Layer-wise Learning of Kernel Dependence Networks

*Chieh Wu[1], *Aria Masoomi[1], Arthur Gretton[2], and Jennifer Dy[1]

[1]*Department of Electrical and Computer Engineering, Northeastern University*
[2]*Gatsby Computational Neuroscience Unit, University College London*

Abstract

We propose a greedy strategy to spectrally train a deep network for multi-class classification. Each layer is defined as a composition of linear weights with the feature map of a Gaussian kernel acting as the activation function. At each layer, the linear weights are learned by maximizing the dependence between the layer output and the labels using the Hilbert Schmidt Independence Criterion (HSIC). By constraining the solution space on the Stiefel Manifold, we demonstrate how our network construct (Kernel Dependence Network or *KNet*) can be solved spectrally while leveraging the eigenvalues to automatically find the width and the depth of the network. We theoretically guarantee the existence of a solution for the global optimum while providing insight into our network's ability to generalize. This workshop paper is only part one of the full paper. For the full paper, see `https://arxiv.org/abs/2006.08539`.

1 Network Model

Let $X \in \mathbb{R}^{n \times d}$ be a dataset of n samples with d features and let $Y \in \mathbb{R}^{n \times \tau}$ be the corresponding one-hot encoded labels with τ number of classes. Let $\mathcal{S}$ be a set of i, j sample pairs that belong to the same class. Its complement, $\mathcal{S}^c$ contains all sample pairs from different classes. Let $\odot$ be the element-wise product. The i^{th} sample and label of the dataset is written as x_i and y_i. H is a centering matrix defined as $H = I_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ where I_n is the identity matrix of size $n \times n$ and $\mathbf{1}_n$ is a vector of 1s also of length n. Given H, we let $\Gamma = HYY^TH$.

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

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

Notice that *KNet* fundamentally models a traditional fully connected multilayer perceptron (MLP) where each layer consists of linear weights W_l and a activation function ψ_l. We propose to solve Eq. (1) greedily; this is equivalent to solving a sequence of *single-layered* networks where the previous network output becomes the current layer's input. At each layer, we find the W_l that maximizes the dependency between the layer output and the label via the Hilbert Schmidt Independence Criterion (HSIC) [1]:

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

*Signifies equal contribution.

While Mean Squared Error (MSE) and Cross-Entropy (CE) have traditionally been used for classification, HSIC is instead chosen because it solves an underlying prerequisite of classification, i.e., learning a mapping for X where similar and different classes become distinguishable. However, since there are many notions of similarity, it is not always clear which is best for a particular situation. *KNet* overcomes this uncertainty by discovering the optimal similarity measure as a kernel function during training. To understand how, first realize that the i, j_{th} element of Γ, denoted as $\Gamma_{i,j}$, is a positive value for samples in $\mathcal{S}$ and negative for $\mathcal{S}^c$. By defining a kernel function $\mathcal{K}$ as a similarity measure between 2 samples, HSIC as Eq. (2) becomes a kernel discovery objective written as

$$\max_{W_l} \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \mathcal{K}_{W_l}(r_i, r_j) - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \mathcal{K}_{W_l}(r_i, r_j) \quad \text{s.t.}\, W_l^T W_l = I. \tag{3}$$

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

While *KNet* uses a MLP structure as a basis, we deviate from a traditional MLP in two respects. First,the traditional concept of activation functions is replace by a feature map of a kernel. For *KNet*, we use the feature map of a Gaussian kernel (GK) to simulate an infinitely wide network. Yet, the kernel trick spares us the direct computation of $\psi(R_{l-1}W_l)\psi^T(R_{l-1}W_l)$; we instead compute the GK matrix given $\mathcal{K}(W_l^T r_i, W_l^T r_j) = \exp\{-||W_l^T r_i - W_l^T r_j||^2/2\sigma^2\}$. Second, the constraint $W^T W = I$ is inspired by the recent work on the geometric landscape of the network solutions [2, 3, 4]. Their work suggests that the network solution can be represented by a linear subspace where the only the direction of the weights matter and not the magnitude. If the solution indeed lives on a linear subspace independent of their scale, we can exploit this prior knowledge to narrow the search space during optimization specific to the Stiefel Manifold where $W_l^T W_l = I$, rendering Eq. (2) solvable spectrally. Consequently, this prior enables us to solve Eq. (2) by leveraging the iterative spectral method (ISM) proposed by Wu et al. [5, 6] to simultaneously avoid SGD and identify the network width. Applying ISM to our model, each layer's weight is initialized using the most dominant eigenvectors of

$$\mathcal{Q}_{l^0} = R_{l-1}^T(\Gamma - \text{Diag}(\Gamma 1_n))R_{l-1}, \tag{4}$$

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

$$\mathcal{Q}_{l^i} = R_{l-1}^T(\hat{\Gamma} - \text{Diag}(\hat{\Gamma} 1_n))R_{l-1}, \tag{5}$$

where $\hat{\Gamma}$ is a function of $W_{l^{\text{i}}}$ computed with $\hat{\Gamma} = \Gamma \odot K_{R_{l-1}W_{l^{\text{i}}}}$. This iterative weight-updating process stops when $\mathcal{Q}_{l^{\text{i}}} \approx \mathcal{Q}_{l^{\text{i}+1}}$, whereupon $\mathcal{Q}_{l^{\text{i}+1}}$ is set to $\mathcal{Q}_l^*$, and its most dominant eigenvectors W_l^* becomes the solution of Eq. (2) where $\partial \mathcal{H}/\partial W_l = 0$.

ISM solves Eq. (2) via the kernel trick directly on an infinitely wide network during training, obtaining W_l^*. Once W_l^* is solved, we approximate ψ with Random Fourier Features (RFF) [7] to finitely simulate an infinitely wide network and acquire the layer output. This output is then used as input for the next layer. Capitalizing on the spectral properties of ISM, the spectrum of $\mathcal{Q}_l^*$ completely determines the the width of the network $W_l^* \in \mathbb{R}^{m \times q}$, i.e., m is equal to the size of the RFF, and q is simply the rank of $\mathcal{Q}_l^*$. Furthermore, since Eq. (2) after normalization is upper bounded by 1, we can stop adding new layers when the HSIC value of the current layer approaches this theoretical bound, thereby prescribing a natural depth of the network. The resulting network ϕ after training will map samples of the same class into its own cluster, allowing the test samples to be classified by matching their network outputs to the nearest cluster center. The source code is included in the supplementary materials and made publicly available at `https://github.com/endsley/kernel_dependence_network`.

2 Theoretical Origin of Kernel Dependence Networks

Background and Notations. Let the composition of the first l layers be $\phi_{l^\circ} = \phi_l \circ ... \circ \phi_1$ where $l \leq L$. This notation enables us to connect the data directly to the layer output where $R_l = \phi_{l^\circ}(X)$.

Since *KNet* is greedy, it solves MLPs by replacing ϕ in Eq. (1) incrementally with a sequence of functions $\{\phi_{l^\circ}\}_{l=1}^{L}$ where each layer relies on the weights of the previous layer. This implies that we are also solving a sequence of empirical risks $\{\mathcal{H}_l\}_{l=1}^{L}$, i.e., different versions of Eq. (1) given the current ϕ_{l°. We refer to $\{\phi_{l^\circ}\}_{l=1}^{L}$ and $\{\mathcal{H}_l\}_{l=1}^{L}$ as the *Kernel Sequence* and the *$\mathcal{H}$-Sequence*.

Following the concept mentioned above, we guarantee the existence of a solution to reach the global optimum given the theorem below with its proof in App. A .

