# FourierFormer: Transformer Meets Generalized Fourier Integral Theorem

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

Multi-head attention empowers the recent success of transformers, the state-of-the-art models that have achieved remarkable success in sequence modeling and beyond. These attention mechanisms compute the pairwise dot products between the queries and keys, which results from the use of unnormalized Gaussian kernels with the assumption that the queries follow a mixture of Gaussian distribution. There is no guarantee that this assumption is valid in practice. In response, we first interpret attention in transformers as a nonparametric kernel regression. We then propose the FourierFormer, a new class of transformers in which the dot-product kernels are replaced by the novel generalized Fourier integral kernels. Different from the dot-product kernels, where we need to choose a good covariance matrix to capture the dependency of the features of data, the generalized Fourier integral kernels can automatically capture such dependency and remove the need to tune the covariance matrix. We theoretically prove that our proposed Fourier integral kernels can efficiently approximate any key and query distributions. Compared to the conventional transformers with dot-product attention, FourierFormers attain better accuracy and reduce the redundancy between attention heads. We empirically corroborate the advantages of FourierFormers over the baseline transformers in a variety of practical applications including language modeling and image classification.

## 1 Introduction

Transformers [76] are powerful neural networks that have achieved tremendous success in many areas of machine learning [38, 69, 34] and become the state-of-the-art model on a wide range of applications across different data modalities, from language [22, 1, 17, 12, 55, 4, 8, 20] to images [23, 41, 71, 56, 52, 26], videos [3, 42], point clouds [90, 29], and protein sequence [58, 32]. In addition to their excellent performance on supervised learning tasks, transformers can also effectively transfer the learned knowledge from a pretraining task to new tasks with limited or no supervision [53, 54, 22, 87, 40]. At the core of transformers is the dot-product self-attention, which mainly accounts for the success of transformer models [13, 49, 39]. This dot-product self-attention learn self-alignment between tokens in an input sequence by estimating the relative importance of a given token with respect to all other tokens. It then transform each token into a weighted average of the feature representations of other tokens where the weight is proportional to a importance score between each pair of tokens. The importance scores in self-attention enable a token to attend to other tokens in the sequence, thus capturing the contextual representation [6, 76, 36].

### 1.1 Self-Attention

Given an input sequence $\boldsymbol{X} := [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D_x}$ of $N$ feature vectors, self-attention computes the output sequence $\mathbf{H}$ from $\boldsymbol{X}$ as follows:

**Step 1: Projecting the input sequence into different subspaces.** The input sequence $\boldsymbol{X}$ is transformed into the query matrix $\mathbf{Q}$, the key matrix $\mathbf{K}$, and the value matrix $\mathbf{V}$ via three linear

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

transformations

$$\mathbf{Q} = \boldsymbol{X}\mathbf{W}_Q^\top; \mathbf{K} = \boldsymbol{X}\mathbf{W}_K^\top; \mathbf{V} = \boldsymbol{X}\mathbf{W}_V^\top,$$

where $\mathbf{W}_Q, \mathbf{W}_K \in \mathbb{R}^{D \times D_x}$, and $\mathbf{W}_V \in \mathbb{R}^{D_v \times D_x}$ are the weight matrices. We denote $\boldsymbol{Q} := [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N]^\top, \mathbf{K} := [\boldsymbol{k}_1, \cdots, \boldsymbol{k}_N]^\top$, and $\mathbf{V} := [\mathbf{v}_1, \cdots, \mathbf{v}_N]^\top$, where the vectors $\boldsymbol{q}_i, \boldsymbol{k}_i, \mathbf{v}_i$ for $i = 1, \cdots, N$ are the query, key, and value vectors, respectively.

