Efficient Statistical Tests: A Neural Tangent Kernel Approach

Sheng Jia [1 2] **Ehsan Nezhadarya** [3] **Yuhuai Wu** [1 2] **Jimmy Ba** [1 2]

Abstract

For machine learning models to make reliable predictions in deployment, one needs to ensure the previously unknown test samples need to be sufficiently similar to the training data. The commonly used shift-invariant kernels do not have the compositionality and fail to capture invariances in high-dimensional data in computer vision. We propose a *shift-invariant convolutional neural tangent kernel* (SCNTK) based outlier detector and two-sample tests with maximum mean discrepancy (MMD) that is $\mathcal{O}(n)$ in the number of samples due to using the random feature approximation. On MNIST and CIFAR10 with various types of dataset shifts, we empirically show that statistical tests with such compositional kernels, inherited from infinitely wide neural networks, achieve higher detection accuracy than existing non-parametric methods. Our method also provides a competitive alternative to adaptive kernel methods that require a training phase.[1]

1. Introduction

In the two-sample hypothesis testing task, given two sets of sampled data, we are interested in knowing whether they come from the same distribution. In the extreme case when either set has a sample size of 1, such testing can be regarded as *out-of-distribution* (OOD) detection. For two-sample tests, one of the simplest methods is to evaluate the *maximum mean discrepancy* (MMD) statistics (Gretton et al., 2012). MMD measures the distance between the kernel mean embeddings of two distributions (Muandet et al., 2016) in the reproducing kernel hilbert space (RKHS) (Berlinet & Thomas-Agnan, 2004).

As the choice of kernel and associated hyperparameters can largely affect the efficacy of the MMD-based method, there have been methods that optimize the kernel for maximizing the test power (Liu et al., 2020; Gao et al., 2020). These methods have been shown to empirically work well for image data. Since the distributions of high-dimensional data are expected to have complex function form, deep kernel based methods are becoming popular (Wilson et al., 2016; Sutherland et al., 2016; Li et al., 2017; Jean et al., 2018; Wenliang et al., 2019). A common deep kernel approach is to train a network to extract features under different criteria, such as the test power maximization. Extracted features are then fed into a simple Gaussian kernel (Liu et al., 2020; Gao et al., 2020).

One disadvantage of the deep kernel approach is that the method hinges on the generalization of the feature extractor to the new "test" data. Using a portion of the test samples as "training" data (or for fine-tuning) also means that less "test" samples become available. Besides, in online scenarios, nonparametric methods that optimize kernel parameters are not computationally scalable. To address these issues, we investigate an orthogonal direction to see whether the compositional kernels associated with randomly initialized convolutional neural networks (CNNs) provide a better inductive bias than existing kernels for statistical tests in image domains. The hope is that fast testing with their random feature approximation (Rahimi & Recht, 2008; Cho & Saul, 2009) is competitive to deep kernel methods that use naive Gaussian kernels. Our motivation comes from the recent surge of interests in neural tangent kernel (Jacot et al., 2018; Lee et al., 2019; Bietti & Mairal, 2019) and its empirical success in CIFAR10 classification task (Arora et al., 2019; Li et al., 2019). We summarize our contributions below:

1. We show the explicit conditions under which the neural tangent kernel for certain fully-connected and convolutional neural networks are shift-invariant and characteristic, which makes it a valid kernel for two-sample tests with MMD statistics. Shift-invariance also makes the kernel applicable for kernel density estimation (KDE)-based out-of-distribution detection.
2. We show how the proposed *shift- invariant convolutional neural tangent kernel* (SCNTK) is used to compute the MMD statistic for two-sample tests and the

[1]University of Toronto [2]Vector Institute [3]LG Electronics. Correspondence to: Sheng Jia <sheng@cs.toronto.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]Source code https://github.com/Sheng-J/scntk

outlier score for OOD detection. By approximating the exact tangent kernel using the linear kernel of gradients, our computational cost is reduced to linear $\mathcal{O}(n)$ from $\mathcal{O}(n^2)$ w.r.t to the sample size.

3. We empirically show that the SCNTK-based nonparametric methods outperform kernel methods with fixed Gaussian kernels in two-sample tests on CIFAR10 and MNIST. It also provides a competitive and efficient alternative to the deep kernel methods that optimize the kernel.

2. Related work

MMD-based Two Sample Test: Recent works with deep kernel approaches have pointed out the limited representation power of the naive Gaussian kernel in pixel space (Wenliang et al., 2019; Kirchler et al., 2020). Optimizing the bandwidth of such a Gaussian kernel is also insufficient to detect the distributional difference between two datasets for image domains and complex distributions (Liu et al., 2020). As such, a feature extractor that extracts meaningful semantic information is trained by maximizing the test power on held-out samples. Since some data from $S_{\mathbb{Q}}$ are adversarially created using samples from $\mathbb{P}$, Gao et al. (2020) notices that these adversarial data are non-independent, which causes issues in the MMD computation. Hence, they apply wild bootstrap process to achieve a better result in similar domains.

However, all these methods still rely on training the feature extrator network on the held-out data. This makes the learning-based methods potentially very expensive in online scenarios where the distribution of the data may change over time. As such, we are seeking an alternative fixed kernel approach that could improve upon the survey results with existing fixed kernel methods in Rabanser et al. (2019).

Out-of-Distribution Detection: There have been many outlier detection methods proposed under different problem setups. Specifically, when the label information about the data is given, Hendrycks & Gimpel (2016) uses the confidence score of a softmax classifier trained on the labelled data as the outlier score. Lee et al. (2017b); Liang et al. (2017) further use temperature scaling and adversarial noise. Another improvement to the softmax confidence-based outlier detector is to use the variance of an ensemble model (Choi et al., 2018).

Fourier Feature Network: Motivated by Rahimi & Recht (2008), which shows that simple shift-invariant kernels such as Gaussian can be approximated with random fourier features, we also use the *periodic activation* to have the shift-invariant property for NTK. Tancik et al. (2020) shows that such a Fourier feature mapping could allow a fully-connected network to learn high-frequency functions in low dimensional domains. Sitzmann et al. (2020) also shows that sinusoidal representation networks are good fits for representing complex natural signals and their derivatives. In our work, we take a further step to look at the effect of using such a periodic activation for a highly composite neural tangent kernel associated with a convolutional network.

3. Technical Background

3.1. Two-sample hypothesis testing

Let $\mathbb{P}$ and $\mathbb{Q}$ be the underlying distributions for two sets of samples $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$. Our task is to determine whether $\mathbb{P} = \mathbb{Q}$. Under the framework of hypothesis testing, we have the null hypothesis $h_0 : \mathbb{P} = \mathbb{Q}$ and alternative hypothesis $h_1 : \mathbb{P} \neq \mathbb{Q}$. In this work, permutation tests (Dwass, 1957; Fernández et al., 2008) with MMD statistics is carried out to estimate the sampling distribution of the MMD statistics under null hypothesis. Null hypothesis is rejected if the the prior observed result is too extreme under the estimated sampling distribution.

