an algorithm capable of solving Eq. (15) while satisfying the First and Second Order Necessary Conditions as defined by Bertsekas [29]. Given these assumption, the contribution of the prove is to show that in an ideal case, applying the feature map of RBF kernels as the non-linearity of a network layer leads to a sequence that satisfies Inequality (18).

Proof. We first note that $f_{l\circ} = f_l \circ f_{l-1} \circ ... \circ f_1$, therefore the globally optimal argument of Eq. (19) can be rewritten and solved as

$$W_l^* = \arg\max_{W_l} \quad \sum_{i,j} \Gamma_{i,j} \langle f_{l\circ}(x_i), f_{l\circ}(x_j) \rangle \quad \text{s. t: } W_l^T W_l = I. \tag{19}$$

If we let the set $\mathcal{Q}_l$ be the space of potential W matrices for Eq. (19), then W_l^* is the optimal W at layer l within the set $\mathcal{Q}_l$. Next, if we let $\mathcal{Q}_{l+1}$ be the space of potential W matrices at the $l+1$ layer. If we can show that W_l^* is also inside the set of $\mathcal{Q}_{l+1}$, then the best W matrix in $\mathcal{Q}_{l+1}$ must be at least as good as W_l^*, and therefore, $\mathcal{E}_l^* \geq \mathcal{E}_{l+1}^*$ is also true.

We next note that Eq. (19) using a RBF kernel can also be written as

$$W_l^* = \arg\max_{W_l} \quad \sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W_l W_l^T (r_i - r_j)}{2\sigma^2}} \qquad \text{s. t: } W_l^T W_l = I. \tag{20}$$

Since $\Gamma_{i,j}$ is positive for all $(r_i, r_j) \in \mathcal{S}$ and negative for $(r_i, r_j) \in \mathcal{S}^c$, Eq. (20) can be further broken into

$$W_l^* = \arg\max_{W_l} \quad \underbrace{\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W_l W_l^T (r_i - r_j)}{2\sigma^2}}}_{\text{1st term}} - \underbrace{\sum_{i,j \in \mathcal{S}^c} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W_l W_l^T (r_i - r_j)}{2\sigma^2}}}_{\text{2nd term}} \qquad \text{s. t: } W_l^T W_l = I. \tag{21}$$

Since the $\Gamma_{i,j}$ of Eq. (21) are all positive for the 1st term and negative for the 2nd term. It immediately tells us what the global optimal solution should be. Namely, we wish for the first exponential terms to all be 1s, and the 2nd expoential term to all be 0s. This solution is possible if there exists a W_l^* such that

$$\underbrace{W_l^T (r_i - r_j) = 0}_{\text{Condition 1}} \quad \forall \quad (r_i, r_j) \in \mathcal{S} \tag{22}$$

and

$$\underbrace{W_l^T (r_i - r_j) > 0}_{\text{Condition 2}} \quad \forall \quad (r_i, r_j) \in \mathcal{S}^c \tag{23}$$

We emphasize that $W_l^T (r_i - r_j) > 0$ is a sufficient condition because $\sigma > 0$ can be set arbitrarily small for the 2nd exponential term to approach 0. Specifically, we note that if W_l^* exists, then when σ is set arbitrarily small the 1st term becomes

$$\underbrace{\lim_{\sigma \to 0} e^{-\frac{(r_i - r_j)^T W_l W_l^T (r_i - r_j)}{2\sigma^2}}}_{\text{1st term}} = 1 \quad \forall i, j \in \mathcal{S}, \tag{24}$$

and the 2nd term becomes

$$\underbrace{\lim_{\sigma \to 0} e^{-\frac{(r_i - r_j)^T W_l W_l^T (r_i - r_j)}{2\sigma^2}}}_{\text{1st term}} = 0 \quad \forall i, j \in \mathcal{S}^c. \tag{25}$$

Therefore, as $\sigma \to 0$, $W_l^T (r_i - r_j) > 0$ and $W_l^T (r_i - r_j) >> 0$ for all $(r_i, r_j) \in \mathcal{S}^c$ produces the same empirical risk $\mathcal{E}_l$. Our theorem focuses on the usage of the RBF kernels because as $\sigma \to 0$, it simplifies each element of our *risk sequence* into two possibilities when a layered is optimized:

1. Condition 1 and 2 are satisfied (global optimal solution).
2. Condition 1 failed and condition 2 is satisfied (not global optimal solution).

We claim that if the $(l-1)$th layer is in the first possible situation, it will always stay in the 1st situation given the next additional layer. However if the $(l-1)$th layer is in the 2nd situation,

having a tuning σ

allows us

is an injective mapping [45].

possible for f_{l+1} to be an identity map, i.e.,

$$f_{l+1} \circ f_l(x) = f_l(x). \tag{26}$$

This possibility implies that the option of not changing Eq. (19) is a potential solution when f_{l+1} is added. And if not changing the equation is a option, W_l^* must be within the set of $\mathcal{Q}_{l+1}$. To finalize the prove, we provide the mapping where a RBF kernel with an W_{l+1} can induce a f_{l+1} that is an identity map.

, i.e., if we let the feature map of a characteristic kernel be ϕ, then there exists a ϕ^{-1} such that $\phi^{-1}[\phi(R_{l-1})] = R_{l-1}$. Additionally, we know that $W \in \mathbb{R}^{d \times q}$ where d is the dimension of its input and q can range from 1 to ∞, W^{-1} also exists in the potential solution space. Since the composition of an injective W and ϕ forms f_{l+1}, an injective f_{l+1} exists within the set of potential solutions. Therefore, at the $l+1$th layer, it can $\square$

F. Proof for Theorem 2

Theorem 2: For Gaussian kernels, there exist a set of weights $W_1, ..., W_L$ such that the CCN objective generates a uniformly converging *risk sequence*

F.1. Assumptions and Notations of the Proof

Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (x_i, x_j) from a dataset $X \in \mathbb{R}^{n\times d}$ that belongs to the same and different classes respectively. Corresponding to the data matrix X is the label matrix $Y \in \mathbb{R}^{n\times c}$ where c denotes the number of classes. Each label $y_i \in \mathbb{R}^c$ is in the one-hot encoding format. Let $\Gamma_{i,j}$ be a set of scalars, and let κ be a similarity measure between any pairs of (x_i, x_j).

By leveraging the Monotone Convergence Theorem, we know that a monotonic sequence in a bounded space converges to a fixed point. Since the solution space is bounded via the constraint of $W^TW = I$, the objective is to prove the monotonic improvements. This can be accomplished by proving the existence of a solution W_l for the following inequality

$$\max_{f_l} \quad ||\text{Cov}(f_{l^\circ}(X), Y)||_F \geq ||\text{Cov}(f^*_{(l-1)^\circ}(X), Y)||_F. \tag{27}$$

Here, we denote $f^*_{(l-1)^\circ}$ as the optimal mapping discovered from the previous layer. The goal of the proof is to show the existence of f_l such that f_{l° will produce an equal or higher dependence at the current layer.

Note that by using the Gaussian feature map as the activation function, the theoretical dimension of the mapping output is infinity with a norm of 1. Given an input sample x_i, we define r_i such that $r_i = f^*_{(l-1)^\circ}(x_i)$. Following this notation, we let $R_{l-1} = [r_1, r_2, ..., r_n]^T$, therefore, $R_{l-1} = f^*_{(l-1)^\circ}(X)$ where $||r_i|| = 1$ for all i. In addition, by using a Gaussian kernel, the inner product between any two samples is bounded between 0 and 1 where $0 \leq r_i^T r_j \leq 1$. We also leverage Dini's Theorem [31] in our proof, which states:

Dini's Theorem. If a monotone sequence of continuous functions converges pointwise on a compact space and if the limit function is also continuous, then the convergence is uniform.

The proof for Theorem 2 is divided into 2 Lemmas, i.e., lemmas 3 and 4. In the next subsection, we provide a brief summary of how each lemma combine to prove the theorem.

F.2. Summary of the Proof

In Lemma 3, we demonstrate that when a Gaussian kernel is used with HSIC, the argmax of the HSIC objective is equivalent to the argmax of an *affinity objective* using a negative Euclidean distance as a similarity measure. We refer to this objective as the *Euclidean Affinity* defined as

$$\arg\max_{W} \quad \text{Tr}(W^T A_= W) - \text{Tr}(W^T A_{\neq} W) \;\; \text{s.t} \;\; W^TW = I. \tag{28}$$

$A_=$ is defined as

$$A_= = \frac{1}{\sigma^2} \sum_{i,j\in\mathcal{S}} (r_i - r_j)(r_i - r_j)^T, \tag{29}$$

and $A_{\neq}$ is defined as

$$A_{\neq} = \frac{1}{\sigma^2} \sum_{i,j\in\tilde{\mathcal{S}}} (r_i - r_j)(r_i - r_j)^T. \tag{30}$$

Therefore, instead of solving the HSIC objective directly, we can optimize the *Euclidean Affinity* as a surrogate objective since the argmax for both objectives are equivalent.

In Lemma 4, we first define V as the set of eigenvectors of $A_{\neq}$ that have positive eigenvalues. Given V, we show that if the null space of $A_=$ is sufficiently large to intersect with the span of V, then a globally optimal solution W^* exists. Since the null space of $A_=$ for a Gaussian kernel is infinitely large, unless the span of $A_=$ completely overlaps the span of V (not possible), an intersection always exists, i.e., a global optimal W^* exists. Given the existence of W^*, there must also exist a set of Ws, $\{W_1, ..., W^*\}$, sampled from interpolation points between a point within

the neighborhood of W^* and W^* that monotonically improves the objective. Lastly, since the Gaussian kernel is a Universal kernel, all of its functions in RKHS are continuous, therefore, the monotonic convergence is also uniform.

F.3. Lemmas of the Proof

Lemma 3. *Assuming that the feature map of a Gaussian kernel is used as the activation function, then the LHS of the Inequality (27) we denote here as*

$$\arg\max_{W} \quad ||Cov(\Psi(R_{l-1}W), Y)||_F \quad s.t: \quad W^TW = I \tag{31}$$

is equivalent to Eq. (28).