**Step 2: Computing the output as a weighted average.** The output sequence $\mathbf{H} := [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N]^\top$ is then given by

$$\mathbf{H} = \mathrm{softmax}\Big(\mathbf{Q}\mathbf{K}^\top / \sqrt{D}\Big)\mathbf{V} := \mathbf{A}\mathbf{V}, \tag{1}$$

where the softmax function is applied to each row of the matrix $(\mathbf{Q}\mathbf{K}^\top)/\sqrt{D}$. For each query vector $\boldsymbol{q}_i$, $i = 1, \cdots, N$, Eqn. (1) can be written in the vector form to compute the output vector $\boldsymbol{h}_i$ as follows

$$\boldsymbol{h}_i = \sum_{j=1}^{N} \mathrm{softmax}\Big(\boldsymbol{q}_i^\top \boldsymbol{k}_j / \sqrt{D}\Big)\mathbf{v}_j := \sum_{j=1}^{N} a_{ij}\mathbf{v}_j. \tag{2}$$

The matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ and its component $a_{ij}$ for $i, j = 1, \cdots, N$ are the attention matrix and attention scores, respectively. The self-attention computed by equations (1) and (2) is called the dot-product attention or softmax attention. In our paper, we refer a transformer that uses this attention as the baseline transformer with the dot-product attention or the dot-product transformer. The structure of the attention matrix $\mathbf{A}$ after training governs the ability of the self-attention to capture contextual representation for each token.

**Multi-head Attention** Each output sequence $\mathbf{H}$ forms an attention head. Multi-head attention concatenates multiple heads to compute the final output. Let $H$ be the number of heads and $\mathbf{W}^O \in \mathbb{R}^{HD_v \times HD_v}$ be the projection matrix for the output. The multi-head attention is defined as

$$\mathrm{MultiHead}(\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\}_{i=1}^{H}) = \mathrm{Concat}(\mathbf{H}_1, \ldots, \mathbf{H}_H)\mathbf{W}^O.$$

The capacity of the attention mechanism and its ability to learn diverse syntactic and semantic relationships determine the success of transformers [70, 77, 16, 78, 30]. However, equations (1) and (2) implies that the dot-product attention assumes the features $(q_{i1}, \ldots, q_{iD})$ in $\boldsymbol{q}_i$, as well as the features $(k_{j1}, \ldots, q_{jD})$ in $\boldsymbol{k}_j$, are independent. Thus, the dot-product attention fail to capture the correlations between these features, limiting its representation capacity and inhibit the performance of transformers on practical tasks where there is no guarantee that independent features can learned from complex data. One solution to capture correlations between features $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ is to introduce covariance matrices into the formulation of the dot-product attention with the cost of significantly increasing of the computational complexity. Also, choosing good covariance matrices is difficult.

### 1.2 Contribution

In this paper, we first establish a correspondence between self-attention and nonparametric kernel regression. Under this new perspective of self-attention, we explain the limitation of the dot-product self-attention that it may fail to capture correlations between the features in the query and key vectors. We then leverage the generalized Fourier integral theorems, which can automatically capture these correlations, and derive the generalized Fourier integral estimators for the nonparametric regression problem. Using this new density estimator, we propose the FourierFormer, a novel class of transformers that can capture correlations between features in the query and key vectors of self-attention. In summary, our contribution is three-fold:

1. We derive the formula of self-attention from solving a nonparametric kernel regression problem, thus providing a nonparametric regression interpretation to study and further develop self-attention.
2. We develop the generalized Fourier integral estimators for the nonparametric regression problem and provide theoretical guarantees for these estimator.
3. We propose the FourierFormer whose attentions use the generalized Fourier integral estimators to capture more efficiently correlations between features in the query and key vectors.

Finally, we empirically show that the FourierFormer attains significantly better accuracy than the baseline transformer with the dot-product attention on a variety of tasks including the WikiText language modeling and ImageNet image classsification. We also demonstrate in our experiments that FourierFormer helps reduce the redundancy between attention heads.

**Organization** We structure this paper as follows: In Section 2, we present the correspondence between self-attention and nonparametric kernel regression. In Section 3, we discuss the generalized Fourier integral estimators and define the FourierFormer. We validate and empirically analyze the advantages of FourierFormer in Section 4. We discuss related works in Section 5. The paper ends with concluding remarks. Technical proofs and more experimental details are provided in the Appendix.