Maximum mean discrepancy (MMD). Assume a kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ has its associated reproducing kernel Hilbert space $\mathcal{H}_K$. The feature map from $\mathcal{X}$ to $\mathcal{H}_K$ is $K(\cdot, \mathbf{x}) \in \mathcal{H}_k$. One common test statistics for two-sample tests is the MMD metric, which measures the distance between two distributions $\mathbb{P}, \mathbb{Q}$ using their embeddings in RKHS (Berlinet & Thomas-Agnan, 2004). Let $\mathbf{x}, \mathbf{x}' \sim \mathbb{P}$ and $\mathbf{y}, \mathbf{y}' \sim \mathbb{Q}$, their kernel mean embeddings are $\mu_{\mathbb{P}} := \mathbb{E}[K(\cdot, \mathbf{x})]$, $\mu_{\mathbb{Q}} := \mathbb{E}[K(\cdot, \mathbf{y})]$. Then MMD under this kernel $K(\cdot, \cdot)$ is the RKHS norm of two embeddings:

$$\text{MMD}(\mathbb{P}, \mathbb{Q}; \mathcal{H}_K) = \|\mu_{\mathbb{P}} - \mu_{\mathbb{Q}}\|_{\mathcal{H}_K} \tag{3.1}$$

MMD expression can be simplified into the following form:

$$\text{MMD}^2(\mathbb{P}, \mathbb{Q}) = \mathbb{E}\big[K(\mathbf{x}, \mathbf{x}') + K(\mathbf{y}, \mathbf{y}') - 2K(\mathbf{x}, \mathbf{y})\big] \tag{3.2}$$

Then the unbiased estimate of MMD^2, $\widehat{\text{MMD}}^2_u$, can be computed as in Equation 3.3 using m samples $\{\mathbf{x}_1, ..., \mathbf{x}_n\} \sim \mathbb{P}$ and n samples $\{\mathbf{y}_1, ..., \mathbf{y}_n\} \sim \mathbb{Q}$. The time complexity of computing this estimate is $\mathcal{O}(mn)$, or simply $\mathcal{O}(n^2)$ if $m = n$. It is quadratic w.r.t the number of test samples due to the pairwise kernel evaluations:

$$\widehat{\text{MMD}}^2_u = \frac{1}{m^2 - m} a + \frac{1}{n^2 - n} b - \frac{2}{m(n-1)} c \tag{3.3}$$

$$a = \sum_{i=1}^{m} \sum_{j \neq i}^{m} K(\mathbf{x}_i, \mathbf{x}_j) \quad b = \sum_{i=1}^{n} \sum_{j \neq i}^{n} K(\mathbf{y}_i, \mathbf{y}_j) \tag{3.4}$$

and $c = \sum_{i=1}^{m} \sum_{j \neq i}^{n} K(\mathbf{x}_i, \mathbf{y}_j)$. For MMD to be a proper metric, $\text{MMD}(\mathbb{P}, \mathbb{Q}; \mathcal{H}_K) = 0$ if and only if $\mathbb{P} = \mathbb{Q}$, the kernel $K(\cdot, \cdot)$ needs to be characteristic. The most common choice is the naive gaussian kernel. In this work, we review the kernels that operate on learned features and propose our fixed shift-invariant kernel derived from randomly initialized neural networks.

Permutation tests and asymptotic distributions of MMD statistics. Based on the asymptotic distributions of $\widehat{\text{MMD}}_u$, a permutation test under the assumption of $h_0 : \mathbb{P} = \mathbb{Q}$ is carried out to obtain the p-value for deciding whether to accept the null hypothesis. Under the null hypothesis $h_0 : \mathbb{P} = \mathbb{Q}$, this MMD statistic converges to a chi-squared distribution:

$$n\widehat{\text{MMD}}_u^2 \xrightarrow{d} \sum_i \lambda_i (Z_i^2 - 2); \quad Z_i \sim \mathsf{N}(0, 2) \tag{3.5}$$

where λ_i are the eigenvalues in the eigen-value equation for the centered kernel (Gretton et al., 2012). Under the alternative hypothesis $h_1 : \mathbb{P} \neq \mathbb{Q}$, it converges in distribution to a Gaussian: (Serfling, 1980).

$$\sqrt{n}(\widehat{\text{MMD}}_u^2 - \text{MMD}^2) \xrightarrow{d} \mathsf{N}(0, \sigma_{h_1}^2) \tag{3.6}$$

where the variance is governed by

$$\sigma_{h_1}^2 := 4\Big(\mathbb{E}[H_{12}H_{13}] - \mathbb{E}[H_{12}]^2\Big) \tag{3.7}$$

$$H_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) + K(\mathbf{y}_i, \mathbf{y}_j) - K(\mathbf{x}_i, \mathbf{y}_j) - K(\mathbf{y}_i, \mathbf{x}_j)$$

Under the null hypothesis, samples in $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$ can be shuffled k times to recompute test statistics $\text{MMD}_{u,i}^2$ for $i = 1, ..., k$. These statistics are used to estimate the null distribution in Equation 3.5. We use this sampling distribution to determine whether the observed statistic $\text{MMD}_{u,0}^2$, which is computed with the original unshuffled samples, is too large to be outside the $1 - \alpha$ quantile. α is generally chosen 0.05. Null hypothesis h_0 is rejected if this is the case; p-value $< \alpha$.

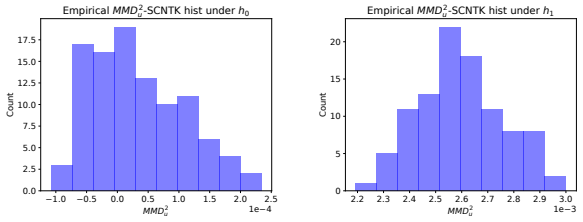

Figure 1. Both histograms are plotted from 100 MMD statistics, each of which is computed using 40 samples from each $\mathbb{P}$ and $\mathbb{Q}$ distributions. The dimension is set to $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{8\times8\times1}$ **Left:** Empirical distribution of MMD_u^2 under h_0, with $\mathbb{P}$ and $\mathbb{Q}$ Gaussians with $\sigma^2 = 5$, using 100 samples from each. **Right:** Empirical distribution of MMD_u^2 under h_1, with $\mathbb{P} : \mathsf{N}(\mathbf{0}, 5\mathbf{I})$ and $\mathbb{Q} : \mathsf{N}(\mathbf{0}, 10\mathbf{I})$. A simple three-layer 2-strided convolution network with a filter size of 2 and widths of 100 is used to derive the SCNTK similar to Equation 4.6 for MMD.

