KERNEL ALIGNMENT USING MANIFOLD APPROXIMATION

Mohammad Tariqul Islam, Du Liu, Deblina Sarker*
Nano-Cybernetic BioTrek, Media Lab,
Massachusetts Institute of Technology
75 Amherst St, Cambridge, MA 02139
{mhdtariq,liudu,deblina}@mit.edu

ABSTRACT

Centered kernel alignment (CKA) is a popular metric for comparing representation, determining equivalence of networks, and conducting neuroscience research. However, CKA does not account for the underlying manifold and relies on many heuristics that make it behave differently at different scales of data. In this work, we propose Manifold-approximated Kernel Alignment (MKA) that incorporates manifold geometry into the alignment task. We derive a theoretical framework for MKA. We perform empirical evaluations on synthetic datasets and real-world examples to characterize and compare MKA to CKA. Our findings suggest that manifold-aware kernel alignment provides a more robust foundation for measuring representations, with potential applications in representation learning.

1 INTRODUCTION

Centered Kernel Alignment (CKA) (Cortes et al., 2010; Kornblith et al., 2019) is a statistical method used to compare the similarity between representations of data, often in the form of feature maps or embeddings. It works by aligning kernels, which capture pairwise relationships within datasets, and measuring their agreement. CKA is widely used in studies to compare layers of neural networks, analyze representational similarity, and study how models process information (Ramasesh et al., 2021; Nguyen et al., 2022; Ciernik et al., 2024). Its ability to handle datasets of different sizes and dimensions makes it a powerful tool for understanding complex models and evaluating their performance. However, very few studies have characterized CKA under known representations/topologies. Moreover, the reliability of the CKA measure has been under scrutiny numerous times (Davari et al., 2023; Murphy et al., 2024).

To address this, we propose Manifold-approximated Kernel Alignment (MKA). Manifold approximation is a way of understanding and simplifying complex data. In many real-world problems, data with many dimensions - like x-rays, medical records, and neuroimaging data - actually lie on a much smaller, curved structure called a "manifold" within the high-dimensional space. Known as the 'manifold hypothesis', this concept is integral to modern statistics and learning algorithms (Fefferman et al., 2016). Manifold approximation uncovers and represents this underlying structure within the high-dimensional data by exploiting the relationships between data points. It is an integral part of non-linear dimensionality reduction, e.g., t-distributed Stochastic Neighbor Embedding (Van der Maaten & Hinton, 2008) and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018).

We use manifold approximation to define a non-linear and non-Mercer kernel. Using this kernel function, we provide a theoretical framework for MKA. With extensive characterization on synthetic datasets, we show that MKA is more consistent under varying dimensionality and shapes that preserve topology. We also discover that MKA captures the underlying topology better and is less sensitive to hyperparameters than CKA. Finally, we revisit neural network representation and provide a new perspective. Overall, this work will pave the way for applying manifold approximation in diverse applications.

*corresponding author

2 CENTERED KERNEL ALIGNMENT (CKA)

Let $X \in \mathbb{R}^{N \times d_1}$ and $Y \in \mathbb{R}^{N \times d_2}$ be feature sets from N samples each with d_1 and d_2 features, respectively. The corresponding symmetric kernel matrices are K and L with $K_{ij} = k(x_i, x_j)$ and $L_{ij} = l(y_i, y_j)$, respectively. The CKA measure between the two feature sets is given by

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\,\mathrm{HSIC}(L, L)}}, \tag{1}$$

where $\mathrm{HSIC}(\cdot, \cdot)$ is the Hilbert-Schmidt independent criterion given by $\mathrm{HSIC}(K, L) = \frac{1}{(n-1)^2}\,\mathrm{trace}(KHLH)$. Here, $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is a centering matrix that mitigates bias in the kernel. There are other debiasing techniques (Song et al., 2007; Sucholutsky et al., 2023), however, we will consider the simplest and most widely used technique in practice. HSIC computes the similarity between the two kernel matrices of the same size, while the CKA measure normalizes this similarity within $[0, 1]$.