**Notation** For any $N \in \mathbb{N}$, we denote $[N] = \{1, 2, \ldots, N\}$. For any $D \geq 1$, $\mathbb{L}_1(\mathbb{R}^D)$ denotes the space of real-valued functions on $\mathbb{R}^D$ that are integrable. For any two sequences $\{a_N\}_{N\geq 1}, \{b_N\}_{N\geq 1}$, we denote $a_N = \mathcal{O}(b_N)$ to mean that $a_N \leq C b_N$ for all $N \geq 1$ where $C$ is some universal constant.

## 2 A Nonparametric Regression Interpretation of Self-attention

In this section, we establish the connection between self-attention and nonparametric kernel regression. In particular, we derive the self-attention in equation (2) as a nonparametric kernel regression in which the key vectors $\boldsymbol{k}_j$ and value vectors $\mathbf{v}_j$ are training inputs and training targets, respectively, while the query vectors $\boldsymbol{q}_i$ and the output vectors $\boldsymbol{h}_i$ form a set of new inputs and their corresponding targets that need to be estimated, respectively, for $i, j = 1, \cdots, N$. In general, we can view the training set $\{\boldsymbol{k}_j, \mathbf{v}_j\}$ for $j \in [N]$ to come from the following *nonparametric regression model*:

$$\mathbf{v}_j = f(\boldsymbol{k}_j) + \varepsilon_j, \tag{3}$$

where $\varepsilon_1, \ldots, \varepsilon_N$ are independent noises such that $\mathbb{E}(\varepsilon_j) = 0$. Furthermore, we consider a random design setting where the key vectors $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N$ are i.i.d. samples from the distribution that admits $p$ as density function. By an abuse of notation, we also denote $p$ as the joint density where the key and value vectors $(\mathbf{v}_1, \boldsymbol{k}_1), \ldots, (\mathbf{v}_N, \boldsymbol{k}_N)$ are i.i.d. samples from. Here, $f$ is a true but unknown function and we would like to estimate it.

**Nadaraya–Watson estimator** Our approach to estimate the function $f$ is based on Nadaraya–Watson's nonparametric kernel regression approach [48]. In particular, from the nonparametric regression model (3), we have $\mathbb{E}[\mathbf{v}_j|\boldsymbol{k}_j] = f(\boldsymbol{k}_j)$ for all $j \in [N]$. Therefore, it is sufficient to estimate the conditional distribution of the value vectors given the key vectors. Given the density function $p$ of the key vectors and the joint density $p$ of the key and value vectors, for any pair of vectors $(\mathbf{v}, \boldsymbol{k})$ generate from model (3) we have

$$\mathbb{E}[\mathbf{v}|\boldsymbol{k}] = \int_{\mathbb{R}^D} \mathbf{v} \cdot p(\mathbf{v}|\boldsymbol{k}) d\mathbf{v} = \int \frac{\mathbf{v} \cdot p(\mathbf{v}, \boldsymbol{k})}{p(\boldsymbol{k})} d\mathbf{v}. \tag{4}$$

The formulation (4) of the conditional expectation indicates that as long as we can estimate the joint density function $p(\mathbf{v}, \boldsymbol{k})$ and the marginal density function $p(\mathbf{v})$, we are able to obtain an estimation for the conditional expectation and thus for the function $f$. This approach is widely known as Nadaraya–Watson's nonparametric kernel regression approach.

**Kernel density estimator** To estimate $p(\mathbf{v}, \boldsymbol{k})$ and $p(\boldsymbol{k})$, we employ the kernel density estimation approach [59, 50]. In particular, by using the isotropic Gaussian kernel with bandwidth $\sigma$, we have the following estimators of $p(\mathbf{v}, \boldsymbol{k})$ and $p(\boldsymbol{k})$:

$$\hat{p}_\sigma(\mathbf{v}, \boldsymbol{k}) = \frac{1}{N} \sum_{j=1}^{N} \varphi_\sigma(\mathbf{v} - \mathbf{v}_j) \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j), \qquad \hat{p}_\sigma(\boldsymbol{k}) = \frac{1}{N} \sum_{j=1}^{N} \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j), \tag{5}$$