Deep Kernel method. Besides using the naive gaussian kernels for MMD, we also investigate recently proposed baselines that optimize the deep feature extraction network to maximize the test power Liu et al. (2020); Gao et al. (2020). Gaussian kernel is applied to the extracted features $\boldsymbol{K}(\phi(\mathbf{x}), \phi(\mathbf{x}'))$ as in Equation 3.8. $\phi_w(\cdot)$ is the parametric feature extractor optimized for testing power on held-out dataset. To ensure the characteristic property of the kernel, a naive gaussian kernel $q(\mathbf{x}, \mathbf{x}')$ in the pixel space is multiplied. ϵ is another tunable parameter.

$$\boldsymbol{K}_w(\mathbf{x}, \mathbf{x}') = \big[(1-\epsilon)\boldsymbol{K}\big(\phi_w(\mathbf{x}), \phi_w(\mathbf{x}')\big) + \epsilon\big] q(\mathbf{x}, \mathbf{x}') \tag{3.8}$$

3.2. Neural tangent kernel

First, we briefly describe the standard NTK formulation and the kernels associated with different neural network architectures. We denote the scalar output of a randomly initialized neural network by $f(\mathbf{x}; \boldsymbol{\theta}_0) \in \mathbb{R}$ where $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ is all the parameters initialized under the standard normal distribution. $\mathbf{x}$ is the input and we are interested in the image data $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$. Consider two image data $\mathbf{x}$ and $\mathbf{x}'$, the empirical neural tangent kernel is given by:

$$K(\mathbf{x}, \mathbf{x}') = \left\langle \frac{\partial f(\mathbf{x}, \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0}, \frac{\partial f(\mathbf{x}', \boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}_0} \right\rangle \tag{3.9}$$

In the large width limit, NTK under a L-layer fully-connected network will converge to the deterministic form in Equation 3.10 (Arora et al., 2019). It is a sum of products of covariance and *derivative covariance* where each $\Sigma^{(l-1)}$ is the covariance function of the i.i.d gaussiance process defined for the neuron at l-th layer pre-activation $f^{(l)}(\mathbf{x})$. $\dot{\Sigma}^{(l)}$ is the *derivative covariance*, which can be computed similarly to $\Sigma^{(l)}$ except replacing the previous layer activation function with its derivative.

$$K(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{L+1} \left(\Sigma^{(l-1)}(\mathbf{x}, \mathbf{x}') \prod_{l'=l}^{L+1} \dot{\Sigma}^{(l')}(\mathbf{x}, \mathbf{x}') \right) \tag{3.10}$$

Arora et al. (2019) also shows the deterministic form of convolutional NTK (CNTK) similar to Equation 3.10 under a L-layer convolutional network. It also has the sum of products of covariance form as shown in the Appendix.

3.3. Characteristic Kernel

For the deterministic form of NTK such as Equation 3.10 to be characteristic, we review the result indicating that a characteristic kernel can be constructed by adding or multiplying positive semi-definite shift-invariant kernels to a characteristic kernel (Sriperumbudur et al., 2010).

Definition 1 (Positive definite function). *A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called positive definite (PD) if, for all $n \in \mathbb{N}$, $\alpha_1, ..., \alpha_n \in \mathbb{R}$ and all $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathcal{X}$,*

$$\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0 \tag{3.11}$$

that is, a positive definite function has a positive semi-definite matrix of size n for all possible n and function arguments $\mathbf{x}_1, ..., \mathbf{x}_n$.

Definition 2 (Characteristic kernel). *A bounded positive definite kernel K is characteristic if the following map is $1-1$; that is, embedding the probability distribution $\mathbb{P}$ into the RKHS $\mathcal{H}_K$ as a feature map $\mu_{\mathbb{P}}$ is injective:*

$$\mathbb{P} \to \mu_{\mathbb{P}} = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}}[K(\cdot, \mathbf{x})] = \int_{\mathcal{X}} K(\cdot, \mathbf{x})\, d\mathbb{P}(\mathbf{x}) \quad (3.12)$$

Under a characteristic kernel, this definition will naturally make MMD a valid metric for two-sample tests, evaluating $\mathbb{P} = \mathbb{Q}$ or $\mathbb{P} \neq \mathbb{Q}$, since MMD in Equation 3.1 is the distance between embeddings $\mu_{\mathbb{P}}$ and $\mu_{\mathbb{Q}}$ in RKHS $\mathcal{H}_K$ space. If MMD $= 0$, $\mu_{\mathbb{P}} = \mu_{\mathbb{Q}}$ and therefore we could say $\mathbb{P} = \mathbb{Q}$.

Using Bochner's theorem (Rudin, 1962), Sriperumbudur et al. (2010) has shown gaussian kernel to be characteristic. The next theorem shows how to construct another characteristic kernel from a simple characteristic kernel such as the gaussian kernel.

Compositionality Theorem 1 (Sriperumbudur et al. (2010))**.** *Let K, K_1, K_2 be shift-invariant kernels that can be expressed as $K(\mathbf{x}, \mathbf{y}) = \Psi(\mathbf{x} - \mathbf{y})$ where $\Psi(\cdot)$ is a bounded continuous real-valued positive definite function on $\mathbb{R}^d$. Suppose K is characteristic and $K_2 \neq 0$ Then $K + K_1$ and $K \cdot K_2$ are characteristic.*

This will become the key theorem in showing the characteristic property of certain NTKs. Previous deep kernel methods in Equation 3.8 also rely on this result for the characteristic property of the kernel. The main difference between their methods and our fixed NTK-based kernel is that this gaussian kernel in pixel space, $q(\mathbf{x}, \mathbf{x}')$, is manually multiplied in Equation 3.8. For the NTK kernel, we will see in Equation 3.10 that this gaussian kernel could naturally appear in one of the terms under the cosine activation though other terms still make this kernel expressive.

When the data is restricted to the hypersphere $\mathbb{S}^{d-1}$, Geifman et al. (2020) has shown that plain NTK for MLP with ReLU activations includes the same sets of functions as the Laplace kernel in their RKHS. As the Laplace kernel is universal, this makes NTK universal and characteristic. Since we focus on general cases, shift-invariance is a simple sufficient condition for NTK to be characteristic. We further discuss conditions under which this property makes CNTK characteristic.

3.4. Out-of-distribution (OOD) detection.

When the sample size of $S_{\mathbb{Q}}$ becomes 1, two-sample tests can be regarded as outlier detections. The simplest approach is to use kernel density estimation (KDE). Outlier can be detected by thresholding the negative of the density as the outlier score. For this paper, we use this task to perform simple ablation studies on our proposed kernel. A popular non-thresholding metric for comparing different methods is to use the area under ROC (AUROC).