Proof. Following the proof by Gretton et al. [33], the norm of the cross-covariance can be reformulated into the empirical HSIC objective where the inequality becomes

$$||\text{Cov}(\Psi(R_{l-1}W), Y)||_F = HSIC(R_{l-1}W, Y). \tag{32}$$

Therefore, optimizing the CCN objective is equivalent to solving

$$\arg\max_{W} \quad \text{Tr}(K_{R_lW}HK_YH) \quad s.t: \quad W^TW = I. \tag{33}$$

If we let $\Gamma_{i,j}$ be the $(i,j)th$ element of the matrix $\Gamma = HK_YH$ and write out the Gaussian kernel, the objective becomes

$$\arg\max_{W} \quad \sum_{i,j} \Gamma_{i,j} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} \quad s.t \quad W^TW = I. \tag{34}$$

Following Lemma 2, $\Gamma_{i,j}$ is positive if (r_i, r_j) pair belongs to the same class, and it is negative when the pair are in different classes. We can split the summation into the positive and negative $\Gamma_{i,j}$ pairs.

$$\arg\max_{W} \quad \sum_{i,j\in S} \Gamma_{i,j} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} - \sum_{i,j\in S^c} \Gamma_{i,j} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} \quad s.t \quad W^TW = I. \tag{35}$$

From Eq. (35), we see that given an appropriate σ value the global optimal solution is achieved when the exponential terms are either 0 or 1 such that

$$\max_{W} \quad \sum_{i,j\in S} \Gamma_{i,j} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} + \sum_{i,j\in S^c} -\Gamma_{i,j} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} = \sum_{i,j\in S} \Gamma_{i,j}[1] - \sum_{i,j\in S^c} \Gamma_{i,j}[0]. \tag{36}$$

Therefore, if there exists a W^* that satisfies the following two conditions, W^* must also be the global optimal.

$$\forall i,j \in S \quad \lim_{\sigma\to 0} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} = 1 \tag{37}$$

$$\forall i,j \in S^c \quad \lim_{\sigma\to 0} e^{-\frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2}} = 0. \tag{38}$$

Next, we consider the global optimal of the objective

$$\arg\min_{W} \quad \frac{1}{\sigma^2} \sum_{i,j\in\mathcal{S}} ||W^Tr_i - W^Tr_j||_2^2 - \frac{1}{\sigma^2} \sum_{i,j\in\mathcal{S}^c} ||W^Tr_i - W^Tr_j||_2^2 \quad \text{s.t}: \quad W^TW = I. \tag{39}$$

Similarly, we can state that if there exists a W^* that satisfies the following two conditions, W^* must also be the global optimal of Eq. (39).

$$\forall i,j \in S \quad \lim_{\sigma\to 0} \frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2} = 0 \tag{40}$$

$$\forall i,j \in S^c \quad \lim_{\sigma\to 0} \frac{||W^Tr_i - W^Tr_j||^2}{2\sigma^2} = \infty. \tag{41}$$

Notice that the W^* that satisfies conditions Eq. (40) and (41) is the same W^* that satisfies Eq. (37) and (38). Therefore, if the global optimal of W^* for Eq. (39) satisfies Eq. (40) and (41), then the same W^* would also be the global optimal for Eq. (35).

Our goal is to use Eq. (35) as a surrogate to understand the conditions that guarantees the existence of W^*, thereby satisfying Eq. (40) and (41). After, multiplying the square terms out, Eq. (39) becomes

$$\arg\min_W \quad \frac{1}{\sigma^2}\sum_{i,j\in\mathcal{S}}(r_i-r_j)^T WW^T(r_i-r_j) - \frac{1}{\sigma^2}\sum_{i,j\in\mathcal{S}^c}(r_i-r_j)^T WW^T(r_i-r_j) \quad \text{s.t}: \quad W^TW=I. \tag{42}$$

Since $(r_i-r_j)^T WW^T(r_i-r_j)$ is a scalar value, we can place a Trace around the term and rotate them to obtain

$$\arg\min_W \quad \frac{1}{\sigma^2}\sum_{i,j\in\mathcal{S}}\text{Tr}[W^T(r_i-r_j)(r_i-r_j)^TW] - \frac{1}{\sigma^2}\sum_{i,j\in\mathcal{S}^c}\text{Tr}[W^T(r_i-r_j)(r_i-r_j)^TW] \quad \text{s.t}: \quad W^TW=I. \tag{43}$$

At this point, we can simply move the $\frac{1}{\sigma^2}$ value and the summation operation into the Trace to obtain

$$\arg\min_W \quad \text{Tr}(W^TA_=W) - \text{Tr}(W^TA_{\neq}W) \quad \text{s.t}: \quad W^TW=I. \tag{44}$$

□

Lemma 4. *There exist a set of weights $W_1, ..., W_L$ such that the CCN objective generates a uniformly converging risk sequence*

Proof. Looking at the 1st term of Eq. (43) more carefully and focus on the $W^T(r_i-r_j)$ portion where r_i and r_j are assume to belong to the same class. In the most ideal case, we wish find a subspace projection W that pulls all sample pairs together, i.e., we want $0 = W^T(r_i-r_j)$ for all (r_i, r_j) pairs. This is possible only in the case where W is in the null space of $A_=$.

Simultaneously, we want a W that pushes (r_i-r_j) distance infinitely large for the 2nd term. While infinity may not be possible, we know that the samples are maximally separated if W is in the span of the eigenvectors of $A_{\neq}$ that has an associated positive eigenvalue. We refer to this set of eigenvectors as V.

At this point, we can mentally draw the 2 spaces as 2 circles in a Venn diagram. One circle is the null space of $A_=$ while the other circle is the span of V. Therefore, the intersection of these 2 spaces would achieve both objectives simultaneously. Since the 1st term is pushed to 0, as long as the 2nd term is $\neq 0$, we can set σ to be very small to push the 2nd term toward infinity. Therefore, the key to the existence of W^* is the existence of this intersection. Alternatively, we can state that if the null space of $A_=$ is sufficiently large such that it covers the span of V, then W^* exists.

Here, we leverage the feature map of a Gaussian kernel in that it has $A_= \in \mathbb{R}^{\infty\times\infty}$. Therefore, given that $A_=$ has a finite rank, its null space must have infinite dimensions. The only possibility for the null space of $A_=$ not to intersect with the span of V, is if $A_= = A_{\neq}$. Since $A_=$ and $A_{\neq}$ are by definition different, the null space of $A_=$ from a Gaussian kernel will always be sufficiently large to intersect the span of V, i.e., the usage of a Gaussian kernel guarantees the existence of W^* for both Eq. (44) and (34).

Given the existence of W^* in the solution space of $\mathcal{W}$, we define its neighborhood of radius r as

$$B_r(W^*) = \{W \in \mathcal{W} : d(W^*, W) < r\}. \tag{45}$$

Given $B_r(W^*)$, there must also exist a r and a set of Ws, $\{W_1, ..., W^*\}$, sampled from interpolation points between a point within $B_r(W^*)$ and W^* that monotonically improves the objective. In addition, since the Gaussian kernel is a Universal kernel, all of its functions in RKHS are continuous, therefore, the monotonic convergence is also uniform. □

We caution that proving the existence of a solution doesn't imply that the solution can be obtained easily or within a reasonable time. We have found experimentally that ISM is a fast algorithm that maintains this inequality with a uniform convergence.

Aria's attempt:

Define $\mathcal{E}_l = \{W^* = W_1, \ldots, W_l | W^* = \arg\min_{W} \quad \mathrm{Tr}(W^T A_= W) - \mathrm{Tr}(W^T A_{\neq} W) \quad \text{s.t}: \quad W^T W = I\}$ to be the set of all weights that achieves optimal solution till depth l.

Claim: The set $\mathcal{E}_l \subseteq (\mathcal{E}_{l+1})_l$.

If the claim is proven, the optimal solution at depth l is consist of solutions at depth $l-1$, thus the optimal solution of depth l is lower or equal than the optimal solution at depth $l-1$, meaning:

$$(\min \quad \mathrm{Tr}(W^T A_= W) - \mathrm{Tr}(W^T A_{\neq} W))_l \leq (\min \quad \mathrm{Tr}(W^T A_= W) - \mathrm{Tr}(W^T A_{\neq} W))_{l-1} \tag{46}$$

Applying Lemma 2, we can conclude the following:

$$\max_{f_l} \quad ||\mathrm{Cov}(f_{l^\circ}(X), Y)||_F \geq ||\mathrm{Cov}(f^*_{(l-1)^\circ}(X), Y)||_F. \tag{47}$$

Hence the optimal W for each depth induces a monotonically decreasing risk sequence. Furthermore the risk sequence is bounded from bellow by zero, thus the risk sequences is converging to its infimum.

Proof of the claim:

$\mathcal{E}_l \subseteq (\mathcal{E}_{l+1})_l$:

For proving the claim we prove the following:

$$W^* \in \mathcal{E}_l \rightarrow W^* \in \mathcal{E}_{l+1}$$

Assume $W^* \in \mathcal{E}_l$. Matrix $A_=, A_{\neq}$ is positive semi definite, hence, $\mathrm{Tr}(W^T A_= W), \mathrm{Tr}(W^T A_{\neq} W)$ is always a positive number, thus if the optimal solution W^* exists in depth l should satisfies that $\mathrm{Tr}(W^T A_= W) = 0$ and $\mathrm{Tr}(W^T A_{\neq} W) \neq 0$.