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

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

$$\lim_{L\to\infty} \mathcal{H}_L = \mathcal{H}^*, \tag{6}$$

II. the convergence is strictly monotonic where

$$\mathcal{H}_l > \mathcal{H}_{l-1} \quad \forall l \geq 1. \tag{7}$$

On Generalization. The ISM algorithm provides some insight into generalization. While the HSIC objective employs an infinitely wide network, Ma et al. [8] have experimentally observed that HSIC can generalize even without any regularizer. We ask theoretically, what makes the HSIC objective special? Recently, Poggio et al. [2] have proposed that traditional MLPs generalize because gradient methods implicitly regularize the normalized weights. We discovered a similar impact ISM has on HSIC, i.e., the objective can be reformulated to isolate out n functions $[D_1(W_l), ..., D_n(W_l)]$ that act as a penalty term during optimization. Let $\mathcal{S}_i$ be the set of samples that belongs to the i_{th} class and let $\mathcal{S}_i^c$ be its complement, then each function $D_i(W_l)$ is defined as

$$D_i(W_l) = \frac{1}{\sigma^2} \sum_{j\in\mathcal{S}_i} \Gamma_{i,j}\mathcal{K}_{W_l}(r_i, r_j) - \frac{1}{\sigma^2} \sum_{j\in\mathcal{S}_i^c} |\Gamma_{i,j}|\mathcal{K}_{W_l}(r_i, r_j). \tag{8}$$

Note that $D_i(W_l)$ is simply Eq. (3) for a single sample scaled by $\frac{1}{\sigma^2}$. Therefore, as we identify better solutions for W_l, this leads to an increase and decrease of $\mathcal{K}_{W_l}(r_i, r_j)$ associated with $\mathcal{S}_i$ and $\mathcal{S}_i^c$ in Eq. (8), thereby increasing the size of the penalty term $D_i(W_l)$. To appreciate how $D_i(W_l)$ penalizes $\mathcal{H}$, we propose an equivalent formulation with its derivation in App. B.

Theorem 2. *Eq. (2) is equivalent to*

$$\max_{W_l} \sum_{i,j} \frac{\Gamma_{i,j}}{\sigma^2} e^{-\frac{(r_i-r_j)^T WW^T(r_i-r_j)}{2\sigma^2}} (r_i^T W_l W_l^T r_j) - \sum_i D_i(W_l)||W_l^T r_i||_2. \tag{9}$$

Based on Thm. 2, $D_i(W_l)$ adds a negative cost to the sample norm in IDS, $||W_l^T r_i||_2$, describing how ISM implicitly regularizes HSIC. As a better W_l attempts to improve the objective, it simultaneously imposes a heavier penalty on Eq. (9) where the overall $\mathcal{H}$ may actually decrease.

3 Experiments

Datasets. We confirm the theoretical properties of *KNet* using three synthetic (Random, Adversarial and Spiral) and five popular UCI benchmark datasets: wine, cancer, car, divorce, and face [9]. To test the flexibility of *KNet*, we design the Adversarial dataset to be highly complex, i.e., the samples pairs in $\mathcal{S}^c$ are significantly closer than sample pairs in $\mathcal{S}$. We next designed a Random dataset with completely random labels. All datasets are included along with the source code in the supplementary, and their comprehensive download link and statistics are in App.D.

Evaluation Metrics and Settings. To evaluate the central claim that MLPs can be solved greedily, we report $\mathcal{H}^*$ at convergence along with the training/test accuracy for each dataset. Here, $\mathcal{H}^*$ is normalized to the range between 0 to 1 using the method proposed by Cortes et al. [10]. Since *KNet* at convergence is itself a feature map, we evaluate the network output quality with the Cosine Similarity Ratio (C). The $\langle\phi(x_i), \phi(x_j)\rangle$ for i, j pairs in $\mathcal{S}$ and $\mathcal{S}^c$ should be 1 and 0, yielding $C = 0$. The equations for $\mathcal{H}^*$ and C are

$$\mathcal{H}^* = \frac{\mathcal{H}(\phi(X), Y)}{\sqrt{\mathcal{H}(\phi(X), \phi(X))\mathcal{H}(Y, Y)}} \quad \text{and} \quad C = \frac{\sum_{i,j \in \mathcal{S}^c} \langle \phi(x_i), \phi(x_j) \rangle}{\sum_{i,j \in \mathcal{S}} \langle \phi(x_i), \phi(x_j) \rangle}. \tag{10}$$

The RFF length is set to 300 for all datasets and the σ_l that maximizes $\mathcal{H}^*$ is chosen. The convergence threshold for *$\mathcal{H}$-Sequence* is set at $\mathcal{H}_l > 0.99$. The network structures discovered by ISM for every dataset are recorded and provided in App. E.The MLPs that use MSE and CE have weights initialized via the Kaiming method [11]. 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 [12, 13, 14]. All experiments were conducted on an Intel Xeon(R) CPU E5-2630 v3 @ 2.40GHz x 16 with 16 total cores.

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

Can *$\mathcal{H}$-Sequence* be optimized greedily? The $\mathcal{H}^*$ column in Table 1 consistently reports results that converge near its theoretical maximum value of 1, thereby corroborating with Thm. 1. As we discover a better kernel, *KNet* discovers a mapping that separates the dataset into distinguishable clusters, producing high training accuracies as $\mathcal{H}_l \to \mathcal{H}^*$. Will our network generalize? Since smooth mappings are associated with better generalization, we also report the smallest σ value used for each network to highlight the smoothness of ϕ learned by ISM. Correspondingly, with the exception of the two random datasets, our test accuracy consistently performed well across all datasets. While we cannot definitively attribute the impressive test results to Thm. 2, the experimental evidence appears to be aligned with its implication.

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

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

Since Thm. 1 also claims that we can achieve $\mathcal{H}^* - \mathcal{H}_l < \delta$ in finite number of layers, we include in Table 1 the average length of the *$\mathcal{H}$-Sequence* (L). The table suggests that the *$\mathcal{H}$-Sequence* converges quickly with 9 layers as the deepest network. The execution time for each objective is also recorded for reference in Table 1. Since *KNet* can be solved via a single forward pass while SGD requires many iterations of BP, *KNet* should be faster. The Time column of Table 1 confirmed this expectation by a wide margin. The biggest difference can be observed by comparing the face dataset, $\mathcal{H}$ finished with 0.78 seconds while MSE required 745 seconds; that is almost 1000 times difference. While the execution times reflect our expectation, techniques that vastly accelerate kernel computations [15, 16] would be required for larger datasets. Lastly, *KNet* induces low C as shown in Table 1, implying that samples in $\mathcal{S}$ and $\mathcal{S}^c$ are being pulled together and pushed apart in RKHS via the angular distance.

References

[1] 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.

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

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

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

[5] 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.

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

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

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

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

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

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

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

[18] 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.

Appendix A Proof for Theorem 1

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

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

$$\mathcal{H}^* - \mathcal{H}_L \leq \delta, \tag{11}$$

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

$$\lim_{L\to\infty} \mathcal{H}_L = \mathcal{H}^*, \tag{12}$$

III. the convergence is monotonic where

$$\mathcal{H}_l > \mathcal{H}_{l-1} \quad \forall l. \tag{13}$$

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

$$\mathcal{H}_l \geq \mathscr{L}(\sigma_0, \sigma_1). \tag{14}$$

Basic Background, Assumptions, and Notations.