Various options exist for the kernel. The common ones include the linear kernel (LIN) given by $k(x_i, xj) = x_i^T x_j$ and the radial basis function (RBF) kernel given by $k(x_i, x_j) = \exp(-||x_i - x_j||/(2\sigma^2))$, where σ is the bandwidth of the Gaussian. The following theorem establishes an equivalence relation between CKA with linear and RBF kernel:

Theorem 2.1 (Alvarez (2022))**.** $\mathrm{CKA}(K_{\mathrm{RBF}}, L) = \mathrm{CKA}(K_{\mathrm{LIN}}, L) + O(1/\sigma^2)$ *as* $\sigma \to \infty$. *Here,* K_{RBF} *is the RBF kernel matrix with bandwidth* σ*,* K_{LIN} *is the linear kernel matrix, and* L *is any positive definite symmetric kernel matrix.*

Softly, it states that at higher values of σ, CKA with linear and RBF kernels behave equivalently. Various studies have reported this in empirical settings (e.g., in Kornblith et al. (2019) and Fig. 4(a) of Davari et al. (2023)). Thus, most researchers use the linear kernel, effectively capturing linear relationships alone. And by Theorem 2.1, even results with an RBF kernel (without properly tuning the bandwidth, σ) potentially suffer from the same pitfalls of the linear one.

3 MANIFOLD-APPROXIMATED KERNEL ALIGNMENT (MKA)

Manifold approximation is a method for defining a graph that quantifies the pairwise relations within the data. CKA already does this job by producing a dense kernel matrix that considers all possible pairs. In the field of non-linear dimensionality reduction, manifold approximation takes a central role in sampling the manifold of the data to reduce the complexity of computing the kernel matrix. This kernel is often sparse and typically obtained by the k-nearest neighbor (KNN) algorithm. Moreover, we will use a kernel function that is non-symmetric (i.e., $k(x_i, x_j) \neq k(x_j, x_i)$). Thus, our kernel will not be positive semidefinite; rather, it will fall in the class of indefinite or non-Mercer kernels (Ong et al., 2004). Here, we adopt the manifold approximation method from UMAP[1]. Our manifold-approximated kernel (K_U) defines a pairwise relationship by

$$K_{ij}^{(U)} = \begin{cases} 1, & \text{if } \; i = j \\ \exp\left(-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}\right) & \text{if } x_j \in \mathrm{KNN}(x_i, k)\ , \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\mathrm{KNN}(x_i, k)$ contains the k-nearest neighbors of x_i, $d(\cdot, \cdot)$ is a distance metric, $\rho_i = \min_{x_j \in \mathrm{KNN}(x_i, k)} d(x_i, x_j)$ is the minimum distance from the nearest neighbor and σ_i is a scaling parameter akin to bandwidth of RBF function. The scaling parameter is computed such that $\sum_j K_{ij}^{(U)} = 1 + \log_2(k)$. This constraint fixes the row of the kernel matrix to a constant and makes the kernel less sensitive to lone outliers. The KNN imposes a stricter constraint on the number of points that are considered related compared to CKA, which allows for a softer, more global measure of similarity. Overall, K_U is a graph on the data that depends on only one hyperparameter: k. Now, we define Manifold-approximated Kernel Alignment (MKA) as:

$$\mathrm{MKA}(K_U, L_U) = \frac{\langle K_U H, L_U H \rangle}{\sqrt{\langle K_U H, K_U H \rangle \langle L_U H, L_U H \rangle}}. \tag{3}$$

[1]UMAP uses a graph-based kernel. It performs a symmetrization step to define it. We skip this step for computational efficiency.