Studying the behavior of $\mathrm{Tr}(W^T A_= W) = 0$:

Again because $A_=$ is positive semi definite, $\mathrm{Tr}(W^T A_= W)$ is only zero, if $W^T A_= W = \vec{0}$,

$$W^T A_= W = W^T \frac{1}{\sigma^2} \sum_{i,j \in \mathcal{S}} (r_i - r_j)(r_i - r_j)^T W = \frac{1}{\sigma^2} \sum_{i,j \in \mathcal{S}} W^T (r_i - r_j)(r_i - r_j)^T W \tag{48}$$

because every elements inside the summation $(r_i - r_j)(r_i - r_j)^T$ is positive semi definite matrix, hence, $W^T(r_i - r_j)(r_i - r_j)^T W = \vec{0}, i, j \in \mathcal{S}$. Matrix W can be seen through its columns $W = [W_1, \ldots, W_q]$, thus $W^T(r_i - r_j)(r_i - r_j)^T W = \vec{0}$ becomes:

$$\begin{aligned} &W_1^T (r_i - r_j)(r_i - r_j)^T W_1 = 0 = ||(r_i - r_j)^T W_1||_2 = 0 \rightarrow (r_i - r_j)^T W_1 = 0 \rightarrow r_i^T W_1 = r_j^T W_1 \\ &. \\ &. \\ &. \\ &W_q^T (r_i - r_j)(r_i - r_j)^T W_q = 0 = ||(r_i - r_j)^T W_q||_2 = 0 \rightarrow (r_i - r_j)^T W_q = 0 \rightarrow r_i^T W_q = r_j^T W_q \end{aligned} \tag{49}$$

Which means the in IDs space r_i and r_j get mapped to a same point through W linear transformation for $i, j \in \mathcal{S}$.

Assuming Gaussian feature map ϕ, for $x_i = x_j$, after the nonlinear mapping, they mapped to the same point is RKHS space, meaning $\phi(x_i) = \phi(x_j)$, and hence for the next depth $l+1$ adding any linear W will still keep i, j points on top of each other. Hence the We concluded that, if W^* till layer l satisfies $\mathrm{Tr}(W^T A_= W) = 0$, adding any W at layer $l+1$ will still keep this part of the objective the same.

Studying the behavior of $\mathrm{Tr}(W^T A_{\neq} W) >> 0$:

With the same argument, we can use the fact that $A_{\neq}$ is semi positive definite, hence, $\mathrm{Tr}(W^T A_{\neq} W) \neq 0$ is equivalent to $W^T A_{\neq} W \neq \vec{0}$, which means To be continued ...

we know that ϕ is injective and hence based on definition of injectivity, we have:

$$x_i \neq x_j \rightarrow \phi(x_i) \neq \phi(x_j)$$

which means if at layer l, the optimal W^* does not map i, j samples, after applying ϕ, they are not going to map to each other, so at layer $l+1$ the samples not in the same class have distance, hence there exist a projection down keep this distances positive, **if point are distinct, is there a projection that keep their distance positive, I need to think to prove it but it makes sence, because if there, we don't really need it becuase of the following**, There exist a W at step $l+1$, which still keeps them apart and that is W that all is zero and the second column is all one, due to the gaussian feature map, second the features map in second dimension in RKHS is invertable meaning, if $x_i \neq x_j$ then that elements is not equal as well, hence projection through that dimension keeps their distance. Hence choosing this W for $l+1$, have all the criterias for optimal solution in $\mathcal{E}_{l+1}$.

Hence every solution in $\mathcal{E}_l$ is inside the solution of $\mathcal{E}_{l+1}$.

End of proof.

Proving there exist a certain projection in Gaussian feature map that is injetive:

Pick W to be projection of the first two dimension of Gaussian kernel. **Claim:** $\phi \circ W$ **is injective**

proof

$$\begin{aligned} X_1 \exp X_1^2 &= X_2 \exp X_2^2 \\ \exp X_1^2 &= \exp X_2^2 \\ \rightarrow X_1 &= X_2 \end{aligned} \tag{50}$$

Proof of global monotonically non-decreasing:

Claim: *There exists W_l, where $\mathcal{E}_{l-1} = \mathcal{E}_l$.*

If the claim is proven, the optimal solution at depth l is consist of solutions at depth $l-1$, thus the optimal solution of depth l is lower or equal than the optimal solution at depth $l-1$, meaning:

$$(\min \quad \mathrm{Tr}(W^T A_{=} W) - \mathrm{Tr}(W^T A_{\neq} W))_l \leq (\min \quad \mathrm{Tr}(W^T A_{=} W) - \mathrm{Tr}(W^T A_{\neq} W))_{l-1} \tag{51}$$

Applying Lemma 2, we can conclude the following:

$$\max_{f_l} \quad ||\mathrm{Cov}(f_{l^\circ}(X), Y)||_F \geq ||\mathrm{Cov}(f^*_{(l-1)^\circ}(X), Y)||_F. \tag{52}$$

Hence the optimal W for each depth induces a monotonically decreasing risk sequence. Furthermore the risk sequence is bounded from bellow by zero, thus the risk sequences is converging to its infimum.

Proof of the claim for the global case:

For proving the claim we prove the following:

$$\exists W^*, \, \mathcal{E}_l(W_1, \ldots, W_{l-1}, W^*) = \mathcal{E}_{l-1}(W_1, \ldots, W_{l-1})$$

Assume $W^* \in \mathcal{E}_l$. Matrix $A_{=}, A_{\neq}$ is positive semi definite, hence, $\mathrm{Tr}(W^T A_{=} W), \mathrm{Tr}(W^T A_{\neq} W)$ is always a positive number, thus if the optimal solution W^* exists in depth l should satisfies that $\mathrm{Tr}(W^T A_{=} W) = 0$ and $\mathrm{Tr}(W^T A_{\neq} W) = \infty$.

Studying the behavior of $\mathrm{Tr}(W^T A_{=} W) = 0$: Again because $A_{=}$ is positive semi definite, $\mathrm{Tr}(W^T A_{=} W)$ is only zero, if $W^T A_{=} W = \vec{0}$,

$$W^T A_{=} W = W^T \frac{1}{\sigma^2} \sum_{i,j \in \mathcal{S}} (r_i - r_j)(r_i - r_j)^T W = \frac{1}{\sigma^2} \sum_{i,j \in \mathcal{S}} W^T (r_i - r_j)(r_i - r_j)^T W \tag{53}$$

because every elements inside the summation $(r_i - r_j)(r_i - r_j)^T$ is positive semi definite matrix, hence, $W^T(r_i - r_j)(r_i - r_j)^T W = \vec{0}, i, j \in \mathcal{S}$. Matrix W can be seen through its columns $W = [W_1, \ldots, W_q]$, thus $W^T(r_i - r_j)(r_i - r_j)^T W = \vec{0}$ becomes:

$$\begin{aligned}
&W_1^T (r_i - r_j)(r_i - r_j)^T W_1 = 0 \to W_1^T (r_i - r_j)(r_i - r_j)^T W_1 W_1^T = 0 \to W_1^T A_{=_{ij}} = 0 \to A_{=_{ij}}^T W_1 = A_{=_{ij}} W_1 = 0. \\
&. \\
&. \\
&. \\
&W_q^T (r_i - r_j)(r_i - r_j)^T W_q = 0 = ||(r_i - r_j)^T W_q||_2 = 0 \to (r_i - r_j)^T W_q = 0 \to r_i^T W_q = r_j^T W_q
\end{aligned} \tag{54}$$

Which means the in IDs space r_i and r_j get mapped to a same point through W linear transformation for $i, j \in \mathcal{S}$.

Assuming Gaussian feature map ϕ, for $x_i = x_j$, after the nonlinear mapping, they mapped to the same point is RKHS space, meaning $\phi(x_i) = \phi(x_j)$, and hence for the next depth $l+1$ adding any linear W will still keep i, j points on top of each other. Hence the We concluded that, if W^* till layer l satisfies $\mathrm{Tr}(W^T A_{=} W) = 0$, adding any W at layer $l+1$ will still keep this part of the objective the same.

Studying the behavior of $\mathrm{Tr}(W^T A_{\neq} W) >> 0$:

With the same argument, we can use the fact that $A_{\neq}$ is semi positive definite, hence, $\mathrm{Tr}(W^T A_{\neq} W) \neq 0$ is equivalent to $W^T A_{\neq} W \neq \vec{0}$, We can continue the same arguments and prove the following:

$$W_q^T (r_i - r_j)(r_i - r_j)^T W_q >> 0 \to A_{\neq_{ij}} W_q >> 0 \tag{55}$$

which corresponds to the largest eigenvalue.

we know that ϕ is injective and hence based on definition of injectivity, we have:

$$x_i \neq x_j \rightarrow \phi(x_i) \neq \phi(x_j)$$

which means if at layer l, the optimal W^* does not map i, j samples, after applying ϕ, they are not going to map to each other, so at layer $l+1$ the samples not in the same class have distance, hence there exist a projection down keep this distances positive, There exist a W at step l, which still keeps them apart and that is $W^* = [I_q 0]$, due to the gaussian feature map, as long as $q > 1$ we have a injective map, if $x_i \neq x_j$ then $\phi(x_i) \neq \phi(x_j)$, hence projection through that dimension keeps them distinct meaning in RKHS space composition of function till layer l and after layer l are geometrically equivalent, hence based on the corollary and theorem when $\sigma \rightarrow \infty$, we have $\mathcal{E}_l = \mathcal{E}_{l-1}$

End of proof.