1. The simulation of this theorem for Adversarial and Random data is also publicly available on `https://github.com/anonymous`.
2. Here we show that this bound can be established given the last 2 layers.
3. σ_0 is the σ value of the previous layer
4. σ_1 is the σ value of the current layer
5. τ is the number of classes
6. n is total number of samples
7. n_i is number of samples in the i^{th} class
8. $\mathcal{S}$ is a set of all i, j sample pairs where r_i and r_j belong to the same class.
9. $\mathcal{S}^c$ is a set of all i, j sample pairs where r_i and r_j belong to different same classes.
10. $\mathcal{S}^\beta$ is a set of all i, j sample pairs that belongs to the same β^{th} classes.
11. $r_i^{(\alpha)}$ is the i^{th} sample in the α^{th} class among τ classes.
12. We assume no $r_i \neq r_j$ pair are equal $\forall i \neq j$.
13. Among all $r_i \neq r_j$ pairs, there exists an optimal r_i^*, r_j^* pair where $\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle$ $\forall r_i \neq r_i^*$ and $r_j \neq r_j^*$. We denote this maximum inner product as
$$u_{\sigma_0} = \langle r_i^*, r_j^* \rangle. \tag{15}$$
14. In *KNet*, each r_i sample is assumed to be a sample in the RKHS of the Gaussian kernel, therefore all inner products are bounded such that
$$0 \leq \langle r_i, r_j \rangle \leq u_{\sigma_0}. \tag{16}$$
15. We let W be
$$W_s = \frac{1}{\sqrt{\zeta}} \left[\sum_\iota r_\iota^{(1)} \quad \sum_\iota r_\iota^{(2)} \quad ... \quad \sum_\iota r_\iota^{(\tau)} \right]. \tag{17}$$
Instead of using an optimal W^* defined as $W^* = \arg\max_W H_l(W)$, we use a suboptimal W_s where each dimension is simply the average direction of each class: $\frac{1}{\sqrt{\zeta}}$ is a normalizing constant $\zeta = ||W_s||_2^2$ that ensures $W_s^T W_s = I$. By using W_s, this implies that the $\mathcal{H}$ we obtain is already a lower bound compare $\mathcal{H}$ obtained by W^*. But, we will use this suboptimal W_s to identify an even lower bound. Note that based on the definition W^*, we have the property $\mathcal{H}(W^*) \geq \mathcal{H}(W)\, \forall W$.

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

$$\mathcal{H} = \underbrace{\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma_1^2}}}_{\mathscr{W}} - \underbrace{\sum_{i,j\in\mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma_1^2}}}_{\mathscr{B}} \tag{18}$$

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

Proof.

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

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

$$\mathscr{W} = \underbrace{\sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{(r_i^{(1)} - r_j^{(1)})^T W W^T (r_i^{(1)} - r_j^{(1)})}{2\sigma_1^2}}}_{\mathscr{W}_1} + ... + \underbrace{\sum_{\mathcal{S}^\tau} \Gamma_{i,j} e^{-\frac{(r_i^{(\tau)} - r_j^{(\tau)})^T W W^T (r_i^{(\tau)} - r_j^{(\tau)})}{2\sigma_1^2}}}_{\mathscr{W}_\tau} \tag{19}$$

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

$$\underbrace{(r_i^{(1)} - r_j^{(1)})^T W}_{\Pi_1} \underbrace{W^T (r_i^{(1)} - r_j^{(1)})}_{\Pi_2}. \tag{20}$$

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

$$\Pi_1 = \underbrace{r_i^{(1)^T} W}_{\xi_1} - \underbrace{r_j^{(1)^T} W}_{\xi_2} \tag{21}$$

$$\xi_1 = \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{22}$$

$$= \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{23}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} r_j^{(1)^T} \begin{bmatrix} \sum_\iota r_\iota^{(1)} & \sum_\iota r_\iota^{(2)} & ... & \sum_\iota r_\iota^{(\tau)} \end{bmatrix} \tag{24}$$

$$= \frac{1}{\sqrt{\zeta}} r_j^{(1)^T} \begin{bmatrix} (r_1^{(1)} + ... + r_{n_1}^{(1)}) & ... & (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \end{bmatrix} \tag{25}$$

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

$$\xi_1 = \frac{1}{\sqrt{\zeta}} [1 + (n_1 - 1)u_{\sigma_0} \quad n_2 u_{\sigma_0} \quad n_3 u_{\sigma_0} \quad ... \quad n_\tau u_{\sigma_0}] \tag{26}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} [1 \quad 0 \quad 0 \quad ... \quad 0]. \tag{27}$$

Which leads to

$$\Pi_1 = \frac{1}{\sqrt{\zeta}} (\xi_1 - \xi_2) = \frac{1}{\sqrt{\zeta}} [(n_1 - 1)u_{\sigma_0} \quad n_2 u_{\sigma_0} \quad n_3 u_{\sigma_0} \quad ... \quad n_\tau u_{\sigma_0}] \tag{28}$$

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

$$\Pi_1 \Pi_2 = \frac{1}{\zeta} [(n_1 - 1)^2 u_{\sigma_0}^2 + n_2^2 u_{\sigma_0}^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2] \tag{29}$$

$$= \frac{1}{\zeta} [(n_1 - 1)^2 + n_2^2 + n_3^2 + ... + n_\tau^2] u_{\sigma_0}^2 \tag{30}$$

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

$$\mathscr{W}_1 \geq \sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{[(n_1-1)^2+n_2^2+n_3^2+...+n_\tau^2]u_{\sigma_0}^2}{2\zeta\sigma_1^2}}. \tag{31}$$

To further condense the notation, we define the following constant

$$\mathscr{N}_g = \frac{1}{2\zeta}[n_1^2 + n_2^2 + ... + (n_g - 1)^2 + ... + n_\tau^2]. \tag{32}$$

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

$$\mathscr{W}_1 \geq \sum_{\mathcal{S}^1} \Gamma_{i,j} e^{-\frac{\mathscr{N}_1 u_{\sigma_0}^2}{\sigma_1^2}} \tag{33}$$

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

$$\mathscr{W}_g \geq \sum_{\mathcal{S}^i} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}. \tag{34}$$

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

$$\sum_{i,j\in\mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i-r_j)^T W W^T (r_i-r_j)}{2\sigma_1^2}} = \mathscr{W}_1 + ... + \mathscr{W}_\tau \geq \underbrace{\sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}}_{\text{Lower bound}}. \tag{35}$$

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

$$-\mathscr{B}_{g_1,g_2} = -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{(r_i^{(g_1)}-r_j^{(g_2)})^T W W^T (r_i^{(g_1)}-r_j^{(g_2)})}{2\sigma_1^2}} \tag{36}$$

$$= -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T((r_i^{(g_1)}-r_j^{(g_1)}))((r_i^{(g_1)}-r_j^{(g_2)}))^T W)}{2\sigma_1^2}} \tag{37}$$

$$= -\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(g_1,g_2)} W)}{2\sigma_1^2}} \tag{38}$$

$$\tag{39}$$

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

$$\begin{aligned}
\mathscr{B} = &-\underbrace{\sum_{i\in\mathcal{S}^1}\sum_{j\in\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(1,2)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{1,2}} - \underbrace{...}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^1}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(1,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{1,\tau}} \\
&-\underbrace{\sum_{i\in\mathcal{S}^2}\sum_{j\in\mathcal{S}^1} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(2,1)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{2,1}} - \underbrace{...}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^2}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(2,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{2,\tau}} \\
&... \\
&-\underbrace{\sum_{i\in\mathcal{S}^\tau}\sum_{j\in\mathcal{S}^1} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(\tau,1)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{\tau,1}} - \underbrace{...}_{\mathscr{B}_{g_1\neq g_2}} - \underbrace{\sum_{i\in\mathcal{S}^{\tau-1}}\sum_{j\in\mathcal{S}^\tau} |\Gamma_{i,j}| e^{-\frac{\mathrm{Tr}(W^T A_{i,j}^{(\tau-1,\tau)} W)}{2\sigma_1^2}}}_{\mathscr{B}_{\tau-1,\tau}}
\end{aligned} \tag{40}$$