where $\varphi_\sigma(.)$ is the isotropic multivariate Gaussian density function with diagonal covariance matrix $\sigma^2 \mathbf{I}_D$. Given the kernel density estimators (5), we obtain the following estimation of the function $f$:

$$\begin{aligned} \widehat{f}_\sigma(\boldsymbol{k}) &= \int_{\mathbb{R}^D} \frac{\mathbf{v} \cdot \hat{p}_\sigma(\mathbf{v}, \boldsymbol{k})}{\hat{p}_\sigma(\boldsymbol{k})} d\mathbf{v} = \int_{\mathbb{R}^D} \frac{\mathbf{v} \cdot \sum_{j=1}^{N} \varphi_\sigma(\mathbf{v} - \mathbf{v}_j) \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)}{\sum_{j=1}^{N} \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)} d\mathbf{v} \\ &= \frac{\sum_{j=1}^{N} \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j) \int \mathbf{v} \cdot \varphi_\sigma(\mathbf{v} - \mathbf{v}_j) d\mathbf{v}}{\sum_{j=1}^{N} \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)} = \frac{\sum_{j=1}^{N} v_j \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)}{\sum_{j=1}^{N} \varphi_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)}. \end{aligned} \tag{6}$$

**Connection between Self-Attention and nonparametric regression** By plugging the query vectors $\boldsymbol{q}_i$ into the function $\widehat{f}_\sigma$ in equation (6), we obtain that

$$\widehat{f}_\sigma(\boldsymbol{q}_i) = \frac{\sum_j^N \mathbf{v}_j \exp\left(-\|\boldsymbol{q}_i - \boldsymbol{k}_j\|^2/2\sigma^2\right)}{\sum_j^N \exp\left(-\|\boldsymbol{q}_i - \boldsymbol{k}_j\|^2/2\sigma^2\right)}$$
$$= \frac{\sum_j^N \mathbf{v}_j \exp\left[-\left(\|\boldsymbol{q}_i\|^2 + \|\boldsymbol{k}_j\|^2\right)/2\sigma^2\right] \exp\left(\boldsymbol{q}_i \boldsymbol{k}_j^\top/\sigma^2\right)}{\sum_j^N \exp\left[-\left(\|\boldsymbol{q}_i\|^2 + \|\boldsymbol{k}_{j'}\|^2\right)/2\sigma^2\right] \exp\left(\boldsymbol{q}_i \boldsymbol{k}_j^\top/\sigma^2\right)}. \tag{7}$$

If we further assume that the keys $\boldsymbol{k}_j$ are normalized, which is usually done in practice to stabilize the training of transformers [64], the value of $\widehat{f}_\sigma(\boldsymbol{q}_i)$ in equation (6) then becomes

$$\widehat{f}_\sigma(\boldsymbol{q}_i) = \frac{\sum_j^N \mathbf{v}_j \exp\left(\boldsymbol{q}_i \boldsymbol{k}_j^\top/\sigma^2\right)}{\sum_{j'}^N \exp\left(\boldsymbol{q}_i \boldsymbol{k}_{j'}^\top/\sigma^2\right)} = \sum_{j=1}^N \mathrm{softmax}\left(\boldsymbol{q}_i^\top \boldsymbol{k}_j/\sigma^2\right)\mathbf{v}_j. \tag{8}$$

When we choose $\sigma^2 = \sqrt{D}$ where $D$ is the dimension of $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$, equation (8) matches equation (2) of self-attention, namely, $\widehat{f}_\sigma(\boldsymbol{q}_i) = \boldsymbol{h}_i$. Thus, we have shown that self-attention performs nonparametric regression using isotropic Gaussian kernels.