Proving there exist a certain projection in Gaussian feature map that is injetive:

Pick W to be projection of the first two dimension of Gaussian kernel. **Claim:** $\phi \circ W$ **is injective**

proof:

For $q > 1$, we have the first two elements, hence:

$$\begin{aligned} X_1 \exp X_1^2 &= X_2 \exp X_2^2 \\ \exp X_1^2 &= \exp X_2^2 \\ \rightarrow X_1 &= X_2 \end{aligned} \tag{56}$$

claim: The same proof works for local solution

Picking $W_l = [I_q, 0]$, would makes $\phi \circ W_l$ injective. Hence if two points $x_i^{l-1} \neq x_j^{l-1}$ then $\phi \circ W_l(x_i^{l-1}) = x_i^l \neq \phi \circ W_l(x_j^{l-1}) = x_j^l$. Therefore the points samples that $x_i^{l-1} = x_j^{l-1}$, we would have $\phi \circ W_l(x_i^{l-1}) = \phi \circ W_l(x_j^{l-1})$, and the point that are different would still be different. Hence

$$\forall x_i^{l-1} \neq x_j^{l-1} \rightarrow r_i \neq r_j \rightarrow ||W^T r_i - W^T r_j|| > 0 \rightarrow \lim_{\sigma \rightarrow 0} e^{-\frac{||W^T r_i - W^T r_j||^2}{2\sigma^2}} = 1 \tag{57}$$

$$\forall x_i^{l-1} = x_j^{l-1} \rightarrow r_i = r_j \rightarrow ||W^T r_i - W^T r_j|| = 0 \rightarrow \lim_{\sigma \rightarrow 0} e^{-\frac{||W^T r_i - W^T r_j||^2}{2\sigma^2}} = 0. \tag{58}$$

Definition 2. *Function $f, g : \mathbb{R}^m \to \mathbb{R}^m$ are geometrically equivalent iff:*

$$\forall x, y \in \mathbb{R}^m, d(f(x), f(y)) > 0 \Leftrightarrow d(g(x), g(y)) > 0 \tag{59}$$

We show this, $f \approx_{GE} g$

Theorem 5. *For function $W : \mathbb{R}^m \to \mathbb{R}^m$, if W is injective:*

$$W \circ f \approx_{GE} f \approx_{GE} f \circ W \tag{60}$$

Proof.

$$0 \sum_{i,j} \Gamma_{i,j} e^{-\frac{||W^T r_i - W^T r_j||^2}{2\sigma^2}} \quad s.t \quad W^T W = I. \tag{61}$$

□

Theorem 6. *If $f \approx_{GE} g$, $\forall \epsilon > 0$, $\exists \sigma_0$ which for $\forall \sigma \leq \sigma_0$ then:*

$$|||cov(f(x), Y)||_F - ||cov(g(x), Y)||_G| \leq \epsilon \tag{62}$$

Corollary 1. *For injecive W_l, $\exists \sigma_0$ which for $\forall \sigma \leq \sigma_0$ then:*

$$\epsilon_l \geq \epsilon_{l-1}$$

Theorem 7. *Assume if: $|S^c| \geq 3$, then, $\forall i, j \in S$*

$$Null(A_{ij}) \cap Span(A_{\neq}) \neq 0$$

Proof. **Claim:** $rank(A_{\neq}) > 1$:

We prove this through contradiction. Assume $rank(A_{\neq}) \leq 1$, so its either zero, or one.

- Case $rank(A_{\neq}) = 0$. We know that $A_{\neq} \geq 0$ hence, if the rank is zero, then $\forall W \in \mathbb{R}^m$:

$$\begin{aligned} A_{\neq} W = \sum_i \phi_i \phi_i^T W \to \\ W^T A_{\neq} W = \sum_i W^T \phi_i \phi_i^T W \to W^T \phi_i \phi_i^T W = 0 \forall i \\ \to W = \phi_i \phi_i^T \phi_i \phi_i^T \phi_i = |\phi_i|^2 = 0 \to \phi_i = 0 \forall i \end{aligned} \tag{63}$$

  which is a contradiction because that means:

$$\forall i, j \in S^c, r_i = r_j$$

  , which means everything are just one point.

- Case $rank(A_{\neq}) = 1$.

  Lemma 5. *For $A = \sum_i^n \phi_i \phi_i^T$, we have:*

$$span(A) = span(\{\phi_1, \ldots, \phi_n\})$$

Proof. For any $W \in \mathbb{R}^m$ we have the following:

$$A_{\neq} W = \sum_i \phi_i \phi_i^T W = \sum_i c_i \phi_i \tag{64}$$

where $c_i = \phi_i^T W$ □

Claim: Now Assuming $|S^c| \geq 3$, we have contradiction.

assume the case $|S^c| = 3$Choose three separate points from S^c, r_1, r_2, r_3 on the positive quarter of unit ball, without loss of generality assume choose r_1 to be the one with largest angle and r_3 with smallest. From previous lemma we know that span of $A_{\neq}$ is equal to $span\{(r_1 - r_2), (r_2 - r_3), (r_1 - r_3)\}$. having the condition that the rank is equal to one means that any vector $\{(r_1 - r_2), (r_2 - r_3), (r_1 - r_3)\}$ can be represented as $\{(r_1 - r_3)\}$. In particular we have the following:

$$\begin{aligned} r_1 - r_2 &= \lambda_1 (r_1 - r_3) \\ r_2 - r_3 &= \lambda_2 (r_1 - r_3) \end{aligned} \tag{65}$$

Hence $r_j = r_3(1 - \lambda_2) + \lambda_2 r_1$, because of the assumption that r_2 is between r_1 and r_3, hence we can assume $0 < \lambda_2 < 1$. It is strict inequalities because the points are separate. Now we have the followings:

$$\begin{aligned} r_2^T r_2 &= (r_3(1 - \lambda_2) + \lambda_2 r_1)^T (r_3(1 - \lambda_2) + \lambda_2 r_1) \\ &= 1 + 2\lambda_2^2 - 2\lambda_2 + 2\lambda_2(1 - \lambda_2) r_3^T r_1 \\ &< 1 + 2\lambda_2^2 - 2\lambda_2 + 2\lambda_2(1 - \lambda_2) = 1 \end{aligned} \tag{66}$$

which we used the fact that $r_3^T r_1 < 1$ using the fact $r_1 \neq r_3$. And $0 < \lambda_2 < 1$. which is a contradiction due to the fact that the norm of r_2 is equal to one.

Hence this proves the claim that $rank(A_{\neq}) > 1$.

Lemma 6. *If $rank(A_{\neq}) > 1$ then, $\forall i, j \in S$:*

$$Null(A_{ij}) \cap Span(A_{\neq}) \neq \{0\}$$

Proof. For $A_{ij} = (r_i - r_j)(r_i - r_j)^T$, hence it has a rank one, call the vector $r_i - r_j = \phi$. we have two cases

- $\phi \in span(A_{\neq})$: but the $span(A_{\neq})$ has more than element, hence other elements should be $null(A_{ij})$, hence $Null(A_{ij}) \cap Span(A_{\neq}) \neq \{0\}$.
- $\phi \in null(A_{\neq})$, but the elements of $span(A_{\neq})$ should be in $null(A_{ij})$, hence $Null(A_{ij}) \cap Span(A_{\neq}) \neq \{0\}$.

So in either case the intersection is not empty.

□

This proves that for any elements $i, j \in S$, the intersection $Null(A_{ij}) \cap Span(A_{\neq}) \neq \{0\}$ has a non trivial element inside of it.

□

Theorem 8. *For $|S^c| \geq 3$ then,$\exists W$, $\exists \sigma_0$ where $\forall \sigma \leq \sigma_0$ we have:*

$$\epsilon_l > \epsilon_{l-1}$$

Proof. So □

Corollary 2. *The global optimal solution can be reach in finite time solving greedy steps.*

Aria's:

Theorem 9. *For all $\sigma \leq \sigma_0$, $\exists W_l$*

$$\underbrace{\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}}_{\textit{1st term}} - \underbrace{\sum_{i,j\in\mathcal{S}^c} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}}_{\textit{2nd term}} \geq \sum_{i,j\in\mathcal{S}\cup\mathcal{S}^c} \Gamma_{i,j} r_i^T r_j \tag{67}$$

Proof. Note that if any term from the last layer are mapped together, meaning $r_i = r_j$, then the LHS and RHS would be equal, so without loss of generality we can assume all the samples here are no equal, meaning $r_i \neq r_j$. For this proof assume we have two classes denoted as $\mathcal{S}^1, \mathcal{S}^2$.

Lemma 7. *For any $1 > \mathcal{E} > 0$,$\exists \sigma_0$ that:*

$$\forall i, j \quad r_i^T r_j \leq \mathcal{E}$$

Proof. For previous kernel property, we know that:

$$< r_i, r_j >= e^{\frac{-|.|^2}{2\sigma^2}}$$

where $|.|$ is the distance before projecting to the RKHS space. We have finitely many of samples, hence there exists a maximum distance $|.|$ of them. Hence by choosing $\sigma_0 = \sqrt{\frac{-|.|}{2\ln \mathcal{E}}}$, we conclude the proof of the lemma.

□

Hence based on the lemma the upper bound of RHS is:

$$\sum_{i,j\in\mathcal{S}\cup\mathcal{S}^c} \Gamma_{i,j} r_i^T r_j \leq \mathcal{E} \sum_{i,j\in\mathcal{S}} \Gamma_{i,j} - \sum_{i,j\in\cup\mathcal{S}^c} \Gamma_{i,j} \leq \mathcal{E} \sum_{i,j\in\mathcal{S}} \Gamma_{i,j} \tag{68}$$

Hence we need to prove the following:

$$\underbrace{\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}}_{\text{1st term}} - \underbrace{\sum_{i,j\in\mathcal{S}^c} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}}_{\text{2nd term}} \geq \mathcal{E} \sum_{i,j\in\mathcal{S}} \Gamma_{i,j} \tag{69}$$

First term:

By decreasing the σ_0, the $< r_i, r_j >$ approaches to zero, meaning they become almost orthogonal.

hence there exist a finite orthogonal axis $\{e_1, \ldots, e_n\}$, where n is the number of samples. and we have the following property:

Lemma 8.

$$\forall i, j < e_i, r_i >\geq 1 - \mathcal{E}, < e_i, r_j >\leq \mathcal{E}. \tag{70}$$

$$\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}} = \sum_{i,j\in\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}} + \sum_{i,j\in\mathcal{S}^2} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}} \tag{71}$$

pick $W = \sum_{i\in\mathcal{S}^1} e_i$. Hence we have:

$$\lim_{\mathcal{E}\longrightarrow 0} \sum_{i,j\in\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}} = \sum_{i,j\in\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{2}{2\sigma^2}} \tag{72}$$

and for $\mathcal{S}^2$:

$$\lim_{\mathcal{E}\longrightarrow 0}\sum_{i,j\in\mathcal{S}^2}\Gamma_{i,j}e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}=\sum_{i,j\in\mathcal{S}^2}\Gamma_{i,j}e^{-\frac{0}{2\sigma^2}}=\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j} \tag{73}$$

For $\mathcal{S}^c$

$$\lim_{\mathcal{E}\longrightarrow 0}\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}=\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{1}{2\sigma^2}} \tag{74}$$

Lemma 9. *We have the following:*

$$\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}e^{-\frac{2}{2\sigma^2}}+\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}-\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{1}{2\sigma^2}}\geq\mathcal{E}\sum_{i,j\in\mathcal{S}}\Gamma_{i,j} \tag{75}$$

Proof. Taking things to one side would gives us the following:

$$\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}\geq\mathcal{E}\sum_{i,j\in\mathcal{S}}\Gamma_{i,j}-\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}e^{-\frac{2}{2\sigma^2}}+\sum_{i,j\in\mathcal{S}^1}+\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{1}{2\sigma^2}} \tag{76}$$

The LHS is a positive number, and the RHS can be controlled to be arbitrary close to zero by making σ of this layer and $\mathcal{E}$ small. □

Lemma 10.

$$\begin{aligned}\lim_{\mathcal{E}\longrightarrow 0}(\sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}+\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}}+\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{(r_i-r_j)^T W_l W_l^T (r_i-r_j)}{2\sigma^2}})=\\ \sum_{i,j\in\mathcal{S}^1}\Gamma_{i,j}e^{-\frac{2}{2\sigma^2}}+\sum_{i,j\in\mathcal{S}^2}\Gamma_{i,j}-\sum_{i,j\in\mathcal{S}^c}\Gamma_{i,j}e^{-\frac{1}{2\sigma^2}}\end{aligned} \tag{77}$$

because every part of the RHS is continuous function of $\mathcal{E}$, hence we can get arbitrary close to the LHS.