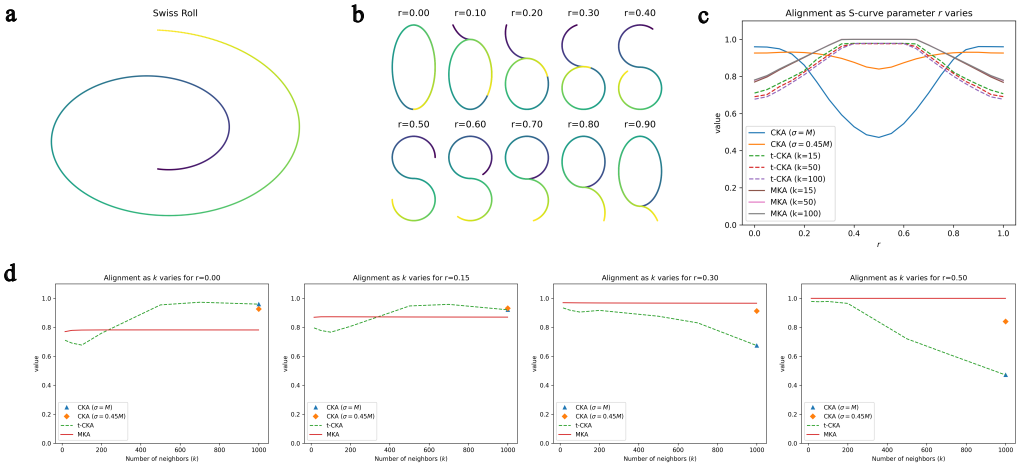

Figure 1: Equivalence of two different shapes with 1-D manifolds. (a) Swiss-roll. (b) S-curve by varying parameter r. (c) Alignment for the methods as S-curve parameter, r, varies. (d) Alignment for different methods as different number of nearest neighbors considered. Note that CKA does not have any notion of nearest neighbors; thus we have plotted the CKA value at the last point on the x-axis.

Despite using non-symmetric kernels, the measure MKA is symmetric ($\text{MKA}(K_U, L_U) = \text{MKA}(L_U, K_U)$). However, unlike CKA, which performs both row- and column-wise centering, we opted for only row-wise centering. This leaves additional bias terms in the estimation, however, we show in Appendix A.2 that this slight oversight does not make MKA less meaningful. Exploiting the properties of the kernel matrix we can simplify and characterize MKA by

Theorem 3.1. *If* $\sum_j K^{(U)}_{i,j} = D$ *and* $\sum_j L^{(U)}_{i,j} = D$, $\forall i$, *then* MKA *reduces to*

$$\text{MKA}(K_U, L_U) = \frac{\langle K_U, L_U \rangle - D^2}{\sqrt{(\langle K_U, K_U \rangle - D^2)(\langle L_U, L_U \rangle - D^2)}}. \tag{4}$$

Corollary 3.2. *If* $D < \sqrt{N}$, *then* $0 < \text{MKA}(K_U, L_U) < 1$.

Theorem 3.1 enables fast computation of MKA, making it more scalable (especially when combined with approximate nearest neighbor search algorithms). Few works (Chen et al., 2021; Huh et al., 2024) have considered sparsifying the kernel matrix of CKA by taking the top-k values in rows/columns. However, these works do not consider constraining rows of the kernel matrix.

4 Experiments

In this section, we empirically characterize MKA using various synthetic datasets. We compare MKA with several CKA variants with RBF kernel: 1) $\text{CKA}(\sigma = M)$: σ is set to the median, M, of the entries of the distance matrix, 2) $\text{CKA}(\sigma = \delta M)$: σ is set to δM (we mostly use $\delta = 0.45$) for considering local relationships, and 3) $t-\text{CKA}$: sparsifying the kernel matrix by considering k-nearest neighbors of each sample and setting σ to be median of the considered distances giving us a simple manifold approximation. We do not consider CKA with a linear kernel as the RBF kernel works as a good proxy of a linear one.

4.1 Equivalence of Shapes: Swiss-roll and S-curve

Here we take two shapes: Swiss-roll (Fig. 1(a)) and S-curve (Fig. 1(b), $r = 0.5$). Even though Swiss-roll and S-curve look drastically different, topologically, they both lie in a 1-D non-linear manifold and thus are equivalent. Furthermore, the parameter r in the S-curve can give it different shapes (Fig. 1(b), for details see Appendix A.3). A color map shows the correspondence among the shapes. The lower (< 0.4) and higher (> 0.6) values of r make the colors overlap, causing