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

$$\underbrace{(r_i^{(1)} - r_j^{(2)})^T W}_{\Pi_1} \underbrace{W^T (r_i^{(1)} - r_j^{(2)})}_{\Pi_2}. \tag{41}$$

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

$$\Pi_1 = \underbrace{r_i^{(1)^T} W}_{\xi_1} - \underbrace{r_j^{(2)^T} W}_{\xi_2} \tag{42}$$

$$\xi_1 = \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \left[\sum_\iota r_\iota^{(1)} \quad \sum_\iota r_\iota^{(2)} \quad ... \quad \sum_\iota r_\iota^{(\tau)} \right] \tag{43}$$

$$= \frac{1}{\sqrt{\zeta}} r_i^{(1)^T} \left[(r_1^{(1)} + ... + r_{n_1}^{(1)}) \quad ... \quad (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \right] \tag{44}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} r_j^{(2)^T} \left[\sum_\iota r_\iota^{(1)} \quad \sum_\iota r_\iota^{(2)} \quad ... \quad \sum_\iota r_\iota^{(\tau)} \right] \tag{45}$$

$$= \frac{1}{\sqrt{\zeta}} r_j^{(2)^T} \left[(r_1^{(1)} + ... + r_{n_1}^{(1)}) \quad ... \quad (r_1^{(\tau)} + ... + r_{n_\tau}^{(\tau)}) \right] \tag{46}$$

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

$$\xi_1 = \frac{1}{\sqrt{\zeta}} [1 \quad 0 \quad 0 \quad ... \quad 0] \tag{47}$$

$$\xi_2 = \frac{1}{\sqrt{\zeta}} [n_1 u_{\sigma_0} \quad 1 + (n_2 - 1) u_{\sigma_0} \quad n_3 u_{\sigma_0} \quad ... \quad n_\tau u_{\sigma_0}] \tag{48}$$

Which leads to

$$\Pi_1 = \frac{1}{\sqrt{\zeta}} (\xi_1 - \xi_2) = \frac{1}{\sqrt{\zeta}} [1 - n_1 u_{\sigma_0} \quad -(1 + (n_2 - 1) u_{\sigma_0}) \quad -n_3 u_{\sigma_0} \quad ... \quad -n_\tau u_{\sigma_0}] \tag{49}$$

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

$$\Pi_1 \Pi_2 = \frac{1}{\zeta} [(1 - n_1 u_{\sigma_0})^2 + (1 + (n_2 - 1) u_{\sigma_0})^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2]. \tag{50}$$

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

$$-\mathscr{B}_{1,2} \geq -\sum_{\mathcal{S}^1} \sum_{\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{(1 - n_1 u_{\sigma_0})^2 + (1 + (n_2 - 1) u_{\sigma_0})^2 + n_3^2 u_{\sigma_0}^2 + ... + n_\tau^2 u_{\sigma_0}^2}{2 \zeta \sigma_1^2}}. \tag{51}$$

To further condense the notation, we define the following function

$$\begin{aligned} \mathscr{N}_{g_1, g_2}(u_{\sigma_0}) = \frac{1}{2\zeta} [&n_1^2 u_{\sigma_0}^2 + n_2^2 u_{\sigma_0}^2 + ... \\ &+ (1 - n_{g_1} u_{\sigma_0})^2 + ... + (1 + (n_{g_2} - 1) u_{\sigma_0})^2 \\ &+ ... + n_\tau^2 u_{\sigma_0}^2]. \end{aligned} \tag{52}$$

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

$$-\mathscr{B}_{1,2} \geq -\sum_{\mathcal{S}^1} \sum_{\mathcal{S}^2} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{1,2}(u_{\sigma_0})}{\sigma_1^2}} \tag{53}$$

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

$$-\mathscr{B}_{g_1,g_2} \geq -\sum_{\mathcal{S}^{g1}}\sum_{\mathcal{S}^{g2}} \Gamma_{i,j} e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}. \tag{54}$$

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

$$-\sum_{i,j\in\mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i-r_j)^T W W^T (r_i-r_j)}{2\sigma_1^2}} = -\mathscr{B}_{1,2} - \mathscr{B}_{1,3} - ... - \mathscr{B}_{\tau-1,\tau} \tag{55}$$

$$\geq -\underbrace{\sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}}_{\text{Lower bound}}. \tag{56}$$

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

$$\mathcal{H} = \mathscr{W} + \mathscr{B} \tag{57}$$

$$\geq \underbrace{\sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}}}_{\text{Lower bound of } \mathscr{W}} - \underbrace{\sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}}_{\text{Lower bound of } \mathscr{B}}. \tag{58}$$

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

$$\mathscr{L}(\sigma_0,\sigma_1) = \sum_{g=1}^{\tau}\sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}} - \sum_{g_1\neq g_2}^{\tau}\sum_{i\in\mathcal{S}^{g_1}}\sum_{j\in\mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}. \tag{59}$$

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

$\square$

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

$$\lim_{\sigma_0\to 0} u_{\sigma_0} = 0. \tag{60}$$

Assumptions and Notations.

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

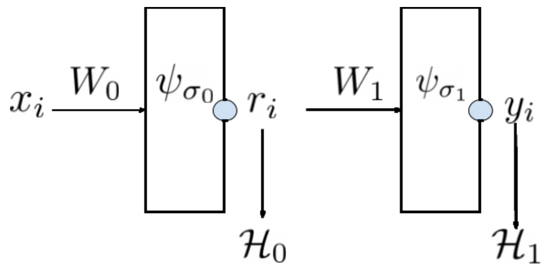

Figure 1: Figure of a 2 layer network.

4. As defined from Lemma 1, among all $r_i \neq r_j$ pairs, there exists an optimal r_i^*, r_j^* pair where $\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle \; \forall r_i \neq r_i^*$ and $r_j \neq r_j^*$. We denote this maximum inner product as
$$u_{\sigma_0} = \langle r_i^*, r_j^* \rangle. \tag{61}$$

Proof.

Given Fig. 1, the equation for $\mathcal{H}_0$ is

$$\mathcal{H}_0 = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(x_i - x_j)^T W W^T (x_i - x_j)}{2\sigma_0^2}} - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(x_i - x_j)^T W W^T (x_i - x_j)}{2\sigma_0^2}} \tag{62}$$

$$= \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle \tag{63}$$

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

$$\lim_{\sigma_0 \to 0} \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle = \begin{cases} 0 & \forall i \neq j \\ 1 & \forall i = j \end{cases}. \tag{64}$$

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

$$\langle \psi_{\sigma_0}(x_i^*), \psi_{\sigma_0}(x_j^*) \rangle \geq \langle \psi_{\sigma_0}(x_i), \psi_{\sigma_0}(x_j) \rangle \tag{65}$$

or equivalently

$$\langle r_i^*, r_j^* \rangle \geq \langle r_i, r_j \rangle \tag{66}$$

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

$$\lim_{\sigma_0 \to 0} u_{\sigma_0} = \lim_{\sigma_0 \to 0} \exp -(|.|/\sigma_0^2) = 0 \tag{67}$$

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

□

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

$$\mathscr{L}(\sigma_1) = \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{1}{\zeta \sigma_1^2}}. \tag{68}$$

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

$$\lim_{\sigma_1 \to 0} \mathscr{L}(\sigma_1) = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \tag{69}$$

$$= \mathcal{H}^*. \tag{70}$$

Proof.