4. Shift-invariant CNTK

The goal of our proposed shift-invariant convolutional neural tangent kernel (SCNTK) is to provide a robust non-parametric approach to two-sample tests and outlier detection that ultimately benefits from the recent advances in deep learning. Although neural tangent kernel (NTK) has shown to provide competitive performance to trained neural networks in classification tasks (Arora et al., 2019), the standard NTKs with ReLU activations are not shift-invariant in the infinite width limit. In this section, we extend the vanilla neural tangent kernel (NTK) to satisfy the shift-invariance property in the infinite width limit. We use K_{sc} to denote SCNTK and K_s to denote shift-invariant NTK for a fully-connected network.

4.1. Motivations for the shift-invariance

1) In section 4.4 and the appendix, we rely on the shift-invariance to show NTK is characteristic in our particular settings. **2)** When the kernel is optimized, Liu et al. (2020) has shown the flexibility of not being shift-invariant allows it to adapt to the local structure of data distributions. However, our fixed kernel cannot take this advantage. Instead, a fixed kernel method can use more samples in the testing process by incorporating the unused "training" data. In this scenario, we consider the shift-invariance to be a particular prior, or a regularization scheme. In OOD applications, it is desirable to have the same kernel evaluation when the same shifts are applied to the training and test example $\mathbf{x}$, $\mathbf{x}'$.

4.2. Shift-invariant convolutional neural tangent kernel (SCNTK) for statistical tests

Two-sample tests. We perform the two-sample hypothesis testings with MMD statistics under our shift-invariant convolutional neural tangent kernel (SCNTK). The computational benefit of using such a linear kernel in Equation 3.9 is that we can sum up the gradients $\phi(\mathbf{x}_i), \phi(\mathbf{y}_i)$ for $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$ separately, reducing the MMD computation or KDE evaluations to $\mathcal{O}(n)$ from $\mathcal{O}(n^2)$. Other exact kernel methods require $\mathcal{O}(n^2)$ due to the pairwise kernel evaluations in

Equation 3.3.

$$\widehat{\text{MMD}}_u^2 = \frac{1}{m^2-m}a + \frac{1}{n^2-n}b - \frac{2}{m(n-1)}c \tag{4.1}$$

$$a = \left(\left\| \sum_{i=1}^{m} \phi(\mathbf{x}_i) \right\|_2^2 - \sum_{i=1}^{m} \|\phi(\mathbf{x}_i)\|_2^2 \right) \tag{4.2}$$

$$b = \left(\left\| \sum_{i=1}^{n} \phi(\mathbf{y}_i) \right\|_2^2 - \sum_{i=1}^{n} \|\phi(\mathbf{y}_i)\|_2^2 \right) \tag{4.3}$$

$$c = \left(\langle \sum_{i=1}^{m} \phi(\mathbf{x}_i), \sum_{i=1}^{n} \phi(\mathbf{y}_i) \rangle - \sum_{i=1}^{n} \langle \phi(\mathbf{x}_i), \phi(\mathbf{y}_i) \rangle \right) \tag{4.4}$$

OOD detection. For a test data $\mathbf{x}'$, the outlier score is computed using the negative of the sum of shift-invariant kernel evaluations at each of the inlier samples $\mathcal{D}_\mathbf{x}$:

$$S(\mathbf{x}'; \mathcal{D}_\mathbf{x}) = -\sum_{i=1}^{|\mathcal{D}_\mathbf{x}|} K_{sc}(\mathbf{x}', \mathbf{x}_i) \tag{4.5}$$

SCNTK. To make the NTK kernel shift-invariant for a convolutional network, we do not include the inner products of the gradients w.r.t the first layer, as in in Equation 4.6. We discuss about this choice in the subsection 4.3.

$$K_{sc}(\mathbf{x}, \mathbf{x}') = \sum_{l=2}^{L} \sum_{\beta=1}^{C^{(\beta)}} \left\langle \frac{\partial f(\mathbf{x}, \boldsymbol{\theta}_0)}{\partial \boldsymbol{W}_{(\beta)}^{(l)}}, \frac{\partial f(\mathbf{x}', \boldsymbol{\theta}_0)}{\partial \boldsymbol{W}_{(\beta)}^{(l)}} \right\rangle \tag{4.6}$$

$\boldsymbol{W}_{(\beta)}^{(l)}$ is the weight matrix associated with the l-th layer and the β-th output channel. The first activation layer $\boldsymbol{h}^{(1)}(\mathbf{x})$ uses the cosine activation and the uniform bias $\boldsymbol{w_0} \sim Unif(\mathbf{0}, 2\pi\mathbf{I})$

$$\boldsymbol{h}_{(\beta)}^{(1)}(\mathbf{x}) = \mu_0 \cos\left(\sum_{\alpha=1}^{C^{(0)}} \boldsymbol{W}_{(\alpha),(\beta)}^{(l)} * \boldsymbol{h}_{(\alpha)}^{(l-1)}(\mathbf{x}) + \boldsymbol{w}_0 \right) \tag{4.7}$$

where α is the input channel index. The constant is $\mu_0 = \left(c_\sigma / \left(C^{(1)} \times q \times q\right)\right)^{1/2}$ with filter size q and c_σ is the standard deviation generally used for the weight initialization.

Figure 1 shows the empirical distribution of MMD_u^2 under SCNTK using the synthetic toy $S_\mathbb{P}$, $S_\mathbb{Q}$ data generated from simple Gaussians, similar to the setup in Gretton et al. (2007). Empirically, we see that MMD_u^2 with SCNTK could still behave as a chi-squared distribution in Equation 3.5 under $h_0 : \mathbb{P} = \mathbb{Q}$. And it behaves as a normal distribution 3.6 under $h_1 : \mathbb{P} \neq \mathbb{Q}$.

4.3. Shift-invariant property

To show that SCNTK has a shift-invariant property, we firstly establish results associated with the covaraince $\Sigma^{(l-1)}(\cdot)$ and derivative covariance $\dot{\Sigma}^{l-1}(\cdot)$ at each l-th pre-activation layer. One important feature is that we only need to use cosine activations for the first layer for shift-invariance but choose to keep the rest of activations as ReLU. This is motivated by the fact that most neural network architectures are designed and tested under the commonly used ReLU activations. Firstly, we note that both fully-connected and CNNs have shift-invariant covariance matrices as long as the first pre-activation covariance is shift-invariant. See the Appendix for more detailed discussions on these results.