**Remark 1** *The assumption that $\boldsymbol{k}_j$ is normalized is to recover the pairwise dot-product attention in transformers. In general, this assumption is not necessary. In fact, the isotropic Gaussian kernel in equation (7) is more desirable than the dot-product kernel in equation (8) of the pairwise dot-product attention since the former is Lipschitz while the later is not Lipschitz [35]. The Lipschitz constraint helps improve the robustness of the model [15, 74, 2] and stabilize the model training [46].*

**Limitation of Self-Attention** From our nonparametric regression interpretation, self-attention is derived from the use of isotropic Gaussian kernels for kernel density estimation and nonparametric regression estimation, which may fail to capture the complex correlations between $D$ features in $\boldsymbol{q}_i$ and $\boldsymbol{k}_j$ [81, 31]. Using multivariate Gaussian kernels with dense covariance matrices can help capture such correlations; however, choosing good covariance matrices is challenging and inefficient [80, 66, 11]. In the following section, we discuss the Fourier integral estimator and its use as a kernel for computing self-attention in order to overcome these limitations.

# 3 FourierFormer: Transformer via Generalized Fourier Integral Theorem

In the following, we introduce generalized integral theorems that are able to capture the complex interactions among the features of the queries and keys. We then apply these theorems to density estimation and nonparametric regression problems. We also establish the convergence rates of these estimators. Given these density estimators, we introduce a novel family of transformers, named *FourierFormer*, that integrates the generalized Fourier integral theorem into the dot-product attention step of the standard transformer.

## 3.1 Generalized Fourier Integral Theorems and Their Applications

The Fourier integral theorem is a beautiful result in mathematics [85, 7] and has been recently used in nonparametric mode clustering, deconvolution problem, and generative modeling [31]. It is a combination of Fourier transform and Fourier inverse transform. In particular, for any function $p \in \mathbb{L}_1(\mathbb{R}^D)$, the *Fourier integral theorem* is given by

$$p(\boldsymbol{k}) = \frac{1}{(2\pi)^D} \int_{\mathbb{R}^D} \int_{\mathbb{R}^D} \cos(\boldsymbol{s}^\top(\boldsymbol{k} - \boldsymbol{y})) p(\boldsymbol{y}) d\boldsymbol{y} d\boldsymbol{s}$$
$$= \frac{1}{\pi^D} \lim_{R\to\infty} \int_{\mathbb{R}^D} \prod_{j=1}^D \frac{\sin(R(k_j - y_j))}{(k_j - y_j)} p(\boldsymbol{y}) d\boldsymbol{y}, \tag{9}$$

where $\boldsymbol{k} = (k_1, \ldots, k_D), \boldsymbol{y} = (y_1, \ldots, y_D), \boldsymbol{s} = (s_1, \ldots, s_D)$, and $R$ is the radius. The detailed derivation of Equation (9) is in Appendix A.3. Equation (9) suggests that $p_R(\boldsymbol{k}) := \frac{1}{\pi^D} \int_{\mathbb{R}^D} \prod_{j=1}^D \frac{\sin(R(y_j - k_j))}{(y_j - k_j)} p(\boldsymbol{y}) d\boldsymbol{y}$ can be used as an estimator of the function $p$.

**Benefits of the Fourier integral over Gaussian kernel** There are two important benefits of the estimator $p_R$: (i) it can automatically preserve the correlated structure lying within $p$ even when $p$ is very complex and high dimensional function. It is in stark contrast to the standard kernel estimator built based on multivariate Gaussian kernel where we need to choose good covariance matrix in the multivariate Gaussian kernel to guarantee such estimator to work well. We note that as the standard soft-max Transformer is constructed based on the multivariate Gaussian kernel, the issue of choosing good covariance matrix in dot-product transformer is inevitable; (ii) The product of sinc kernels in the estimator $p_R$ does not decay to a point mass when $R \to \infty$. It is in stark difference from the multivariate Gaussian kernel estimator, which converges to a point mass when the covariance matrix goes to 0. It indicates that $p_R$ is a non-trivial estimator of the function $p$. Finally, detailed illustrations of these benefits of the Fourier integral over Gaussian kernel in density estimation and nonparametric regression problems, which we have just shown to have connection to the self-attention in transformer, can be found in Section 8 in [31].