Given Lemma 2, we know that

$$\lim_{\sigma_0 \to 0} u_{\sigma_0} = 0. \tag{71}$$

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

$$\mathscr{L}(\sigma_1) = \lim_{u_{\sigma_0} \to 0} \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} e^{-\frac{\mathscr{N}_g u_{\sigma_0}^2}{\sigma_1^2}} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{\mathscr{N}_{g_1,g_2}(u_{\sigma_0})}{\sigma_1^2}}, \tag{72}$$

$$= \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} - \sum_{g_1 \neq g_2}^{\tau} \sum_{i \in \mathcal{S}^{g_1}} \sum_{j \in \mathcal{S}^{g_2}} |\Gamma_{i,j}| e^{-\frac{1}{\zeta \sigma_1^2}}. \tag{73}$$

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

$$\lim_{\sigma_1 \to 0} \mathscr{L}(\sigma_1) = \sum_{g=1}^{\tau} \sum_{\mathcal{S}^g} \Gamma_{i,j} = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} \tag{74}$$
$$= \mathcal{H}^*. \tag{75}$$

□

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

$$\mathcal{H}^* - \mathcal{H}_l \leq \delta. \tag{76}$$

Proof.

Observation 1.

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

$$\begin{aligned}\mathcal{H}_l = \max_W \sum_{i,j \in \mathcal{S}} \Gamma_{i,j} e^{-\frac{(r_i^{(\mathcal{S})} - r_j^{(\mathcal{S})})^T W W^T (r_i^{(\mathcal{S})} - r_j^{(\mathcal{S})})}{2\sigma_1^2}} \\ - \sum_{i,j \in \mathcal{S}^c} |\Gamma_{i,j}| e^{-\frac{(r_i^{(\mathcal{S}^c)} - r_j^{(\mathcal{S}^c)})^T W W^T (r_i^{(\mathcal{S}^c)} - r_j^{(\mathcal{S}^c)})}{2\sigma_1^2}}.\end{aligned} \tag{77}$$

Since the Gaussian kernel is bounded between 0 and 1, the theoretical maximum of $\mathcal{H}^*$ is when the kernel is 1 for $\mathcal{S}$ and 0 for $\mathcal{S}^c$ with the theoretical maximum as $\mathcal{H}^* = \sum_{i,j \in \mathcal{S}} \Gamma_{i,j}$. Therefore Eq. (76) inequality is equivalent to

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} - \mathcal{H}_l \leq \delta. \tag{78}$$

Observation 2.

If we choose a σ_0 such that

$$\mathscr{L}^*(\sigma_1) - \mathscr{L}(\sigma_0, \sigma_1) \leq \frac{\delta}{2} \quad \text{and} \quad \mathcal{H}^* - \mathscr{L}^*(\sigma_1) \leq \frac{\delta}{2} \tag{79}$$

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

$$\sum_{i,j \in \mathcal{S}} \Gamma_{i,j} - \mathscr{L}(\sigma_0, \sigma_1) \leq \delta. \tag{80}$$

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

$$\mathcal{H}^* - \mathscr{L}^*(\sigma_1) \leq \frac{\delta}{2}. \tag{81}$$

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

$$\mathscr{L}^*(\sigma_1) - \mathscr{L}(\sigma_0, \sigma_1) \leq \frac{\delta}{2} \tag{82}$$

Then we have:

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

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

□

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

$$\lim_{l\to\infty} \mathcal{H}_l = \mathcal{H}^*, \quad \mathcal{H}_{l+1} > \mathcal{H}_l \quad \forall l \tag{84}$$

Before, the proof, we use the following figure, Fig. 2, to illustrate the relationship between *Kernel Sequence* $\{\phi_{l\circ}\}_{l=1}^{L}$ that generates the $\mathcal{H}$-*Sequence* $\{\mathcal{H}_l\}_{l=1}^{L}$. By solving a network greedily, we separate the network into L separable problems. At each additional layer, we rely on the weights learned from the previous layer. At each network, we find σ_{l-1}, σ_l, and W_l for the next network. We also note that since we only need to prove the existence of a solution, this proof is done by ***Proof by Construction***, i.e, we only need to show an example of its existence. Therefore, this proof consists of us constructing a $\mathcal{H}$-*Sequence* which satisfies the lemma.

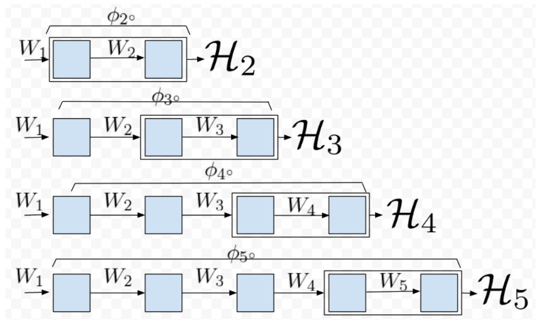

Figure 2: Relating *Kernel Sequence* to $\mathcal{H}$-*Sequence*.

Proof.

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

$$\mathcal{H}^* - \mathcal{H}_l \le \delta_l. \tag{85}$$

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

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

$$\lim_{n\to\infty} \mathcal{E}_n = \mathcal{H}^*, \quad \mathcal{E}_1 = \mathcal{H}_0. \tag{86}$$

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

$$\mathcal{E}_n \le \mathcal{H}_{l-1} < \mathcal{E}_{n+1}. \tag{87}$$

Step 2: For any given l, we choose δ_l to satisfies Eq. (85) by the following procedure, First find an n that satisfies

$$\mathcal{E}_n \le \mathcal{H}_{l-1} < \mathcal{E}_{n+1}, \tag{88}$$

and second define δ_l to be

$$\delta_l = \mathcal{H}^* - \mathcal{E}_{n+1}. \tag{89}$$

To satisfy Eq. (85), the following must be true.

$$\mathcal{H}^* - \mathcal{H}_{l-1} \le \delta_{l-1}. \tag{90}$$

and further we found n such that

$$\mathcal{E}_n \le \mathcal{H}_{l-1} < \mathcal{E}_{n+1} \implies \mathcal{H}^* - \mathcal{E}_n \ge \mathcal{H}^* - \mathcal{H}_{l-1} > \mathcal{H}^* - \mathcal{E}_{n+1}. \tag{91}$$

Thus combining Eq. (89), Eq. (90), and Eq. (91) we have

$$\delta_{l-1} > \delta_l. \tag{92}$$

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

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

$$\lim_{n\to\infty} \mathcal{H}^* - \frac{\mathcal{H}^* - \mathcal{H}_0}{n} = \mathcal{H}^*. \tag{93}$$

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

$$\lim_{n\to\infty} \mathcal{H}^* - \mathcal{H}^* + \frac{\mathcal{H}^* - \mathcal{H}_0}{n} = 0 \tag{94}$$

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

$$\lim_{l\to\infty} \delta_l = 0. \tag{95}$$

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

$$\mathcal{H}^* - \mathcal{H}_l \le \delta_l. \tag{96}$$

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

$$\mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l \le \lim_{l\to\infty} \delta_l, \tag{97}$$

$$\le 0. \tag{98}$$

Since we know that

$$\mathcal{H}^* - \mathcal{H}_l \ge 0 \, \forall l. \tag{99}$$

The condition of

$$0 \le \mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l \le 0 \tag{100}$$

is true only if

$$\mathcal{H}^* - \lim_{l\to\infty} \mathcal{H}_l = 0. \tag{101}$$

This allows us to conclude

$$\mathcal{H}^* = \lim_{l\to\infty} \mathcal{H}_l. \tag{102}$$

Proof of the Monotonic Improvement.