Lemma 1. *If the covariance $\Sigma^{(1)}(\mathbf{x}, \mathbf{x}')$ associated with the second pre-activation layer is shift-invariant for a fully-connected network or a CNN, then the activation covariances $\Sigma^{(l)}(\mathbf{x}, \mathbf{x}')$ and the derivative covariances $\dot{\Sigma}^{(l)}(\mathbf{x}, \mathbf{x}')$ $\forall l = 2, ..., L$ are shift-invariant.*

To ensure the first-layer covariance is shift-invariant, we make use of the random feature approximations (Rahimi & Recht, 2008) with the cosine activation and the uniform bias.

Proposition 1. *In the infinite width limit, CNTK computed by the inner products of gradients w.r.t parameters for $i = 2, ..., L$ layers under first-layer cos activations is shift-invariant; $K_s(\mathbf{x}, \mathbf{x}') = K_s(\mathbf{x} - \mathbf{x}'), K_{sc}(\mathbf{x}, \mathbf{x}') = K_{sc}(\mathbf{x} - \mathbf{x}')$ if the first-layer activation $\sigma(\cdot)$ is $\sin(\cdot)$ or $\cos(\cdot)$ and the bias is uniformly sampled $w_0 \sim Unif(0, 2\pi)$*

We show that first-layer cosine activation with a uniform bias in equation 4.7 will lead to a sum of patch-wise Gaussian kernels for the covariance matrix of the second layer pre-activation $\Sigma^{(1)}(\mathbf{x}, \mathbf{x}')$. that is shift-invariant. Thus, we can use Lemma 1 to show the shift-invariance property of SCNTK K_{sc}.

4.4. Characteristic property

We first show that SNTK under a fully-connected network in equation 3.10 is a characterstic kernel. Note that it is a sum and products of shift-invariant kernels as below:

$$K_s = \underbrace{c_\sigma \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|_2^2}{2} \right)}_{\text{① characteristic}} \prod_{l'=2}^{L+1} \underbrace{\dot{\Sigma}^{(l')}(\mathbf{x}, \mathbf{x}')}_{\text{② shift-inv}} + b \tag{4.8}$$

$$b = \sum_{l=3}^{L+1} \left(\underbrace{\Sigma^{(l-1)}(\mathbf{x}, \mathbf{x}')}_{\text{③ shift-inv}} \prod_{l'=l}^{L+1} \underbrace{\dot{\Sigma}^{(l')}(\mathbf{x}, \mathbf{x}')}_{\text{② shift-inv}} \right) \tag{4.9}$$

Since we use the cosine activation for the first layer, we explicitly write out $\Sigma^1(\mathbf{x}, \mathbf{x}')$ as a gaussian kernel. By our lemma 1, we know ②, ③ are shift-invariant kernels. With ReLU activations for the rest of layers, all the other covariances give an arc-cos kernel. An arc-cos kernel is a valid kernel that gives a positive semi-definite kernel matrix (Cho, 2012). In addition, gaussian kernel is characteristic

(Sriperumbudur et al., 2010). Hence, SNTK K_s is a sum and products of shift-invariant kernel with a characteristic kernel ①. By Compositionality Theorem 1, K_s is characteristic.

For SCNTK K_{sc}, we could use similar arguments for its deterministic form in the large width limit. It also has the sum of products of covariance and derivative covariance, similar to the form in Equation 4.8. At a very high-level, K_{sc} can be factored into a sum of patch-wise kernels

$$K_{sc}(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{r} K_s(\mathbf{x}_p, \mathbf{x}'_p) \tag{4.10}$$

where we argue that each $K_s(\mathbf{x}_p, \mathbf{x}'_p)$ is a characterstic kernel for the patch p of the input image. p is the patch index and r is the total number of patches. Each $K_s(\mathbf{x}_p, \mathbf{x}'_p)$ has the patch-wise gaussian kernel term $K^{(1)}(\mathbf{x}_p, \mathbf{x}'_p) = \exp\left(-\left\|[\mathbf{x}]_p - [\mathbf{x}']_p\right\|_2^2/2\right)$ for the image patch p. For a one-layer convolutional network with a fully-connected output layer, the output GP covariance kernel is the sum of these patch-wise gaussian kernels so MMD^2 under such a kernel can be factored into a sum of *patch-wise* MMDs: $\text{MMD}^2(\mathbf{x}, \mathbf{x}') = \sum_{p=1}^{r} \text{MMD}^2(\mathbf{x}_p, \mathbf{x}'_p)$. Since MMD^2 is non-negative, we could see that the sum of *patch-wise* characteristic kernels can also be characterstic. We leave the detailed discussions for the characterstic property of SCNTK in the Appendix.

5. Experiments

Our main experiments for two-sample tests are carried out on two sets of samples with $S_{\mathbb{P}}$ from MNIST (LeCun et al., 1998) or CIFAR10 (Krizhevsky et al., 2009) datasets and $S_{\mathbb{Q}}$ as used in two prior works (Liu et al., 2020; Rabanser et al., 2019). We compare our method, which we call MMD-SCNTK, against the recently proposed two-sample test methods from these two prior works in their image domains. Toy experiments, using Blob (Liu et al., 2020) and Higgs (Baldi et al., 2014) datasets, are included in the appendix. We also empirically investigate the effect of increasing the network width. Ablation studies are accompanied to observe any performance gap between using finite-width approximations and using the exact NTK. Finally, we present preliminary studies of using NTK in OOD detection.

Network architecture for SCNTK. We aim to be consistent with the network architectures used in previous works. Two main differences are: (1) we replace the relu activations of the first layer with cosine activations; (2) To have our NTK kernel closer to its deterministic form, we use a width of 300, which is wider than 32 and 64 used in those baselines. Further details are provided in the Appendix. To compare with the dimensionality reduced methods (Rabanser et al., 2019) in Figure 2, we use the same three-layer convolutional network architecture. For the comparison with optimized kernel methods (Liu et al., 2020) in Table 2 and with different widths in Table 1, we use the same four layer network with strided convolutions without max-pooling. This is similar to the discriminator of the DCGAN architecture used in Radford et al. (2015). SCNTK kernel is approximated using the inner products between the gradients of the network output w.r.t parameters for two data points $\mathbf{x}, \mathbf{x}'$.