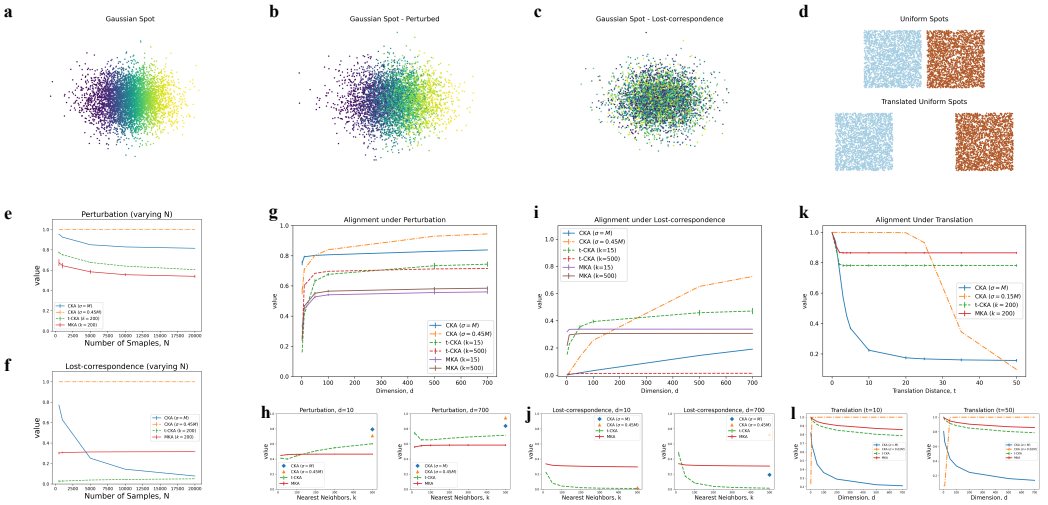

Figure 2: Characterizing MKA using synthetic datasets. (a) A Gaussian spot; colors identify the position of the points on the x-axis. (b) Perturbed Gaussian spot. We added noise to the points sampled in (a) so that colors slightly overlap. (c) A Gaussian spot with no correspondence to the spot in (a). (d) Two uniform spots are located nearby (top) and translated far away (bottom). (e) Alignment between Gaussian spot and when it is perturbed and (f) alignment between two Gaussian spots for various methods as number of samples increases ($d = 1000$). (g,h) Alignment under perturbation as (g) data dimensionality, d, and (h) hyperparameter, k, varies ($N = 5000$). (i,j) Alignment under lost-correspondence as (i) data dimensionality, d, and (j) nearest neighbors, k, varies ($N = 5000$). (k,l) Alignment between uniform spots and translated uniform spots at (k) various translation distances, t, and (l) data dimensionality ($N = 5000$; 2500 in each spot). Note that CKA does not have any notion of nearest neighbors; thus, in (h,j), we have plotted CKA values at the last point on the x-axis. Error bars are drawn up to three standard deviations (10 trials for each experiment).

the disappearance of the 1-D manifold. For experiments, we sampled 1000 points from each of the shapes and computed alignment between the Swiss-roll and the S-curve.

CKA with $\sigma = M$ fails to align the manifold of Swiss-roll and S-curve ($r = 0.5$) giving a value below 0.5 (Fig. 1(c)). However, for cases where the 1-D manifold structure is absent (e.g., $r < 0.4$ and $r > 0.6$), CKA provides a higher value. Similarly, CKA with $\sigma = 0.45M$ fails to capture this information as well and shows high alignment throughout. $t-$CKA and MKA properly capture the alignment of the two shapes. At $r = 0.5$, the alignment of Swiss-roll and S-curve is highest and gets lower as the parameter moves away from this point. However, $t-$CKA is more sensitive to the number of nearest neighbors k (Fig. 1(d)). MKA, on the other hand, is very robust to the parameter k.

4.2 Synthetic Data

In this section, we characterize the algorithms using several synthetic datasets inspired by real-world scenarios. First, we consider the alignment between a d-dimensional Gaussian spot ($x_i \sim \mathcal{N}(\mathbf{0}, I_d)$, Fig. 2(a)) and its perturbed version ($y_i = x_i + 0.5\mathcal{N}(\mathbf{0}, I_d)$, Fig. 2(b)). Such a scenario may occur when a representation learning algorithm runs repeatedly. This results in altered orders of the points in the point cloud (seen as colors slightly overlapping in Fig. 2(b)). As the number of samples in the spots increases ($d = 1000$, Fig. 2(e)), their alignment values using different methods decreases slightly. This is expected, as the denser the spot gets, the higher the change of orders within the point cloud. However, the dimensionality (d) of the data affects the values differently ($N = 5000$, Fig. 2(g)). All methods, except CKA with $\sigma = 0.45M$, are fairly consistent as d increases while the latter approaches the maximum value of 1, making it unreliable in capturing such scenarios. Additionally, $t-$CKA shows inconsistent behavior as the number of nearest neighbors