Lets call the LHS of lemma 4 as f_1, ad RHS of it as l_1, and the RHS of lemma 3 as l_2.

We want to prove that $f_1 \geq l_2$. From lemma 3 we prove that we can make $f_1 - l_2 > \delta > 0$, in fact we make it as large as $\sum_{i,j\in\mathcal{S}^2}\Gamma_{i,j}$. From lemma 4 we know that we can make $|f_1 - l_1| < \delta/3$. hence we have $l_1 - l_2 > 2 * \delta/3 > 0$. Hence the proof is compelte.

□

G. Proof for Property 4

Proof. Given $\Gamma = HYY^TH$ and assuming that we use the traditional Gaussian kernel for Ψ, then at each layer, the objective becomes

$$\max_{W} \quad \sum_{i,j} \Gamma_{i,j} e^{-\gamma(W^T x_i - W^T x_j)^2} \quad \text{s.t} \;\; W^T W = I. \tag{78}$$

By applying Lemma 2, the HSIC objective is maximized when $(W^T x_i - W^T x_j) = 0$ for all (x_i, x_j) samples of the same class, and $(W^T x_i - W^T x_j)$ is maximized for pairs in different classes. Notice that these results come after applying the input by W, therefore, these observations are happening in the IDS. □

H. Proof for Property 5

Proof. We first note that Eq. (78) can also be written as

$$\max_{W} \quad \sum_{i,j} \Gamma_{i,j} \langle \psi(W^T x_i), \psi(W^T x_j) \rangle, \quad \text{s.t} \;\; W^T W = I. \tag{79}$$

Assuming the conditions defined in Lemma 2, HSIC is optimized when the angular distance is pushed towards 0 and $\pi/2$. Therefore, the Gaussian kernel pushes samples to become perfectly aligned or orthogonal on a unit hypersphere in RKHS. □

I. KNet's Relationship to Information Bottleneck

Representing a layer from the HSIC perspective is descriptive of its objective mechanically. Namely, it can be visualized that it is finding the weights W_l to map into the IDS such that when it is mapped back to RKHS, the new representation possesses a maximal dependence on the labels. While this understanding explains the network process mechanically, there also exists a strong relationship to information theory.

Here, let $P(X)$ and $P(Y)$ be the probability distribution for samples for X and Y. From statistics, besides the correlation index, dependence can also be measured by the distance between $P(X)P(Y)$ to $P(X, Y)$, where a distance of 0 implies a complete independence between X and Y. In information theory, this dependence is measured by the Mutual Information (MI) where the distance between $P(X)P(Y)$ to $P(X, Y)$ is measured by KL divergence. Therefore, when the mutual information between X and Y is maximized, the statistical dependence between $P(X)$ and $P(Y)$ is also maximized.

HSIC is related to MI in that it also measures the distances between $P(X)P(Y)$ and $P(X, Y)$. However, instead of using KL divergence, HSIC uses a metric called Maximal Mean discrepancy (MMD). Therefore, when we maximize the HSIC of a layer output to its labels, we are simultaneously discovering a mapping where the distribution of its images is highly dependent on the labels.

By establishing the relationship between HSIC and MI, an MLP using Eq. (3) can be interpreted as an information bottleneck (IB) defined by Tishby et al. [46], i.e., the network compresses the data while filtering out information unrelated to the labels. Here, we provide a short proof of this relationship.

Proof. Tishby et al. [46] defined relevant information in signal $x \in X$ as being the information that this signal provides about another signal $y \in Y$. The information bottleneck concept compresses x into $\hat{x}$ such that $\hat{x}$'s information about y is maximally retained. They measure this minimally sufficient amount of information through the information bottleneck formulation where they maximize

$$\max_{\hat{X}} \quad MI(\hat{X}, Y) - \beta MI(\hat{X}, X). \tag{80}$$

Since HSIC also measure the distance between distributions, by changing KL divergence to MMD, each layer can be defined by the same information bottleneck rewritten into

$$\max_{W} \quad HSIC(R_{l-1}W, Y) - \beta HSIC(R_{l-1}W, R_{l-1}), \tag{81}$$

or

$$\max_{W} \ \operatorname{Tr}(K_{R_{l-1}W} H K_Y H) - \beta \operatorname{Tr}(K_{R_{l-1}W} H K_{R_i} H). \tag{82}$$

Applying the property of trace to sum two terms, it then becomes

$$\max_{W} \ \operatorname{Tr}(K_{R_{l-1}W} H (K_Y - \beta K_{R_i}) H). \tag{83}$$

We next take the Cholesky decomposition and get $\hat{Y}\hat{Y}^T = H(K_Y - \beta K_{R_i})H$, the formulation again reduces down to the proposed HSIC formulation of

$$\max_{W} \ HSIC(R_{l-1}W, \hat{Y}), \tag{84}$$

where the definition of a label $\hat{Y}$ now includes the input information. This suggests that by adjusting the label definition of the HSIC formulation, MLP acts as a information bottleneck if the distances between distributions are measured by MMD.

□

J. Proof for Theorem 3

Theorem 3: *Given a change of basis, the argmax of the HSIC objective is equivalent to the argmin of the CE objective in RKHS.*

J.1. Assumptions and Notations of the Proof

Given n as the number of samples and c as the number of classes, we let $y_i \in \mathbb{R}^{c\times 1}$ be the one-hot encoded label of the ith sample. The output of the neural network parameterized by W after the softmax is denoted as as $\hat{y}_i \in \mathbb{R}^{c\times 1}$ where $\hat{y}_i$ lies on a probability simplex. We also denote $\Gamma_{i,j}$ as scalars where $\Gamma_{i,j}$ is positive if the samples x_i and x_j belong to the same class and $\Gamma_{i,j}$ is negative if x_i and x_j belong to the different classes. When letters i and j are used together, it is used to reference any two samples from the data. In addition, given an output of $\hat{y}_i$, we index each element of the vector via τ, i.e., the τth element of $\hat{y}_i$ is denoted as $\hat{y}_{i,\tau}$. Following these notations, we define the Cross-Entropy objective as

$$\arg\min_{W} - \sum_{i=1}^{n} \sum_{\tau=1}^{c} y_{i,\tau} \log(\hat{y}_{i,\tau}). \tag{85}$$

J.2. Summary the Proof

The proof for **Theorem** 3 is divided into two lemmas.

In Lemma 11, we prove that the cross-entropy is minimized if the following two conditions are satisfied

Condition 1 (Class Orthogonality Condition): For any pairs of $\hat{y}_i$ and $\hat{y}_j$, the inner product between them has the following property.

$$\begin{cases} \langle \hat{y}_i, \hat{y}_j \rangle = 1 & \text{if} \quad i,j \text{ same class} \\ \langle \hat{y}_i, \hat{y}_j \rangle = 0 & \text{if} \quad i,j \text{ not in the same class} \end{cases}. \tag{86}$$

Condition 2 (Bases Alignment Condition): We define a standard basis e_τ as a vector with a value 1 at the τth element and 0 for all other elements. We denote E as a set of all possible standard bases. Then, for any $\hat{y}_i$, it must be an element of E.

In Lemma 12, we show that when HSIC is optimized via KNet, the profile of its output in RKHS for all $\hat{y}_i$ satisfies the *Class Orthogonality Condition*, i.e, they are all unit vectors that are aligned or orthogonal with each other depending on their class designation. However, since its outputs are not aligned along the standard basis, it is not equivalent to Cross-Entropy. Therefore, by adding a final rotation to the bases, the argmax of the HSIC solution can match perfectly to the argmin of CE.

J.3. Lemmas of the Proof

Lemma 11. *If the Class Orthogonality Condition and the Bases Alignment Condition are satisfied then the cross-entropy objective is minimized.*

Proof. The *Class Orthogonality Condition* tells us that the network has discovered a W such that every single sample is either aligned or orthogonal to all other samples. Specifically, if the samples are from the same class, they are all mapped onto the same unit vector. Alternatively, samples from different classes are mapped onto orthogonal vectors. If we further assume to satisfy the *Bases Alignment Condition*, we know that not only are the representations in orthogonal orientations, the bases are perfectly aligned with the standard bases.

Therefore, given S as a set of all $(\hat{y}_i, \hat{y}_j)$ pairs that belong to the same class, $\hat{y}_i = \hat{y}_j$. By appropriately permuting the labels to match the axes, conditions 1 and 2 also implies that $y_i = \hat{y}_i$ for all i, i.e., CE is minimized. □

Lemma 12. *The argmax from solving the HSIC objective using a Gaussian kernel can minimize the Cross-Entropy objective via an additional linear layer.*

Proof. From the proof for Property 5 in Appendix H, we know that by optimizing KNet using the Gaussian kernel with HSIC, the resulting $\hat{y}_i$ satisfies the *Class Orthogonality Condition* for all is. However, since the orthogonal bases produced via HSIC are not aligned along the standard bases, its solution does not satisfy Condition 2.