5.1. Two-sample test comparisons with methods using the dimensionality reduction and a fixed kernel

For each MNIST and CIFAR10 dataset, we randomly sample a subset $S_{\mathbb{P}}$ of size n. We also sample n data points $S_{\mathbb{Q}}$ from the same dataset and apply various types of data shifts such as Gaussian noise and pixel shifts. We evaluate the detection performance of dataset shift using MMD-SCNTK against the state-of-the-art MMD-based method in (Rabanser et al., 2019) with a fixed kernel for this task. The baseline MMD method applies the Gaussian kernel to the output feature of the encoder of an untrained randomly initialized autoencoder (UAE). Under the experiment setup by (Rabanser et al., 2019), we experimented with different number of samples $\mathbf{x} \sim \mathbb{P}$, $\mathbf{y} \sim \mathbb{Q}$ from the test set $\{10, 20, 50, 100, 200\}$ with different percentages of affected data $\delta \in \{0.1, 0.5, 1.0\}$. Each result in our figure compares the detection performance between SCNTK and UAE averaged over the performance from MNIST experiments and CIFAR10 experiments, for different δ and 5 random seeds. Due to the space limit, we show results under 4 representative dataset shifts in Figure 2 and the rest in the Appendix. These are:

1. Medium Gaussian noise: Injecting Gaussian noise with a standard deviation $\sigma = 1$ to the sampled data.
2. Medium rotations and translations: A fraction δ percent of the samples are rotated by 40 degrees, translated by 0.2 and zoomed by 0.2.
3. Adversarial samples: A fraction δ percent of data is turned into adversarial smaples by FGSM (Goodfellow et al., 2014).
4. Imbalanced dataset: A fraction δ percent of data from class 0 is removed.

Overall, Figure 2 shows MMD-SCNTK achieves better detection accuracy, except for the case of imbalanced dataset. This provides an empirical justification on using SCNTK over a simple Gaussian kernel on with nonlinear randomly projected feature with the untrained encoder.

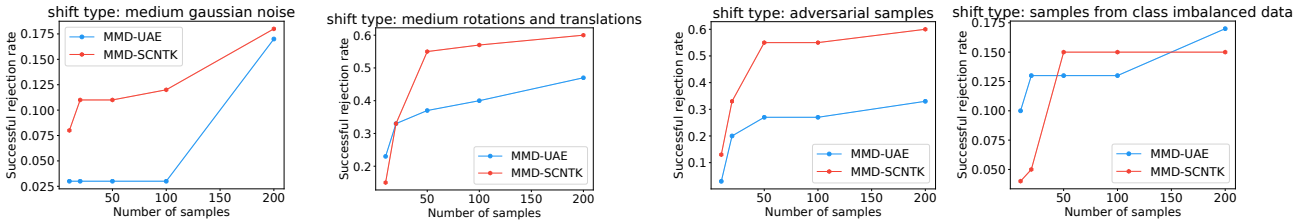

Figure 2. The above four plots show comparisons between MMD-UAE and our MMD-SCNTK method. MMD-UAE baseline applies the gaussian kernel to the output feature of the randomly initialized encoder network. The evaluation metric is the dataset shift detection accuracy. When the dataset is injected with gaussian noise, applied rotations and translations, augmented with adversarial samples, we see a more than 2x performance improvement with MMD-SCNTK. One exception is with the imbalanced dataset where MMD-SCNTK performs similarly to the baseline.

Table 1. MNIST ($\alpha = 0.05$): empirical average test power under N real images and N DCGAN samples. C_i is the number of channels for that layer, which is the width of a conv network. The performance becomes better and stable as the width is increased to $C_i = 300$. 100 is not sufficient.

N	$C_i = 100$	$C_i = 200$	$C_i = 300$	$c_i = 500$
200	0.10±0.02	0.29±0.02	0.32±0.03	0.33±0.02
400	0.09±0.04	0.71±0.08	0.75±0.02	0.83±0.08
600	0.25±0.07	0.92±0.02	0.96±0.02	0.97±0.03
800	0.21±0.08	0.96±0.02	1.00±0.00	1.00±0.00
1000	0.26±0.06	1.00±0.00	1.00±0.00	1.00±0.00
Avg	0.18	0.78	0.81	0.83

5.2. Two-sample test comparisons with methods using the optimized kernel

To compare MMD-SCNTK with optimized kernel methods, we use the following experiment setup for MNIST experiments and CIFAR10 experiments. The result is summarized in Table 2. Similar to the previous set of experiments in section 5.1, $S_{\mathbb{P}}$ is sampled from MNIST or CIFAR10 datasets, however $S_{\mathbb{Q}}$ is generated using a trained generator of a DCGAN for MNIST experiments. For CIFAR10 experiments, $S_{\mathbb{Q}}$ is sampled from CIFAR10.1 dataset (Recht et al., 2019). The main difference from the experiments in section 5.1 is that the dataset shift is no longer applied manually. The successful rejection rate of a null hypothesis $\mathbb{P} = \mathbb{Q}$ is compared between MMD-SCNTK and baselines.

Choosing the width: As the SCNTK in Equation 4.6 only converges to the deterministic form in the large width limit, we empirically investigate the trend in the two-sample test performance as we increase the width, which is the number of channels for a convolutional network. Table 1 compares the results of the MNIST experiments when varying the width of the network. We see that the performance is relatively stable with widths of 200, 300, and 500. So we chose the width of 300 for our experiments in Table 2.

Baselines: We compare MMD-SCNTK with the baseline methods used in (Liu et al., 2020). These baseline methods optimize parameters of the kernel to maximize the test power on the split training dataset. Hence, these methods use half of the test samples as the training data to maximize the testing power and the other half to test whether $\mathbb{P} = \mathbb{Q}$ using the optimized kernel. We give brief a explanation for these methods.

ME (Jitkrittum et al., 2016) optimizes the data points used in the computation of distance in Gaussian kernel mean embeddings. SCF (Jitkrittum et al., 2016) is a similar method but optimizes the frequencies of the kernel instead. C2ST-L (Lopez-Paz & Oquab, 2016) is a classifier based method which trains a binary classifier between $\mathbb{S}_{\mathbb{P}}$ and $\mathbb{S}_{\mathbb{Q}}$. If $\mathbb{P} = \mathbb{Q}$, such a classifier should poorly behave. MMD-O (Rabanser et al., 2019) optimizes the bandwidth σ of a naive gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_2}{2\sigma^2}\right)$ to maximize the test power. MMD-D (Rabanser et al., 2019) optimizes the parameters w of the feature model $\phi_w(\mathbf{x})$ described in Equation 3.8. Such a training phase is carried out using the held-out data split from $S_{\mathbb{P}}$ and $S_{\mathbb{Q}}$. In addition, we provide another fixed kernel baseline MMD-SRF (shift-invariant random feature), which uses the same architecture as SCNTK but uses the inner products of the outputs of the network under $\mathbf{x}, \mathbf{x}'$. This is to simulate MMD-based method under an NNGP kernel (Lee et al., 2017a; Novak et al., 2018).

Results: Table 2 shows that the proposed SCNTK achieves quite a competitive performance relative to the strongest baseline MMD-D without the need of optimizing the kernel parameters. Most importantly, both SRF and SCNTK perform much better than the MMD-O baseline for both datasets under 5 different number of samples used in the experiment. This suggests that picking a composite kernel is more effective than merely optimizing the bandwidth σ of a simple gaussian kernel. In addition, we see that SCNTK is outperforming the SRF kernel (NNGP kernel) in this domain.