**Generalized Fourier integral estimator** Borrowing the above benefits of Fourier integral estimator $p_R$, in the paper we would like to consider a generalization of that estimator, named *generalized Fourier integral estimator*, which is given by:

$$p_R^{\phi}(\boldsymbol{k}) := \frac{R^D}{A^D} \int_{\mathbb{R}^D} \prod_{j=1}^{D} \phi\left(\frac{\sin(R(y_j - k_j))}{R(y_j - k_j)}\right) p(\boldsymbol{y}) d\boldsymbol{y}, \tag{10}$$

where $A := \int_{\mathbb{R}} \phi\left(\frac{\sin(z)}{z}\right) dz$ and $\phi : \mathbb{R} \to \mathbb{R}$ is a given function. When $\phi(\boldsymbol{k}) = \boldsymbol{k}$ for all $\boldsymbol{k} \in \mathbb{R}^D$, the generalized Fourier integral estimator $p_R^{\phi}$ becomes the Fourier integral estimator $p_R$. Under appropriate conditions on the function $\phi$ (see Theorem 1 in Section 3.1.1 and Theorem 3 in Appendix B.1), the estimator $p_R^{\phi}$ converges to the true function $p$, namely,

$$p(\boldsymbol{k}) = \lim_{R \to \infty} p_R^{\phi}(\boldsymbol{k}) = \lim_{R \to \infty} \frac{R^D}{A^D} \int_{\mathbb{R}^D} \prod_{j=1}^{D} \phi\left(\frac{\sin(R(y_j - k_j))}{R(y_j - k_j)}\right) p(\boldsymbol{y}) d\boldsymbol{y}. \tag{11}$$

We name the above limit as *generalized Fourier integral theorem*. Furthermore, the estimator $p_R^{\phi}$ also inherits similar aforementioned benefits of the Fourier integral estimator $p_R$. Therefore, we will use the generalized Fourier integral theorem as a building block for constructing density estimators and nonparametric regression estimators, which are crucial to develop the FourierFormer in Section 3.2.

#### 3.1.1 Density Estimation via Generalized Fourier Integral Theorems

We first apply the generalized Fourier integral theorem to the density estimation problem. To ease the presentation, we assume that $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N \in \mathbb{R}^D$ are i.i.d. samples from a distribution admitting density function $p$ where $D \geq 1$ is the dimension. Inspired by the generalized Fourier integral theorem, we obtain the following *generalized Fourier density estimator* $p_{N,R}^{\phi}$ of $p$ as follows:

$$p_{N,R}^{\phi}(\boldsymbol{k}) := \frac{R^D}{NA^D} \sum_{i=1}^{N} \prod_{j=1}^{D} \phi\left(\frac{\sin(R(k_j - k_{ij}))}{R(k_j - k_{ij})}\right), + \tag{12}$$

where $A = \int_{\mathbb{R}} \phi\left(\frac{\sin(z)}{z}\right) dz$ and $\boldsymbol{k}_i = (k_{i1}, \ldots, k_{iD})$ for all $i \in [N]$. To quantify the error between the generalized Fourier density estimator $p_{n,R}^{\phi}$ and the true density $p$, we utilize mean integrated squared errors (MISE) [84], which is given by:

$$\text{MISE}(p_{N,R}^{\phi}, p) := \int_{\mathbb{R}^D} (p_{N,R}^{\phi}(\boldsymbol{k}) - p(\boldsymbol{k}))^2 d\boldsymbol{k}. \tag{13}$$

We start with the following bound on the MISE between $p_{n,R}^{\phi}$ and $p$.

**Theorem 1** *Assume that $\int_{\mathbb{R}} \phi(\sin(z)/z) z^j dz = 0$ for all $j \in [m]$ and $\int_{\mathbb{R}} |\phi(\sin(z)/z)||z|^{m+1} dz < \infty$ for some $m \in \mathbb{N}$. Then, there exist universal constants $C$ and $C'$ depending on d and $A$ such that*

$$\textit{MISE}(p_{N,R}^{\phi}, p) \leq \frac{C}{R^{m+1}} + \frac{C' R^D}{N}.$$

Proof of Theorem 1 is in Appendix C.1. A few comments are in order. First, by choosing $R$ to balance the bias and variance in the bound of MISE in Theorem 1, we have the optimal $R$ as