Fortunately, Condition 2 can be easily satisfied by adding an additional linear transformation to rotate the final layer. Specifically, assuming that after optimization, the HSIC solution maps all samples to c orthogonal bases represented by $\Xi = [\xi_1, \xi_2, ..., \xi_c]$ where the indexing position of ξ_i is assigned based on the ordering of the label. And if we let $\hat{Y} = [\hat{y}_1, \hat{y}_2, ..., \hat{y}_n]^T$ and $Y = [y_1, y_2, ..., y_n]^T$, then

$$Y = \hat{Y}\Xi. \tag{87}$$

Since $\hat{Y}\Xi$ is equal to the label, by passing this rotated result into a softmax, the Cross-Entropy error becomes minimized at 0. □

K. Proof for Theorem 4

Theorem 4 *The argmin of the MSE objective is satisfied by the argmax of the HSIC objective if the final Ψ is discarded to produce an output in IDS.*

K.1. Assumptions and Notations of the Proof

Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (x_i, x_j) from a dataset X that belongs to the same and different classes respectively. We let c be the number of classes and let $\Gamma_{i,j}$ be a set of scalars derived from the labels following Lemma 2.

We prove this theorem by breaking the proof into three lemmas. Previously in Appendix J, we have denoted $\hat{y}_i$ as the output of a layer after the activation function. We have specifically emphasized that $\hat{y}_i$ resides within RKHS. Here we denote the output of a layer for x_i as $\hat{z}_i$ where the output is the point after the linear transformation W *prior* to the activation function, i.e., $W^T x_i = \hat{z}_i$. Therefore, although we are looking at the same formulation, we now focus on what is happening within IDS as HSIC is being optimized. Following this notation, we denote z_i as the ground truth label for the sample x_i.

K.2. Summary the Proof

To summarize the proof, we first prove in Lemma 13 that the MSE is minimized if and only if

$$\begin{cases} (\hat{z}_i - \hat{z}_j)^2 = 0 & \text{if} \quad i, j \text{ same class} \\ (\hat{z}_i - \hat{z}_j)^2 \neq 0 & \text{if} \quad i, j \text{ not in the same class} \end{cases}.$$

In Lemma 14, we show that that the HSIC objective using a Gaussian kernel is optimized if and only if

$$\begin{cases} (\hat{z}_i - \hat{z}_j)^2 = 0 & \text{if} \quad i,j \text{ same class} \\ (\hat{z}_i - \hat{z}_j)^2 = 2 & \text{if} \quad i,j \text{ not in the same class} \end{cases}.$$

Finally, in Lemma 15, we draw from Lemma 13 and 14 to show since the solutions space using the HSIC objective is a subset within the solution space using MSE, a solution that solves the HSIC objective must also satisfy MSE.

K.3. Lemmas of the Proof

Lemma 13. *The MSE is minimized if and only if*

$$\begin{cases} (\hat{z}_i - \hat{z}_j)^2 = 0 & \textit{if} \quad i,j \textit{ same class} \\ (\hat{z}_i - \hat{z}_j)^2 \neq 0 & \textit{if} \quad i,j \textit{ not in the same class} \end{cases}.$$

Proof. First, we assume that the MSE objective is already optimized, this implies that the output must satisfy $\hat{z}_i = z_i$ for any sample. Since $z_i = z_j$ for all samples of the same class, the condition $\hat{z}_i = \hat{z}_j$ must also be satisfied. As for the 2nd condition of $(\hat{z}_i - \hat{z}_j)^2 \neq 0$ when i, j are not in the same class, this is immediately satisfied by the definition of classification by MSE.

Conversely, we next assume that $(\hat{z}_i - \hat{z}_j)^2 = 0$ if sample pair (x_i, x_j) are in the same class, and $(\hat{z}_i - \hat{z}_j)^2 \neq 0$ if they are in different classes. This implies that all samples are mapped onto c distinct points. Since MSE is a supervised problem, the labels can then be easily matched to one of the c distinct points such that for each sample $\hat{z}_i$ we get $z_i = \hat{z}_i$. From here, it can be easily seen that the MSE objective is minimized to 0.

$$0 = \sum_{i,j \in \mathcal{S}} (z_i - \hat{z}_i)^2 \tag{88}$$

□

Lemma 14. *The HSIC objective is optimized if and only if*

$$\begin{cases} (\hat{z}_i - \hat{z}_j)^2 = 0 & \textit{if} \quad i,j \textit{ same class} \\ (\hat{z}_i - \hat{z}_j)^2 = 2 & \textit{if} \quad i,j \textit{ not in the same class} \end{cases}.$$

Proof. First we assume that the HSIC objective using a Gaussian kernel is already optimized. This yields the optimal parameters for the MLP where

$$\theta^* = \arg\max_{\theta} \quad \sum_{i,j} \Gamma_{i,j} e^{-\gamma ||g_\theta(x_i) - g_\theta(x_j)||^2}. \tag{89}$$

Applying Lemma 2, when $\Gamma_{i,j} > 0$ the objective is maximized when g_θ forces $||g_\theta(x_i) - g_\theta(x_j)||^2$ towards 0. Conversely, when $\Gamma_{i,j} < 0$, g_θ will push $||g_\theta(x_i) - g_\theta(x_j)||$ towards the diameter of the solutions space. Here, by using the Gaussian kernel, we know that the maximum distance between two unit vectors is $\sqrt{2}$. By appropriately setting γ, we can ensure that

$$e^{-2\gamma} \approx 0. \tag{90}$$

Converse, if we assume that $(\hat{z}_i - \hat{z}_j)^2 = 0$ for all (x_i, x_j) sample pairs in the same class and $(\hat{z}_i - \hat{z}_j)^2 = \sqrt{2}$ for samples from different classes, if we plug these conditions into Eq. (89) then

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} = \max_{\theta} \quad \sum_{i,j} \Gamma_{i,j} e^{-\gamma (g_\theta(x_i) - g_\theta(x_j))^2}. \tag{91}$$

Since $\sum_{i,j \in \mathcal{S}} \Gamma_{i,j}$ is the absolute upper bound for Eq. (91), this solution must also be the optimal.

□

Lemma 15. *The the optimal condition of the MSE objective is satisfied if the HSIC objective is optimized.*

Proof. When solving MSE, the labels can be anywhere. As long as different labels are distinguishable from each other, the distance between samples of different classes is not part of the objective.

For HSIC since the distance between samples is part of the objective, the labels must be exactly $\sqrt{2}$ apart at an optimal solution. Therefore, the HSIC solution is a specific case of the MSE solution space.

We let the argmin solution space of MSE be $\mathcal{M}$ and the argmax solution space of HSIC be $\mathcal{H}$, then $\mathcal{H} \subset \mathcal{M}$. Therefore, a solution within $\mathcal{H}$ must also be a solution within $\mathcal{M}$. Which lead us to conclude that the argmax of the HSIC objective must also satisfy the MSE objective. □

L. ISM Algorithm

To define Φ, we must first define Ψ and $\mathcal{L}_\Psi$. If we let $\odot$ be the Hadamard product between matrices, then $\Psi = \Gamma \odot K_{XW}$. Given Ψ, we define D_Ψ and $\mathcal{L}_\Psi$ respectively as the degree matrix and the Laplacian of Ψ where $D_\Psi = \text{Diag}(\Psi 1_n)$ and $\mathcal{L}_\Psi = D_\Psi - \Psi$. Having the key notations defined, the Φ matrix associated with the Gaussian kernel is defined as

$$\Phi = -X^T \mathcal{L}_\Psi X. \tag{92}$$

Since Φ associated with the Gaussian kernel is itself a function of W, an approximation of Φ via the 2nd order Taylor expansion produces an Φ_0 independent of W defined as

$$\Phi_0 = -X^T \mathcal{L}_\Gamma X. \tag{93}$$

Since the Laplacian matrix based on Γ no longer require W, an initial W_0 can be used for Eq. (92) to obtain W_1. By follow this pattern, ISM iteratively use W_i to find W_{i+1} until convergence. While this work focuses on the Gaussian kernel, we include the Φ/Φ_0 equation for other kernels in Tables 2 and 3. The ISM algorithm is also included in Algorithm 2.

For convergence, we can converge when the Frobenius Norm $||W_i - W_{i-1}||_F$ falls below a predefined threshold δ. However, based on the ISM algorithm, they found that in practice, the comparison of the most dominant eigenvalues is faster.

Kernel	Approximation of Φs
Linear	$\Phi_0 = X^T \Gamma X$
Squared	$\Phi_0 = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi_0 = X^T \Gamma X$
Gaussian	$\Phi_0 = -X^T \mathcal{L}_\Gamma X$
Multiquadratic	$\Phi_0 = X^T \mathcal{L}_\Gamma X$

Table 2. Equations for the approximate Φs for the common kernels.

Kernel	Φ Equations
Linear	$\Phi = X^T \Gamma X$
Squared	$\Phi = X^T \mathcal{L}_\Gamma X$
Polynomial	$\Phi = X^T \Psi X \quad , \quad \Psi = \Gamma \odot K_{XW,p-1}$
Gaussian	$\Phi = -X^T \mathcal{L}_\Psi X$, $\Psi = \Gamma \odot K_{XW}$
Multiquadratic	$\Phi = X^T \mathcal{L}_\Psi X$, $\Psi = \Gamma \odot K_{XW}^{(-1)}$

Table 3. Equations for Φs for the common kernels.

Algorithm 2 ISM Algorithm

Input : Data X, kernel, Subspace Dimension q
Output : Projected subspace W
Initialization : Initialize Φ_0 using Table 2.
Set W_0 to $V_{\max}$ of Φ_0.
while $||\Lambda_i - \Lambda_{i-1}||_2 / ||\Lambda_i||_2 < \delta$ **do**
  Compute Φ using Table 3
  Set W_k to $V_{\max}$ of Φ
end

M. Dataset Details

Wine. This dataset has 13 features and 178 samples. The features are continuous and heavily unbalanced in magnitude. During the experiments, the dimension is reduced down to 3 prior to performing supervised or unsupervised tasks. The dataset can be downloaded at https://archive.ics.uci.edu/ml/datasets/wine.