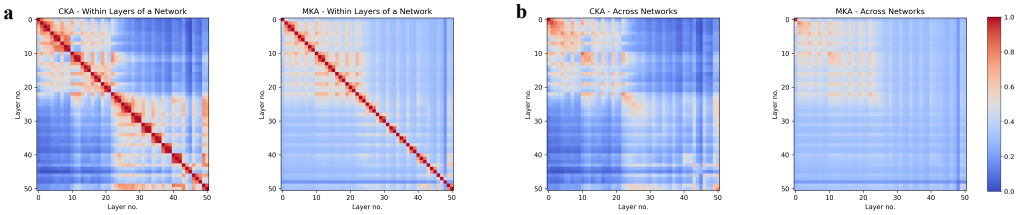

Figure 3: Alignment between features from different layers of ResNet-50 trained on the CIFAR-10 dataset. (a) Alignment between layers of a network using (left) CKA and (right) MKA. (b) Alignment between layers across different networks using (left) CKA and (right) MKA. The results are an average of 10 instances of ResNet-50 on CIFAR-10, each initialized randomly and using a subset of 5000 samples from the test set.

(k) increases, while MKA values remain consistent across a wide range (Fig. 2(h)). Overall, MKA is more restrictive to perturbations in the features than other methods.

We can take this scenario to the extreme and make the colors completely overlap each other (Fig. 2(c)). The orderings (based on some criterion) of both the Gaussian spots will not correspond to each other at all, and thus, we call it a lost-correspondence scenario. The CKA measure is sensitive to the number of samples for both choices of σ, while $t-$CKA and MKA are fairly consistent ($d = 1000$, Fig. 2(f)). The CKA measure tends to increase with higher data dimensionality, reflecting the effect of the curse of dimensionality ($N = 5000$, Fig. 2(i)). $t-$CKA and MKA, on the other hand, are fairly robust and less affected from the curse. However, $t-CKA$ is highly sensitive to the number of nearest neighbors (k) which gets resolved at a higher value of $k \geq 200$ (Fig. 2(j)). Like previously, MKA is consistent for a wide range of k, even for values smaller than 200. Overall, MKA is more consistent with varying hyperparameters than other methods.

Finally, we consider two uniform spots separated by a small distance (Fig.2(d); this scenario is inspired by Davari et al. (2023)). Both spots ($N = 2500$ each) are drawn from uniform distribution by $x_i \sim \mathcal{U}(-0.5, 0.5)$ and $y_i \sim p+\mathcal{U}(-0.5, 0.5)$ with $p = [1.1+t, 0, 0, \ldots, 0]$, where the translation distance, $t(> 0)$, controls the separation of the two spots. Regardless of the translation distance, the topology of the data remains the same, and alignment should be high. However, CKA fails to capture this phenomenon. As t increases CKA value decreases; even using smaller bandwidth $\sigma = 0.15M$ fails (Fig. 2(k)). In contrast, $t-$CKA and MKA settle to a constant and higher number as t increases. Following the results of previous experiments, we used $k = 200$. The methods provide similar results as dimensionality increases (Fig. 2(l)).

4.3 Neural Network Representations

In this section, we explore the representational similarity using ResNet-50 models trained on the CIFAR-10 dataset. First, we compute alignment between feature representations extracted from different layers (after activation) of the network to investigate how representational structure evolves across the depth of the model (Fig. 3(a)). Using CKA, we can reproduce the famous block structure (Kornblith et al., 2019; Nguyen et al., 2022). However, when MKA is used, the block structure is less pronounced in the latter layers of the network, indicating some perturbation as the data flows within the network. When features from 10 randomly initialized ResNet-50 networks are compared, this block structure becomes less pronounced for CKA and disappears in the latter layers for MKA (Fig. 3(b)). This suggests that the same network using different initialization, even with similar accuracy, can obtain different internal orientations or perturbations in the manifold.

5 Conclusions

In this paper, we introduced Manifold-approximated Kernel Alignment (MKA) and formalized and characterized it using several datasets. Here, we computed the kernel matrix and compared it to CKA (and its variations) on equal terms. By analyzing representations of neural networks, we have discovered that MKA perceives the neural network representations differently than CKA. Future

works could explore other kernel functions, e.g., effective resistance (Doyle & Snell, 1984) and diffusion distance Coifman & Lafon (2006) and focus on additional debiasing techniques (Sucholutsky et al., 2023). This alignment technique would find usage wherever alignment is beneficial, e.g., in neuroscience for monitoring brain activity, neural decoding, and brain representation analysis and graph learning for evaluating embeddings and measuring protein interactions.