Given Eq. (87) and Eq. (89), at each step we have the following:

$$\mathcal{H}_{l-1} < \mathcal{E}_{n+1} \tag{103}$$

$$\le \mathcal{H}^* - \delta_l. \tag{104}$$

Rearranging this inequality, we have

$$\delta_l < \mathcal{H}^* - \mathcal{H}_{l-1}. \tag{105}$$

By combining the inequalities from Eq. (105) and Eq. (96), we have the following relationships.

$$\mathcal{H}^* - \mathcal{H}_l \le \delta_l < \mathcal{H}^* - \mathcal{H}_{l-1} \tag{106}$$

$$\mathcal{H}^* - \mathcal{H}_l < \mathcal{H}^* - \mathcal{H}_{l-1} \tag{107}$$

$$-\mathcal{H}_l < -\mathcal{H}_{l-1} \tag{108}$$

$$\mathcal{H}_l > \mathcal{H}_{l-1}, \tag{109}$$

$$\tag{110}$$

which concludes the proof of theorem.

□

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

Proof. We start with the Lagrangian

$$\mathcal{L} = -\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \text{Tr}(\Lambda(W^T W - I)). \tag{111}$$

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

$$\nabla \mathcal{L} = \frac{1}{\sigma^2} \sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i - r_j)(r_i - r_j)^T W - 2W\Lambda. \tag{112}$$

By setting the gradient to 0, we have

$$\left[\frac{1}{2\sigma^2} \sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i - r_j)(r_i - r_j)^T \right] W = W\Lambda. \tag{113}$$

$$\mathcal{Q}_l W = W\Lambda. \tag{114}$$

From Eq. (114), we see that W is only the optimal solution when W is the eigenvector of Q_l. Therefore, by setting W to $W_s = \frac{1}{\sqrt{\zeta}} \sum_\alpha r_\alpha$, it is not guaranteed to yield an optimal.

□

Appendix B Proof for Theorem 2

Theorem 2: *Eq. (3) objective is equivalent to*

$$\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} (r_i^T W W^T r_j) - \sum_i D_i(W) ||W^T r_i||_2. \tag{115}$$

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

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

Our layer wise HSIC objective becomes

$$\min_W -\sum_{i,j} \Gamma_{i,j} e^{-\frac{(r_i - r_j)^T W W^T (r_i - r_j)}{2\sigma^2}} - \text{Tr}[\Lambda(W^T W - I)]. \tag{117}$$

We take the derivative of the Lagrangian, the expression becomes

$$\nabla_W \mathcal{L}(W, \Lambda) = \sum_{i,j} \frac{\Gamma_{i,j}}{\sigma^2} e^{-\frac{\text{Tr}(W^T A_{i,j} W)}{2\sigma^2}} A_{i,j} W - 2W\Lambda. \tag{118}$$

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

$$\left[\sum_{i,j} \frac{\Gamma_{i,j}}{2\sigma^2} e^{-\frac{\text{Tr}(W^T A_{i,j} W)}{2\sigma^2}} A_{i,j} \right] W = W\Lambda \tag{119}$$

$$\left[\frac{1}{2} \sum_{i,j} \hat{\Gamma}_{i,j} A_{i,j} \right] W = W\Lambda \tag{120}$$

$$\mathcal{Q} W = W\Lambda. \tag{121}$$

From here, we see that the optimal solution is an eigenvector of $\mathcal{Q}$. Based on ISM, it further proved that the optimal solution is not just any eigenvector, but the eigenvectors associated with the smallest values of $\mathcal{Q}$. From this logic, ISM solves objective (117) with a surrogate objective

$$\min_W \quad \text{Tr}\left(W^T \left[\frac{1}{2} \sum_{i,j} \hat{\Gamma}_{i,j} A_{i,j} \right] W \right) \quad \text{s. t.}\, W^T W = I. \tag{122}$$

Given $D_{\hat{\Gamma}}$ as the degree matrix of $\hat{\Gamma}$ and $R = [r_1, r_2, ...]^T$, ISM further shows that Eq. (122) can be written into

$$\min_W \quad \text{Tr}\left(W^T R^T \left[D_{\hat{\Gamma}} - \hat{\Gamma}\right] RW\right) \quad \text{s.t.}\, W^T W = I \tag{123}$$

$$\max_W \quad \text{Tr}\left(W^T R^T \left[\hat{\Gamma} - D_{\hat{\Gamma}}\right] RW\right) \quad \text{s.t.}\, W^T W = I \tag{124}$$

$$\max_W \quad \text{Tr}\left(W^T R^T \hat{\Gamma} RW\right) - \text{Tr}\left(W^T R^T D_{\hat{\Gamma}} RW\right) \quad \text{s.t.}\, W^T W = I \tag{125}$$

$$\max_W \quad \text{Tr}\left(\hat{\Gamma} RWW^T R^T\right) - \text{Tr}\left(D_{\hat{\Gamma}} RWW^T R^T\right) \quad \text{s.t.}\, W^T W = I \tag{126}$$

$$\max_W \quad \sum_{i,j} \hat{\Gamma}_{i,j} [RWW^T R^T]_{i,j} - \sum_{i,j} D_{\hat{\Gamma}_{i,j}} [RWW^T R^T]_{i,j} \quad \text{s.t.}\, W^T W = I. \tag{127}$$

Since the jump from Eq. (122) can be intimidating for those not familiar with the literature, we included a more detailed derivation in App. C.

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

$$\max_W \quad \sum_{i,j} \hat{\Gamma}_{i,j} (r_i^T WW^T r_j) - \sum_i D_i(W) ||W^T r_i||_2 \quad \text{s.t.}\, W^T W = I. \tag{128}$$

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

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

$$\hat{\Gamma} = \frac{1}{\sigma^2} \begin{bmatrix} \left[\Gamma_{\mathcal{S}} e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma^2}}\right] & \left[-|\Gamma_{\mathcal{S}^c}| e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma^2}}\right] & ... \\ \left[-|\Gamma_{\mathcal{S}^c}| e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma^2}}\right] & \left[\Gamma_{\mathcal{S}} e^{-\frac{(r_i - r_j)^T WW^T (r_i - r_j)}{2\sigma^2}}\right] & ... \\ ... & ... & ... \end{bmatrix}. \tag{129}$$

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

$$D_i(W_l) = \frac{1}{\sigma^2} \sum_{j \in \mathcal{S}|i} \Gamma_{i,j} \mathcal{K}_{W_l}(r_i, r_j) - \frac{1}{\sigma^2} \sum_{j \in \mathcal{S}^c|i} |\Gamma_{i,j}| \mathcal{K}_{W_l}(r_i, r_j). \tag{130}$$

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

$$\hat{\Gamma}^* = \frac{1}{\sigma^2} \begin{bmatrix} [\Gamma_{\mathcal{S}}] & [0] & ... \\ [0] & [\Gamma_{\mathcal{S}}] & ... \\ ... & ... & ... \end{bmatrix}. \tag{131}$$

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

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

$$D_i^*(W) > D_i(W). \tag{132}$$

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

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

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

If we expand the 1st term, we get

$$2\sum_i^n\sum_j^n \Psi_{i,j}(x_i x_i^T) = 2\sum_i \Psi_{i,1}(x_i x_i^T) + \ldots + \Psi_{i,n}(x_i x_i^T) \tag{133}$$

$$= 2\sum_i^n [\Psi_{1,1} + \Psi_{1,2} + ...] x_i x_i^T \tag{134}$$

$$= 2\sum_i^n d_i x_i x_i^T \tag{135}$$

$$= 2X^T D_\Psi X \tag{136}$$

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

$$2\sum_i\sum_j \Psi_{i,j} x_i x_j^T = 2\sum_i \Psi_{i,1} x_i x_1^T + \Psi_{i,2} x_i x_2^T + \Psi_{i,3} x_i x_3^T + ... \tag{137}$$

$$= 2\sum_i x_i(\Psi_{i,1} x_1^T) + x_i(\Psi_{i,2} x_2^T) + x_i(\Psi_{i,3} x_3^T) + ... \tag{138}$$

$$= 2\sum_i x_i \left[(\Psi_{i,1} x_1^T) + (\Psi_{i,2} x_2^T) + (\Psi_{i,3} x_3^T) + ...\right] \tag{139}$$

$$= 2\sum_i x_i \left[X^T \Psi_i^T\right]^T \tag{140}$$

$$= 2\sum_i x_i \left[\Psi_i X\right] \tag{141}$$

$$= 2\left[x_1 \Psi_1 X + x_2 \Psi_2 X + x_3 \Psi_3 X + ...\right] \tag{142}$$

$$= 2\left[x_1 \Psi_1 + x_2 \Psi_2 + x_3 \Psi_3 + ...\right] X \tag{143}$$

$$= 2X^T \Psi X \tag{144}$$

$$\tag{145}$$

Putting both terms together, we get

$$\sum_{i,j} \Psi_{i,j} A_{i,j} = 2X^T D_\Psi X - 2X^T \Psi X a \tag{146}$$

$$= 2X^T [D_\Psi - \Psi] X \tag{147}$$

$$\tag{148}$$

Appendix D Dataset Details

No samples were excludes from any of the dataset.