Divorce. This dataset has 54 features and 170 samples. 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 and 1728 samples. 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 and 683 samples. The features are discrete and unbalanced in magnitude. During the experiments, the dimension is reduced down to 2 prior to performing supervised or unsupervised tasks. 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. During the experiments, the dimension is reduced down to 20 prior to performing supervised or unsupervised tasks. This dataset is commonly used for alternative clustering since it can be clustered by the identity or the pose of the individuals. The dataset can be downloaded at https://archive.ics.uci.edu/ml/datasets/CMU+Face+Images.

N. Equations for Evaluation Metrics

For all metrics assume that $X \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times c}$ as the data and the label respectively.

Normalized HSIC. The normalized HSIC can be calculated with

$$\mathbb{H} = \frac{HSIC(X,Y)}{\sqrt{HSIC(X,X)HSIC(Y,Y)}}. \tag{94}$$

Silhouette Score. We used the Silhouette score library from Sklearn with the function https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html. The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is (b - a) / max(a, b). To clarify, b is the distance between a sample and the nearest cluster that the sample is not a part of.

Average Cosine Similarity Ratio (CS). Given $\mathcal{S}$ and $\mathcal{S}^c$ as sets of all pairs of samples of (x_i, x_j) from a dataset X that belongs to the same and different classes respectively. The average cosine similarity ratio is defined as

$$CS = \frac{\sum_{i,j\in\mathcal{S}^c}\langle f(x_i), f(x_j)\rangle}{\sum_{i,j\in\mathcal{S}}\langle f(x_i), f(x_j)\rangle}. \tag{95}$$

Since the inner product between samples not in the same class should be 0, this ratio should approach 0 as the ratio improves.

O. 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. [16] and Ansuini et al. [44].

By reporting the weights of each layer, the number of layers is also reported for every network. Although more work is required to identify the rate of convergence, it can be seen experimentally that convergence is achieved relatively quickly, i.e., the length of the *kernel sequence* is generally below 10.

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
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)

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)			

P. Optimal 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 $Y \in \mathbb{R}^{n \times k}$ be the corresponding one-hot encoded labels where k 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

$$d_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{96}$$

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

$$d_{\mathcal{S}^c} = \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{97}$$

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

$$\max_{\sigma} \quad \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} - \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{98}$$

It turns that 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

$$Q = -gK_Y + \bar{g}(\mathbf{1}_{n \times n} - K_Y). \tag{99}$$

Or Q can be written more compactly as

$$Q = \bar{g}\mathbf{1}_{n \times n} - (g + \bar{g})K_Y. \tag{100}$$

Given Q, Eq. (98) becomes

$$\min_{\sigma} \quad \text{Tr}(K_X Q). \tag{101}$$

Since this is a convex objective, it can be solved with BFGS.

Below in Figure 4, 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. (101), 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

$$\min_{\sigma} \quad \text{Tr}(K_X H K_Y H), \tag{102}$$

by setting $Q = HK_Y H$, we can see how the two formulations are equivalent. 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 equivalent for both objectives. Below in Figure 5, 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.

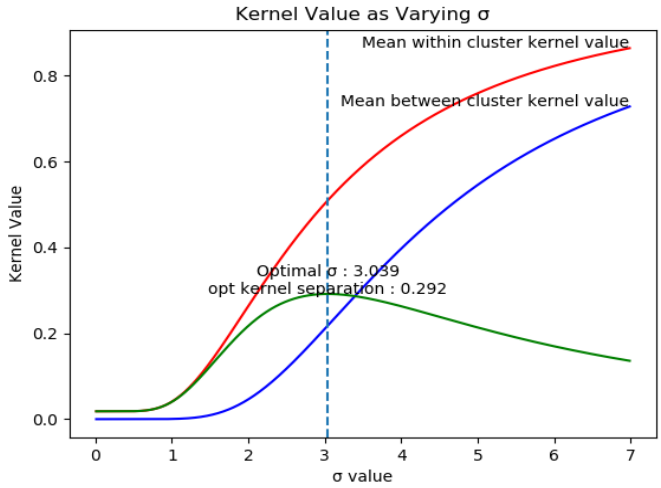

Figure 4. Maximum Kernel separation.

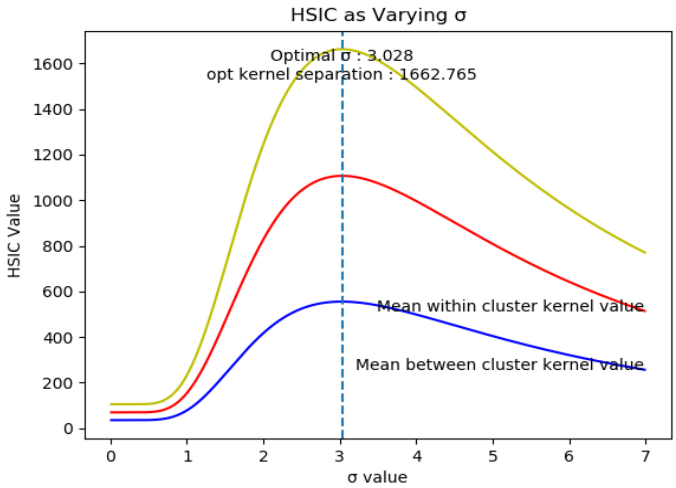

Figure 5. Maximal HSIC.

Q. Graphs of Kernel Sequences

For each dataset, a representative kernel sequence is displayed in the figures below. 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. Ideally, we wish to have a *kernel sequence* to evolve from an uninformative kernel into a highly discriminating kernel of perfect block structures.

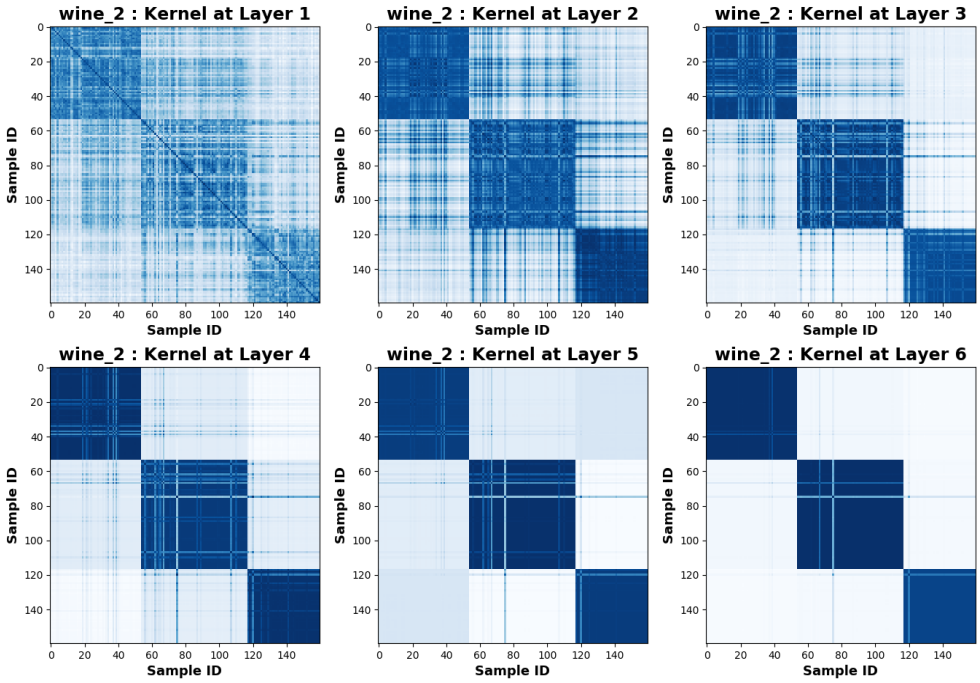

Figure 6. The kernel sequence for the wine dataset.

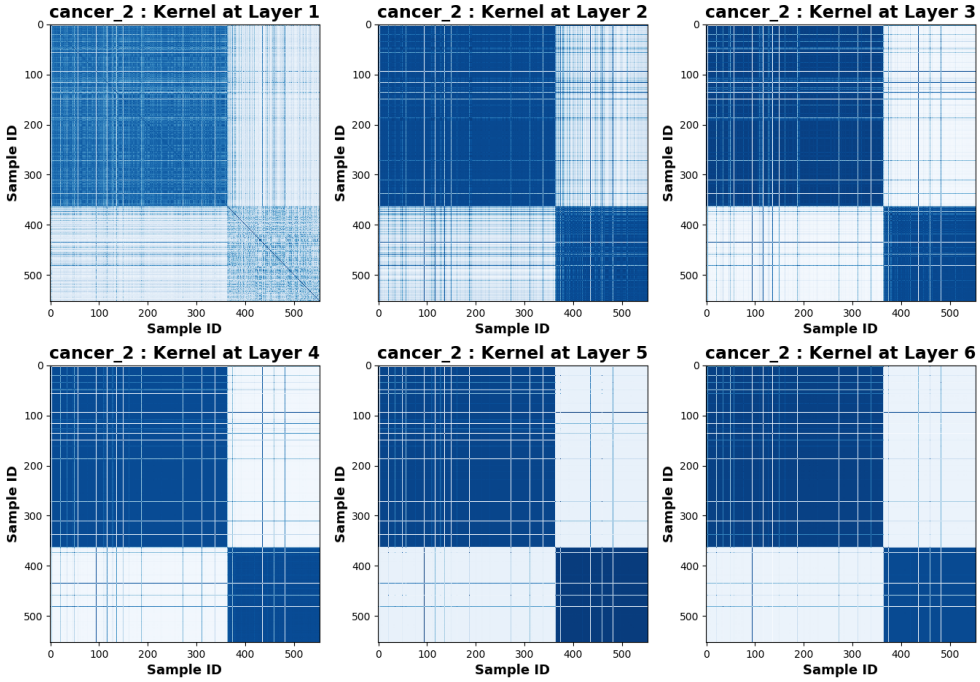

Figure 7. The kernel sequence for the cancer dataset.