$R = \mathcal{O}(N^{1/(D+m+1)})$. With that choice of $R$, the MISE rate of $p^{\phi}_{N,R}$ is $\mathcal{O}(N^{-(m+1)/(D+m+1)})$. Second, when $\phi(z) = z^l$ for $l \geq 4$ and $z \in \mathbb{R}$, the assumptions in Theorem 1 are satisfied when $m = 1$. Under this case, the MISE rate of $p^{\phi}_{N,R}$ is $\mathcal{O}(N^{-2/(D+2)})$. However, these assumptions do not satisfy when $\phi(z) = z^l$ and $l \in \{1, 2, 3\}$, which is due to the limitation of the current proof technique of Theorem 1 that is based on Taylor expansion of the estimator $p^{\phi}_{n,R}$.

To address the limitation of the Taylor expansion technique, we utilize the Plancherel theorem in Fourier analysis to establish the MISE rate of $p^{\phi}_{N,R}$ when $\phi(z) = z^l$ and $l \in \{1, 2, 3\}$. The details of the theoretical analyses for such setting are in Appendix B.

### 3.2 FourierFormer: Transformers with Fourier Attentions

Motivated by the preservation of the correlated structure of the function from the generalized Fourier integral theorem as well as the theoretical guarantees of density estimators, in this section we adapt the nonparametric regression interpretation of self-attention in Section 2 and propose the generalized Fourier nonparametric regression estimator in Section 3.2.1. We also establish the convergence properties of that estimator. Then, based on generalized Fourier nonparametric regression estimator, we develop the Fourier Attention and its corresponding FourierFormer in Section 3.2.2.

#### 3.2.1 Nonparametric Regression via Generalized Fourier Integral Theorem

We now discuss an application of the generalized Fourier integral theorems to the nonparametric regression setting (3), namely, we assume that $(\mathbf{v}_1, \boldsymbol{k}_1), \ldots, (\mathbf{v}_N, \boldsymbol{k}_N)$ are i.i.d. samples from the following nonparametric regression model:

$$\mathbf{v}_j = f(\boldsymbol{k}_j) + \varepsilon_j,$$

where $\varepsilon_1, \ldots, \varepsilon_N$ are independent noises such that $\mathbb{E}(\varepsilon_j) = 0$ and the key vectors $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_N$ are i.i.d. samples from $p$. Given the generalized Fourier density estimator (12), following the argument in Section 2, the Nadaraya–Watson estimator of the function $f$ based on the generalized Fourier density estimator is given by:

$$f_{N,R}(\boldsymbol{k}) := \frac{\sum_{i=1}^{N} \mathbf{v}_i \prod_{j=1}^{D} \phi\left(\frac{\sin(R(k_j - k_{ij}))}{R(k_j - k_{ij})}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{D} \phi\left(\frac{\sin(R(k_j - k_{ij}))}{R(k_j - k_{ij})}\right)}. \tag{14}$$

The main difference between the generalized Fourier nonparametric regression estimator $f_{N,R}$ in equation (14) and the estimator $\widehat{f}_\sigma$ in equation (6) is that the estimator $f_{N,R}$ utilizes the generalized Fourier density estimator to estimate the conditional distribution of the value vectors given the key vectors instead of the isotropic Gaussian kernel density estimator as in $\widehat{f}_\sigma$. As we highlighted in Section 3, an important benefit of the generalized Fourier density estimator is that it can capture the complex dependencies of the features of the value vectors and the key vectors while the Gaussian kernel needs to have good covariance matrix to do that, which is computationally expensive in practice.

We now have the following result establishing the mean square error (MSE) of $f_{N,R}$.