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

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

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

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

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

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

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

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

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

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

Appendix E 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. [17] and Ansuini et al. [18].

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

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

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

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

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

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

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

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

Appendix F Graphs of Kernel Sequences

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

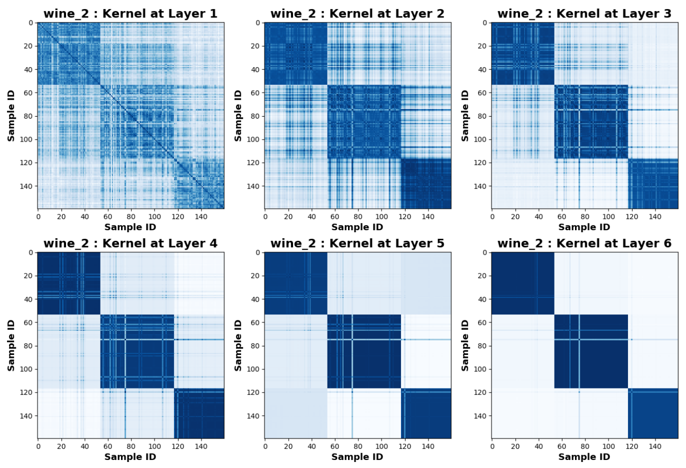

Figure 3: The kernel sequence for the wine dataset.

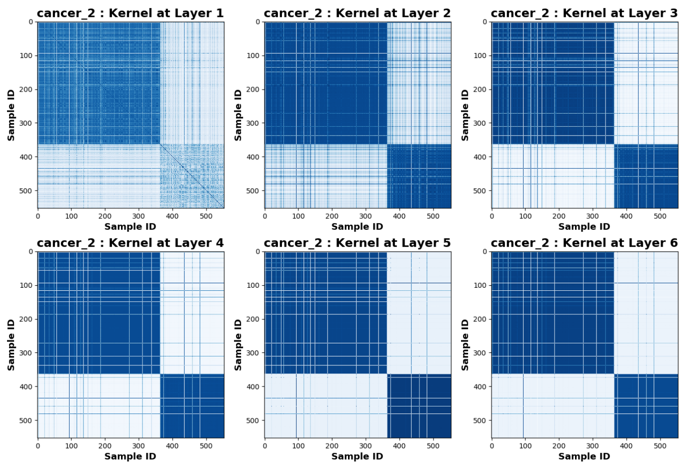

Figure 4: The kernel sequence for the cancer dataset.

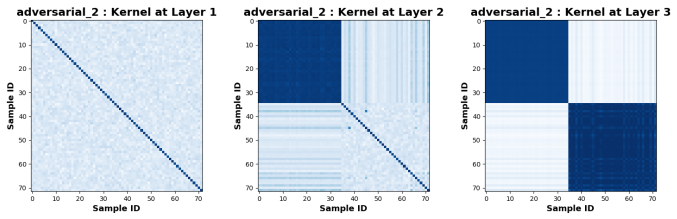

Figure 5: The kernel sequence for the Adversarial dataset.

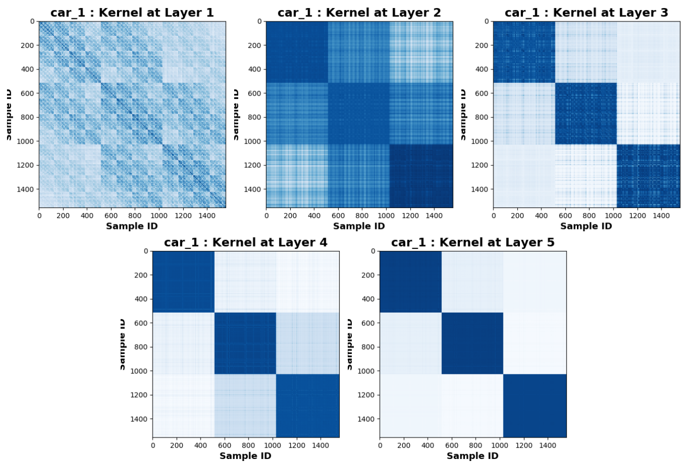

Figure 6: The kernel sequence for the car dataset.

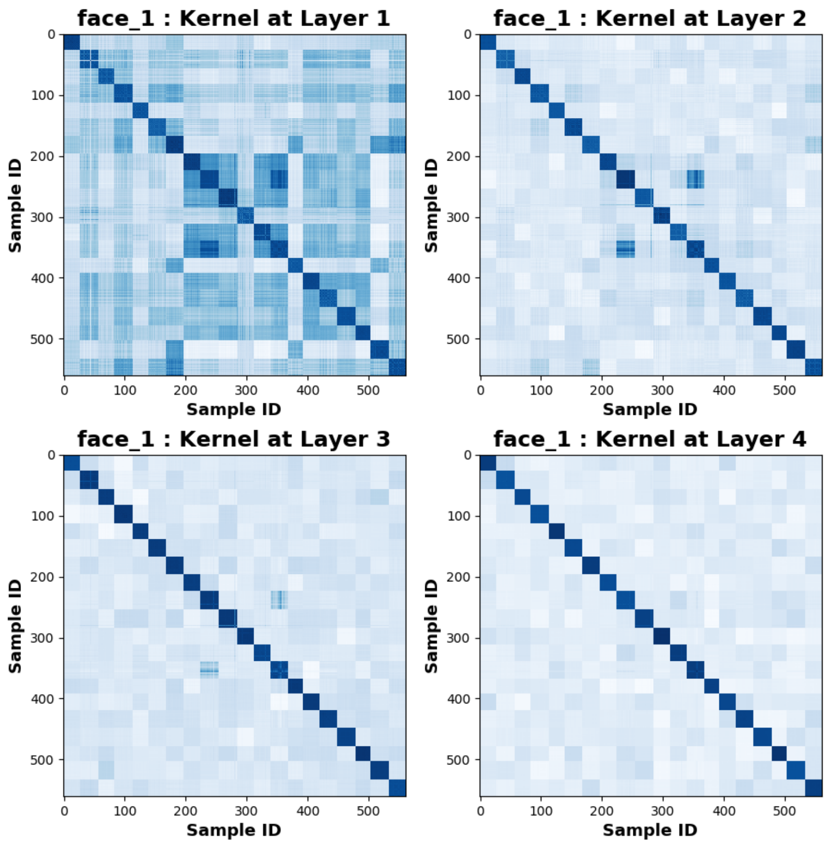

Figure 7: The kernel sequence for the face dataset.