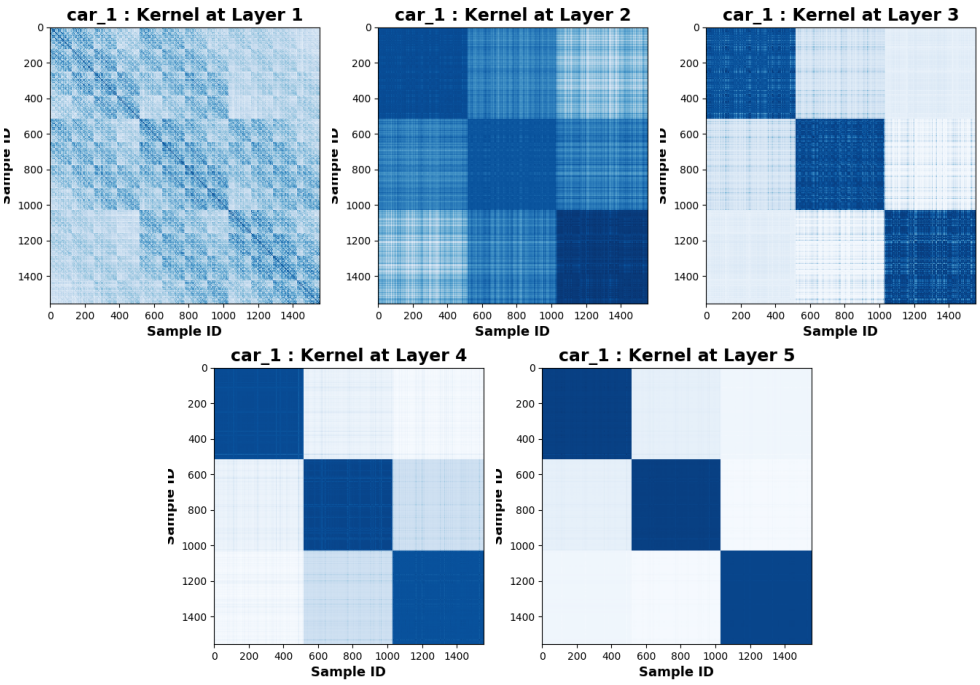

Figure 8. The kernel sequence for the car dataset.

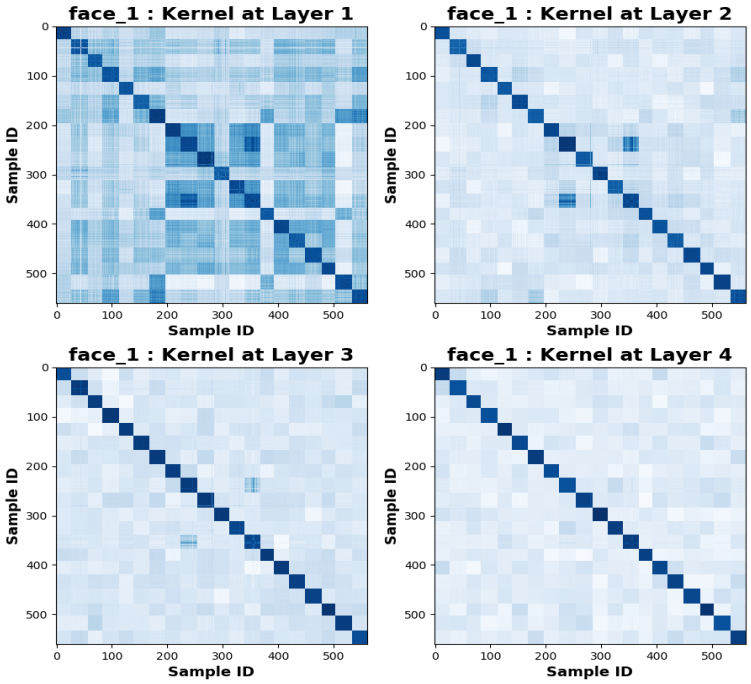

Figure 9. The kernel sequence for the face dataset.

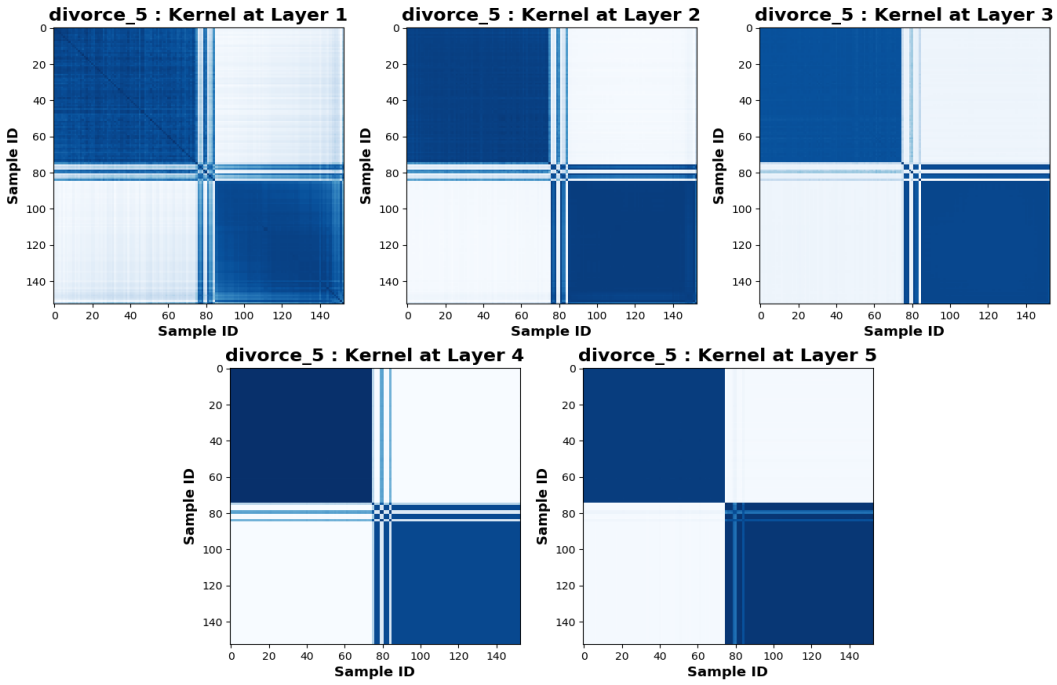

Figure 10. The kernel sequence for the divorce dataset.

R. Evaluation Metrics Graphs

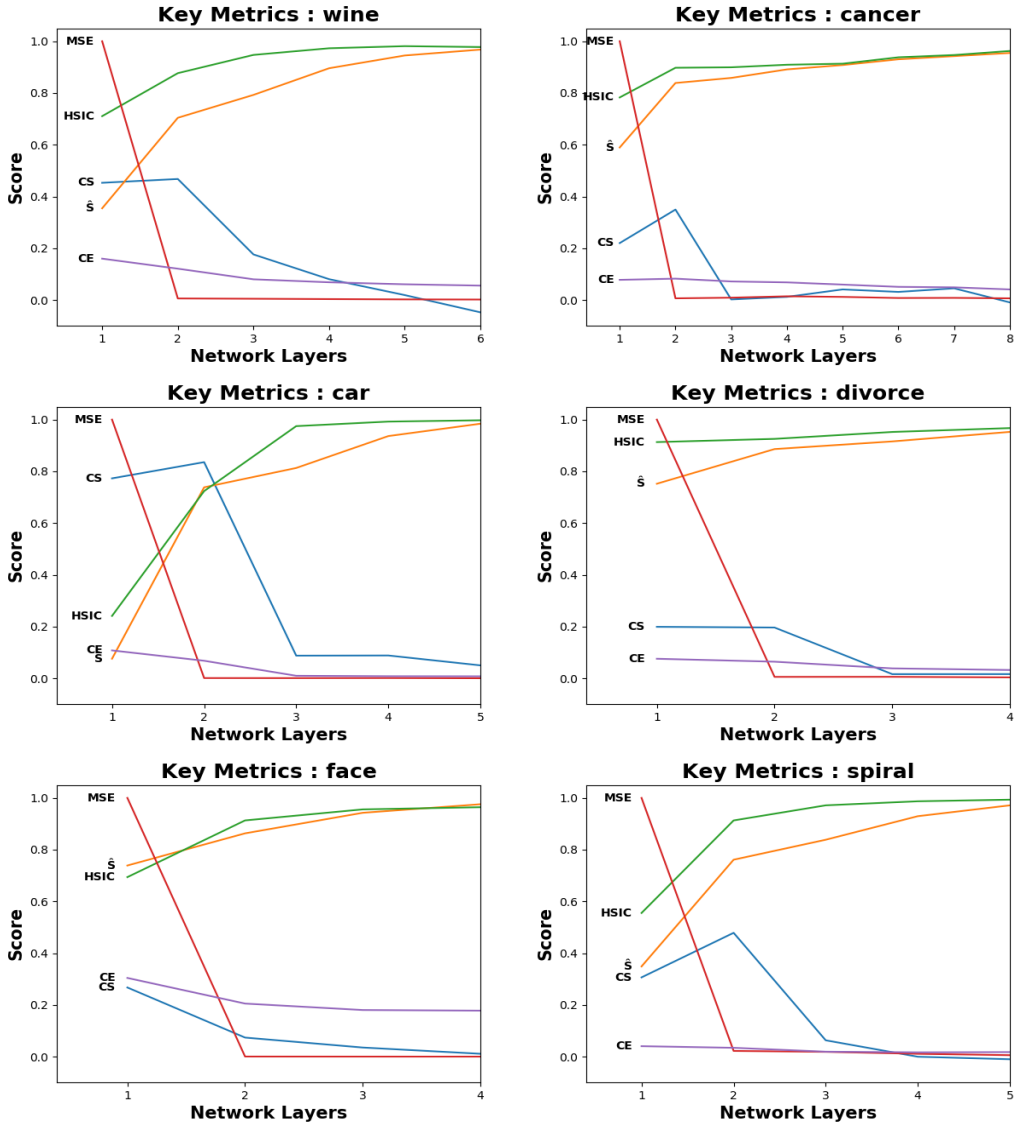

Figure 11. Figures of key metrics for all datasets as samples progress through the network. It is important to notice the uniformly and monotonically increasing HSIC value for each plot since this guarantees a converging kernel/risk sequence. As the Silhouette score approach 1, samples of the same/difference classes in IDS are being pulled into a single point or pushed maximally apart respectively. As CS approach 0, samples of the same/difference classes in RKHS are being pulled into 0 or $\frac{\pi}{2}$ cosine similarity respectively.