As seen, MMD-SCNTK outperforms the strongest baseline

Table 2. MNIST and CIFAR10 (Significance level $\alpha = 0.05$): Successful null hypothesis rejection accuracy for comparing N real images with N DCGAN samples on MNIST, and for comparing CIFAR10 images with CIFAR10.1 images. All the methods except SCNTK use a portion of test samples as the training data to maximize the test power with respect to kernel related parameters or parameters of the feature model. For both SCNTK and SRF methods, the default bandwidth 1.0 is used. Convolutions with a stride of 2 are used for all the layers. The width, i.e. the number of channels, is set to 300. More details are provided in the Appendix.

MNIST	ME	SCF	C2ST-S	C2ST-L	M-O	M-D	SRF	**SCNTK**
200	0.414±0.050	0.107±0.018	0.193±0.037	0.234±0.031	0.188±0.010	0.555±0.044	0.272±0.039	0.324±0.032
400	0.921±0.032	0.152±0.021	0.65±0.039	0.706±0.047	0.363±0.017	0.996±0.004	0.691±0.025	0.750±0.022
600	1.000±0.000	0.294±0.008	1.000±0.000	0.977±0.012	0.619±0.021	1.000±0.000	0.901±0.036	0.963±0.018
800	1.000±0.000	0.317±0.017	1.000±0.000	1.000±0.000	0.797±0.015	1.000±0.000	0.952±0.011	1.000±0.000
1000	1.000±0.000	0.346±0.019	1.000±0.000	1.000±0.000	0.894±0.016	1.000±0.000	1.000±0.000	1.000±0.000
Avg	0.867	0.243	0.768	0.783	0.572	0.91	0.763	0.807
CIFAR	ME	SCF	C2ST-S	C2ST-L	M-O	M-D	SRF	**SCNTK**
2000	0.588	0.171	0.452	0.529	0.316	0.744	0.440	**0.805**

Table 3. Using the exact NTK and Monte-Carlo approximations, same MNIST and CIFAR10 experiments in Table 2 are performed. **MC** is the monte-carlo approximation of NTK kernel using 20 samples of the model. **E** is the exact NTK kernel. CNTK, CNTK-MC, and CNTK-E use relu activations for all the layers whereas SCNTK kernels use the cosine activation for the first layer. Laplace uses the naive Laplace kernel. We perform a grid search for the coefficient of the Laplace kernel between 0.1 and 10.

MNIST	**SCNTK**	**SCNTK-MC**	**SCNTK-E**	CNTK	CNTK-MC	CNTK-E	Laplace
200	0.324±0.032	0.315±0.037	0.340±0.041	0.281±0.031	0.312±0.022	0.298±0.018	0.182±0.011
400	0.750±0.022	0.763±0.032	0.760±0.016	0.683±0.019	0.712±0.024	0.702±0.013	0.283±0.21
600	0.963±0.018	0.969±0.019	0.971±0.017	0.945±0.021	0.942±0.018	0.967±0.017	0.421±0.23
800	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	0.638±0.31
1000	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	1.000±0.000	0.662±0.28
Avg	0.807	0.809	0.814	0.782	0.793	0.793	0.437
CIFAR	**SCNTK**	**SCNTK-MC**	**SCNTK-E**	CNTK	CNTK-MC	CNTK-E	Laplace
2000	0.805	0.816	0.808	0.742	0.781	0.776	0.261

MMD-O in the CIFAR10 experiment but not in the MNIST experiment. We conjecture that this is because of higher diversity of CIFAR10 images versus MNIST images, which limits its generalization power with only 1000 training samples. It should be noted that unlike MMD-O, MMD-SCNTK can use all the 2000 samples for testing.

Ablation studies for finite widths and shift-invariance: Since the shift-invariance property is derived under the infinite-width limit, we include additional experiments using exact NTK and monte-carlo (MC) approximations of the kernel. These MNIST and CIFAR10 experiments are done under the same setup used in Table 2. We can see from the first 3 columns in Table 3 that the performance gap between using the finite width, MC approximations, and the exact NTK is relatively small. This motivates the use of computationally cheap random feature approximations in practice. In addition, we include the results with standard CNTK in the next 3 columns of Table 3. As can be seen, SCNTK variants do provide a slight advantage in these domains. In the last column, we show the testing performance of using a naive Laplace kernel, which does not take advantage of the spatial information.

Computational advantage of MMD-SCNTK: For each MMD-SCNTK computation used in MNIST experiments in Table 2, we compare its wall-clock time between using pairwise kernel evaluation in Equation 3.3 and using direct MMD evaluation in Equation 4.1. Figure 3 shows that naive kernel evaluation yields a quadradic time complexity w.r.t the number of samples, while taking advantage of the random feature approximation (linear kernel) for direct MMD computation yields a linear time complexity.

We also compare our method against MMD-O in terms of the wall-clock time for computing one MMD value. Table 4 shows that MMD-O has a linear growth in the training time and a quadratic growth in the MMD computation. Overall, we could see a competitive fixed kernel method would also bring computational advantage when the number of samples keeps growing.

5.3. Outlier detection as a proxy for one-sample test experiments

For the OOD detection task, we use one of the CIFAR10 or SVHN (Netzer et al., 2011) datasets as the inlier and the other one as outlier, similar to the setup used in (Hendrycks

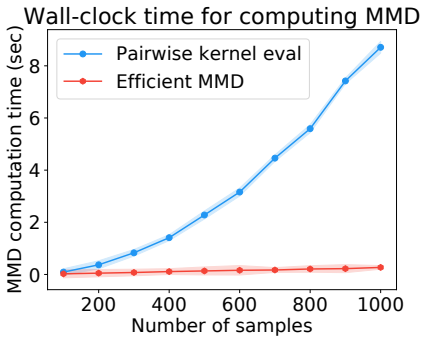

Figure 3. This plot compares the wall-clock time for computing MMD under SCNTK using two different methods. Pairwise kernel evaluation means that we explicitly compute the inner products between each pair of data points first, as in Equation 3.3. Efficient MMD directly computes the MMD using the simplified form in Equation 4.1.

Table 4. This table compares the wall-clock time needed for computing the MMD for the first time using MMD-SCNTK and MMD-O. It uses the same MNIST experiment setup in Table 2. Our fixed kernel method has no training time and presents a linear growth for the MMD computation time. Due to computing per-example gradients, our MMD computation time is longer than that of MMD-O, however the fixed kernel method saves the training time. In addition, MMD-O has a quadratic growth for the MMD computation time, and thus the gap between the computation time for SCNTK and trained MMD-O also decrease.