Appendix G Optimal Gaussian σ for Maximum Kernel Separation

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

Let $X \in \mathbb{R}^{n \times d}$ be a dataset of n samples with d features and let $Y \in \mathbb{R}^{n \times \tau}$ be the corresponding one-hot encoded labels where τ denotes the number of classes. Given $\kappa_X(\cdot, \cdot)$ and $\kappa_Y(\cdot, \cdot)$ as two

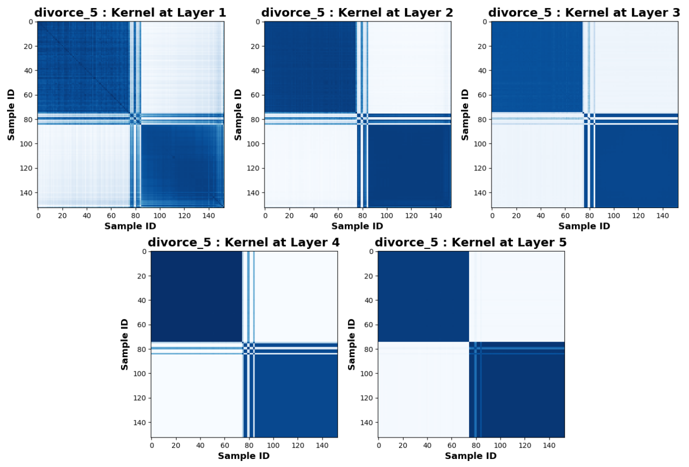

Figure 8: The kernel sequence for the divorce dataset.

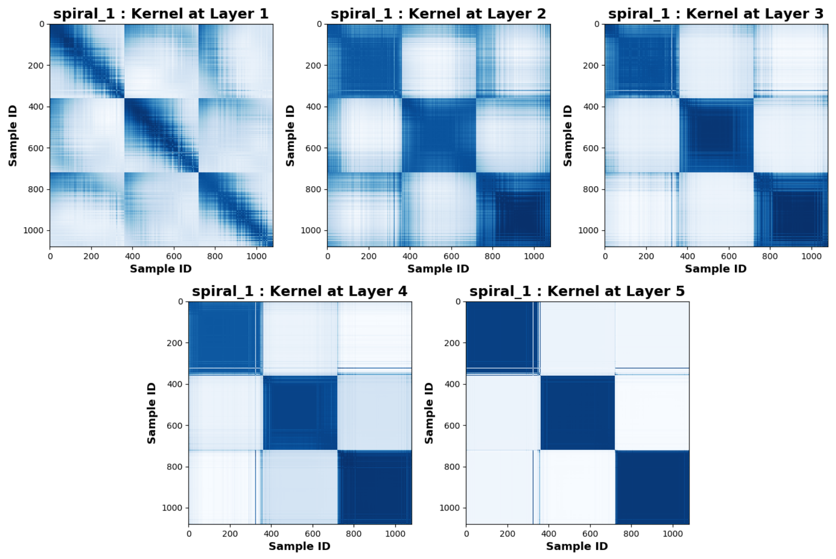

Figure 9: The kernel sequence for the spiral dataset.

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

$$d_{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \tag{149}$$

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

$$d_{\mathcal{S}^c} = \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{150}$$

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

$$\max_{\sigma} \quad \frac{1}{|\mathcal{S}|} \sum_{i,j \in \mathcal{S}} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} - \frac{1}{|\mathcal{S}^c|} \sum_{i,j \in \mathcal{S}^c} e^{-\frac{||x_i - x_j||^2}{2\sigma^2}}. \tag{151}$$

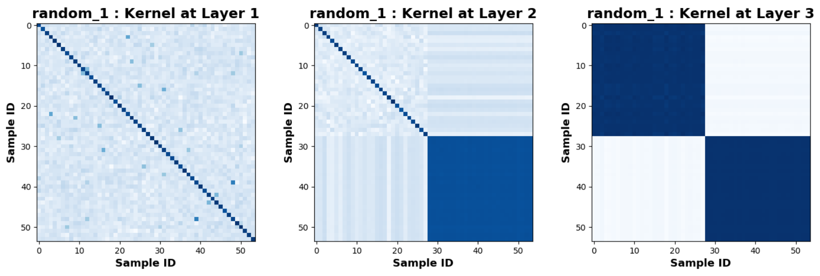

Figure 10: The kernel sequence for the Random dataset.

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

$$Q = -gK_Y + \bar{g}(\mathbf{1}_{n\times n} - K_Y). \tag{152}$$

Or Q can be written more compactly as

$$Q = \bar{g}\mathbf{1}_{n\times n} - (g + \bar{g})K_Y. \tag{153}$$

Given Q, Eq. (151) becomes

$$\min_{\sigma} \quad \text{Tr}(K_X Q). \tag{154}$$

This objective can be efficiently solved with BFGS.

Below in Fig. 11, 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.

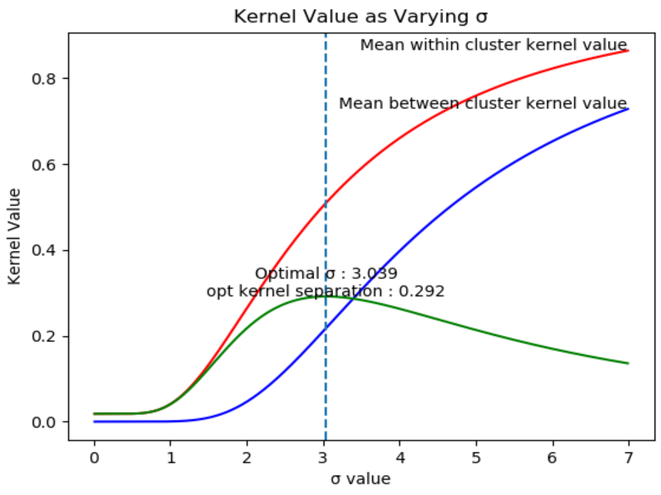

Figure 11: Maximum Kernel separation.

Relation to HSIC. From Eq. (154), 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

$$\min_{\sigma} \quad \text{Tr}(K_X H K_Y H), \tag{155}$$

by setting $Q = HK_YH$, we can see how the two formulations are related. While the maximum kernel separation places the weight of each sample pair equally, HSIC weights the pair differently. We also notice that the $Q_{i,j}$ element is positive/negative for (x_i, x_j) pairs that are with/between classes respectively. Therefore, the argument for the global optimum should be relatively close for both objectives. Below in Figure 12, we show a figure of HSIC values as we vary σ. Notice how the optimal σ is almost equivalent to the solution from maximum kernel separation. For the purpose of *KNet*, we use σ that maximizes the HSIC value.

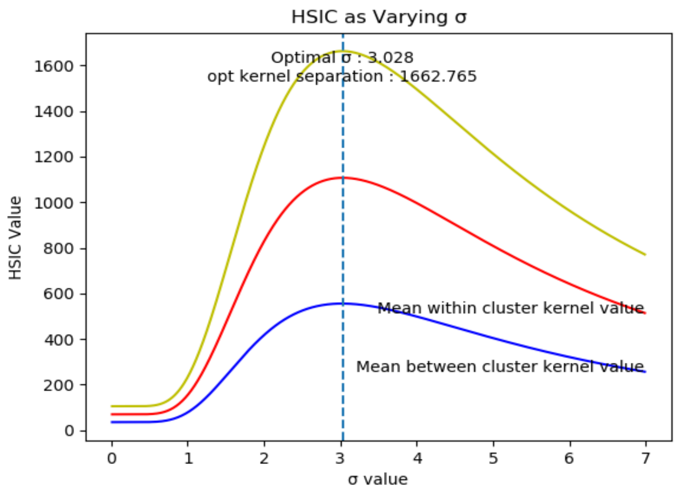

Figure 12: Maximal HSIC.