ACKNOWLEDGMENTS

Mohammad Tariqul Islam is supported by MIT-Novo Nordisk Artificial Intelligence Fellowship. Special thanks Baju C. Joy and Pengrui Zhang for the discussion.

REFERENCES

Sergio A Alvarez. Gaussian RBF centered kernel alignment (CKA) in the large-bandwidth limit. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6587–6593, 2022.

Zuohui Chen, Yao Lu, J Hu, Wen Yang, Qi Xuan, Zhen Wang, and Ziaoniu Yang. Revisit similarity of neural network representations from graph perspective. *arXiv preprint arXiv:2111.11165*, 2021.

Laure Ciernik, Lorenz Linhardt, Marco Morik, Jonas Dippel, Simon Kornblith, and Lukas Muttenthaler. Training objective drives the consistency of representational similarity across datasets. *arXiv preprint arXiv:2411.05561*, 2024.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Two-stage learning kernel algorithms. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 239–246, 2010.

MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of CKA as a similarity measure in deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Peter G Doyle and J Laurie Snell. *Random walks and electric networks*, volume 22. American Mathematical Soc., 1984.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. In *Forty-first International Conference on Machine Learning*, 2024.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pp. 3519–3529. PMLR, 2019.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Alex Graeme Murphy, Joel Zylberberg, and Alona Fyshe. Correcting biased centered kernel alignment measures in biological and artificial neural networks. In *ICLR 2024 Workshop on Representational Alignment*, 2024.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. On the origins of the block structure phenomenon in neural network representations. *Transactions on Machine Learning Research*, 2022.

Cheng Soon Ong, Xavier Mary, Stéphane Canu, and Alexander J Smola. Learning with non-positive kernels. In *Proceedings of the Twenty-first International Conference on Machine learning*, pp. 81, 2004.

Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *International Conference on Learning Representations*, 2021.

Le Song, Alex Smola, Arthur Gretton, Karsten M Borgwardt, and Justin Bedo. Supervised feature selection via dependence estimation. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 823–830, 2007.

Ilia Sucholutsky, Lukas Muttenthaler, Adrian Weller, Andi Peng, Andreea Bobu, Been Kim, Bradley C Love, Erin Grant, Iris Groen, Jascha Achterberg, et al. Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

A Appendix

A.1 Proofs

(Proof of Theorem 3.1). Let $K_U H = \bar{K}$ and $L_U H = \bar{L}$. Then,

$$\begin{aligned}\bar{K}_{ij} &= K^{(U)}_{ij} - \frac{1}{N}\sum_j K^{(U)}_{ij} \\ &= K^{(U)}_{ij} - \frac{1}{N}D. \end{aligned} \tag{5}$$

Now we can compute the inner product,

$$\begin{aligned}\langle \bar{K}, \bar{K} \rangle &= \sum_{i,j} (K^{(U)}_{ij} - \frac{1}{N}D)^2 \\ &= \sum_{i,j} \left(\left(K^{(U)}_{ij}\right)^2 - \frac{2}{N} D K^{(U)}_{ij} + \frac{1}{N^2}D^2 \right) \\ &= \sum_{i,j} \left(K^{(U)}_{ij}\right)^2 - \frac{2}{N} D \sum_{i,j} K^{(U)}_{ij} + \frac{1}{N^2}D^2 \sum_{i,j} 1 \\ &= \sum_{i,j} \left(K^{(U)}_{ij}\right)^2 - D^2 \\ &= \langle K_U, K_U \rangle - D^2 \end{aligned} \tag{6}$$

We used the fact that $\sum_{i,j} K^{(U)}_{ij} = ND$ and $\sum_{i,j} 1 = N^2$. Similarly, $\bar{L}_{ij} = L^{(U)}_{ij} - \frac{1}{N}D$ and $\langle \bar{L}, \bar{L} \rangle = \langle L_U, L_U \rangle - D^2$. Finally,

$$\begin{aligned}\langle \bar{K}, \bar{L} \rangle &= \sum_{i,j} (K^{(U)}_{ij} - \frac{1}{N}D)(L^{(U)}_{ij} - \frac{1}{N}D) \\ &= \sum_{i,j} K^{(U)}_{ij} L^{(U)}_{ij} - \frac{1}{N} D (K^{(U)}_{ij} + L^{(U)}_{ij}) - \frac{1}{N^2}D^2 \\ &= \sum_{i,j} K^{(U)}_{ij} L^{(U)}_{ij} - D^2 \\ &= \langle K^{(U)}_{ij}, L^{(U)}_{ij} \rangle - D^2 \end{aligned} \tag{7}$$