MNIST	SCNTK	M-O train	M-O
200	0.05 sec	37 sec	0.0034 sec
400	0.11 sec	75 sec	0.0141 sec
600	0.16 sec	110 sec	0.027 sec
800	0.21 sec	150 sec	0.051 sec
1000	0.27 sec	183 sec	0.076 sec

& Gimpel, 2016). Outlier scores for all inlier and outlier data are computed using the negative of kernel density estimation. We consider naive Gaussian KDE, CNTK KDE with all relu activations, and CNTK KDE with all relu activations, except the first layer which uses cosine activation (SCNTK KDE), as described in equation 4.7.

AUROC is computed as our metric to compare the performance of using these kernels. To compute NTK, we use a three-layer convolutional network with max-pooling for this task. The results in Table 5 empirically shows that (1) shift-invariance property is crucial for KDE task, and (2) composite kernel in the form of SCNTK is quite effective for such OOD detection tasks in image domains.

6. Conclusion

For the NTK kernel to be a valid kernel in MMD, we investigate the conditions under which it is characteristic. We make the kernel shift-invariant and the characteristic by using cosine activations for the first layer, and only use the inner products of gradients w.r.t the parameters of the layer 2 and the above. Such a shift-invariance can be understood as "watermark" invariance. In CIFAR10 experiments, we find our fixed kernel method with NTK could prevent the issue of overfitting to the split training data a learning-based deep kernel method could have.

Table 5. SCNTK for outlier detection. SCNTK, CNTK with all relu activations (CNTK-relu), and naive Gaussian KDE are compared for the outlier detection task with CIFAR10 and SVHN datasets. With a fixed kernel, SCNTK shows a promising results for OOD detection in both settings.

Inlier	Outlier	Gaussian	CNTK-relu	SCNTK
CIFAR10	SVHN	0.82	0.71	0.85
SVHN	CIFAR10	0.20	0.51	0.80

Our two-sample test experiments mainly use MNIST and CIFAR10. The most relavant three baselines we consider are: (1) Gaussian kernel applied to the features of a random encoder network, (2) Gaussian kernel with the optimized bandwidth applied in the pixel space, and (3) Gaussian kernel applied to the output feature of an optimized network for the test power maximization.

Our results show the expressivity of SCNTK does bring benefits over the other fixed kernel methods. Furthermore, SCNTK provides a competitive and efficient alternative to optimized kernel methods with a training phase.

Acknowledgements

We would like to give special thanks to Denny Wu and anonymous reviewers for their helpful comments and discussions. SJ and JB were funded by LG Electronics and NSERC. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute for Artificial Intelligence.

References

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pp. 8141–8150, 2019.

Baldi, P., Sadowski, P., and Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):1–9, 2014.

Berlinet, A. and Thomas-Agnan, C. Reproducing kernel hilbert spaces in probability and statistics. 2004.

Bietti, A. and Mairal, J. On the inductive bias of neural tangent kernels. *arXiv preprint arXiv:1905.12173*, 2019.

Cho, Y. *Kernel methods for deep learning*. PhD thesis, UC San Diego, 2012.

Cho, Y. and Saul, L. Kernel methods for deep learning. *Advances in neural information processing systems*, 22: 342–350, 2009.

Choi, H., Jang, E., and Alemi, A. A. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Dwass, M. Modified randomization tests for nonparametric hypotheses. *The Annals of Mathematical Statistics*, pp. 181–187, 1957.

Fernández, V. A., Gamero, M. J., and Garcia, J. M. A test for the two-sample problem based on empirical characteristic functions. *Computational statistics & data analysis*, 52 (7):3730–3748, 2008.

Gao, R., Liu, F., Zhang, J., Han, B., Liu, T., Niu, G., and Sugiyama, M. Maximum mean discrepancy is aware of adversarial attacks. *arXiv preprint arXiv:2010.11415*, 2020.

Geifman, A., Yadav, A., Kasten, Y., Galun, M., Jacobs, D., and Basri, R. On the similarity between the laplace and neural tangent kernels. *arXiv preprint arXiv:2007.01580*, 2020.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., and Smola, A. J. A kernel approach to comparing distributions. 2007.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Jean, N., Xie, S. M., and Ermon, S. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. In *Advances in Neural Information Processing Systems*, pp. 5322–5333, 2018.

Jitkrittum, W., Szabó, Z., Chwialkowski, K. P., and Gretton, A. Interpretable distribution features with maximum testing power. In *Advances in Neural Information Processing Systems*, pp. 181–189, 2016.

Kirchler, M., Khorasani, S., Kloft, M., and Lippert, C. Two-sample testing using deep learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 1387–1398. PMLR, 2020.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017a.

Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

Lee, K., Lee, H., Lee, K., and Shin, J. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017b.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y., and Póczos, B. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pp. 2203–2213, 2017.

Li, Z., Wang, R., Yu, D., Du, S. S., Hu, W., Salakhutdinov, R., and Arora, S. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

Liang, S., Li, Y., and Srikant, R. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Liu, F., Xu, W., Lu, J., Zhang, G., Gretton, A., and Sutherland, D. J. Learning deep kernels for non-parametric two-sample tests. *arXiv preprint arXiv:2002.09116*, 2020.

Lopez-Paz, D. and Oquab, M. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

Muandet, K., Fukumizu, K., Sriperumbudur, B., and Scholkopf, B. Kernel mean embedding of distributions: A review and beyond. *arXiv preprint arXiv:1605.09522*, 2016.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Novak, R., Xiao, L., Lee, J., Bahri, Y., Yang, G., Hron, J., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

Rabanser, S., Günnemann, S., and Lipton, Z. Failing loudly: An empirical study of methods for detecting dataset shift. In *Advances in Neural Information Processing Systems*, pp. 1396–1408, 2019.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pp. 5389–5400. PMLR, 2019.

Rudin, W. Fourier analysis on groups. 1962.

Serfling, R. Approximation theorems of mathematical statistics. 1980.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

Sriperumbudur, B. K., Gretton, A., Fukumizu, K., Schölkopf, B., and Lanckriet, G. R. Hilbert space embeddings and metrics on probability measures. *The Journal of Machine Learning Research*, 11:1517–1561, 2010.

Sutherland, D. J., Tung, H.-Y., Strathmann, H., De, S., Ramdas, A., Smola, A., and Gretton, A. Generative models and model criticism via optimized maximum mean discrepancy. *arXiv preprint arXiv:1611.04488*, 2016.

Tancik, M., Srinivasan, P. P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J. T., and Ng, R. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020.

Wenliang, L., Sutherland, D., Strathmann, H., and Gretton, A. Learning deep kernels for exponential family densities. In *International Conference on Machine Learning*, pp. 6737–6746, 2019.

Wilson, A. G., Hu, Z., Salakhutdinov, R., and Xing, E. P. Deep kernel learning. volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cadiz, Spain, 09–11 May 2016. PMLR. URL http://proceedings.mlr.press/v51/wilson16.html.