□

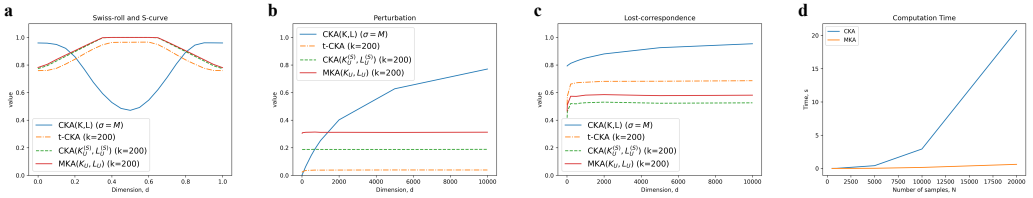

Figure 4: Effect of Kernel Approximation on the CKA algorithm. (a) Alignment between Swiss-roll and S-curve. (b,c) Gaussian spots under (b) perturbation and (c) Lost-correspondence. CKA with manifold approximation ($\text{CKA}(K_U^{(S)}, K_L^{(S)})$ behave similar to MKA, but with less bias. (d) Computation time for CKA and MKA. MKA require much less time than CKA (average of 5 runs). Note that we have excluded the computation time for the kernel matrix.

(Proof of Corollary 3.2). We start from the inner products,

$$\begin{aligned}\langle K_U, K_U \rangle - D^2 &= \sum_{i,j} \left(K_{ij}^{(U)}\right)^2 - D^2 \\ &= \sum_{i,i} 1 + \sum_{i,j,i\neq j} \left(K_{ij}^{(U)}\right)^2 - D^2 \\ &= N - D^2 + \sum_{i,j,i\neq j} \left(K_{ij}^{(U)}\right)^2. \end{aligned} \tag{8}$$

Similarly,

$$\langle L_U, L_U \rangle - D^2 = N - D^2 + \sum_{i,j,i\neq j} \left(L_{ij}^{(U)}\right)^2 \tag{9}$$

And finally,

$$\langle K_U, L_U \rangle - D^2 = N - D^2 + \sum_{i,j,i\neq j} K_{ij}^{(U)} L_{ij}^{(U)} \tag{10}$$

The value $\sum_{i,j,i\neq j} K_{ij}^{(U)} L_{ij}^{(U)}$ can be zero if the nearest neighbors in the kernels do not overlap each other. Otherwise, this value is positive. Thus, the lower bound is guaranteed when $N > D^2$. The upper bound is due to Cauchy–Schwarz inequality. □

A.2 CKA with Manifold Approximation

We can symmetrize the manifold approximated kernel matrix, K_U, using the probabilistic t-conorm given by

$$K_U^{(S)} = K_U + K_U^T - K_U \circ K_U^T, \tag{11}$$

where $\circ$ denotes element-wise multiplication. This operation does not guarantee a positive semidefinite kernel. However, we can now directly apply CKA on the approximated kernels $K_U^{(S)}$ and $L_U^{(S)}$. The CKA results obtained from this kernel matrix behave similarly to that of MKA but with less bias (Fig. 4(b-c)). However, computing MKA requires much less time compared to CKA (Fig. 4(d), using `NumPy` Harris et al. (2020)).

A.3 Details of Swiss-roll and S-Curve

Swiss-roll and S-curve are parameterized by variable $t \in [0, 1]$. S-curve contains an additional control parameter $r \in [0, 1]$ that determines the shape. $r = 0.5$ gives the familiar S-curve used in

many studies. We only consider 2-D shapes in this study.

Swiss-Roll:

$$z = \frac{3\pi}{2}(1 + 2t) \tag{12}$$

$$x_1 = z\cos(z) \tag{13}$$

$$x_2 = z\sin(z) \tag{14}$$

S-Curve:

$$z = 3\pi(t - r) \tag{15}$$

$$y_1 = \sin(z) \tag{16}$$

$$y_2 = \mathrm{sgn}(z)(\cos(z) - 1) \tag{17}$$