**Theorem 2** *Assume that $\int_{\mathbb{R}} \phi\left(\frac{\sin(z)}{z}\right) z^j dz = 0$ for all $1 \leq j \leq m$ and $\int_{\mathbb{R}} \left|\phi\left(\frac{\sin(z)}{z}\right)\right| |z|^j dz < \infty$ for any $m+1 \leq j \leq 2m+2$ for some $m \in \mathbb{N}$. Then, for any $\boldsymbol{k} \in \mathbb{R}^D$, there exist universal constants $C_1, C_2, C_3, C_4$ such that the following holds:*

$$\mathbb{E}\left[(f_{N,R}(\boldsymbol{k}) - f(\boldsymbol{k}))^2\right] \leq \left(\frac{C_1}{R^{2(m+1)}} + \frac{(f(\boldsymbol{k}) + C_2)R^D}{N}\right) \Big/ \left(p^2(\boldsymbol{k}) J(R)\right),$$

*where $J(R) = 1 - \frac{1}{p^2(\boldsymbol{k})}\left(\frac{C_3}{R^{2(m+1)}} + \frac{C_4 R^d \log(NR)}{N}\right)$. Here, the outer expectation is taken with respect to the key vectors $\boldsymbol{k}_1, \ldots, \boldsymbol{k}_N$ and the noises $\varepsilon_1, \ldots, \varepsilon_N$.*

Proof of Theorem 2 is in Appendix C.3. A few comments with Theorem 2 are in order. First, by choosing $R$ to balance the bias and variance in the bound of the MSE of the nonparametric generalized Fourier estimator $f_{N,R}$, we have the optimal radius $R$ as $R = \mathcal{O}(N^{\frac{1}{2(m+1)+D}})$. With that choice of the optimal radius $R$, the rate of $f_{N,R}$ is $\mathcal{O}(N^{-\frac{2(m+1)}{D+2(m+1)}})$. Second, when $\phi(z) = z^l$ for $l \geq 6$, the assumption on the function $\phi$ of Theorem 2 is satisfied with $m = 1$. Under this case, the rate of $f_{N,R}$ becomes $\mathcal{O}(N^{-\frac{4}{D+4}})$. In Appendix B, we also provide the rate of $f_{N,R}$ when $\phi(z) = z^l$ for some $l \leq 5$, which includes the original Fourier integral theorem.

#### 3.2.2 FourierFormer

Given the generalized Fourier nonparametric regression estimator $f_{N,R}$ in equation (14), by plugging the query values $\boldsymbol{q}_1, \ldots, \boldsymbol{q}_N$ into that function, we obtain the following definition of the Fourier attention:

**Definition 1 (Fourier Attention)** *A Fourier attention is a multi-head attention that does nonparametric regression using the generalized Fourier nonparametric regression estimator $f_{N,R}$. The output $\hat{\boldsymbol{h}}_i$ of the Fourier attention is then computed as*

$$\hat{\boldsymbol{h}}_i := f_{N,R}(\boldsymbol{q}_i) = \frac{\sum_{i=1}^{N} \mathbf{v}_i \prod_{j=1}^{D} \phi\left(\frac{\sin(R(q_{ij}-k_{ij}))}{R(q_{ij}-k_{ij})}\right)}{\sum_{i=1}^{N} \prod_{j=1}^{D} \phi\left(\frac{\sin(R(q_{ij}-k_{ij}))}{R(q_{ij}-k_{ij})}\right)} \qquad \forall\, i \in [N]. \tag{15}$$

Given the Fourier Attention in Definition 1, we then give the definition of FourierFormer as follows.

**Definition 2 (FourierFormer)** *A FourierFormer is a transformer that uses Fourier attention to capture dependency between tokens in the input sequence and the correlation between features in each token.*

**Remark 2 (The Nonnegativity of the Fourier Kernel)** *The density estimation via generalized Fourier integral theorem in Section 3.1.1 does not require the generalized Fourier density estimator to be nonnegative. However, empirically, we observe that negative density estimator can cause instability in training the FourierFormer. Thus, in FourierFormer, we choose the function $\phi$ to be a nonnegative function to enforce the density estimator to be nonnegative. In particular, we choose $\phi$ to be power functions of the form $\phi(x) = x^{2m}$, where $m$ is an positive integer. Note that when $m = 2$ and $m = 4$, the kernels in our generalized Fourier integral estimators are the well-known Fejer-de la Vallee Poussin and Jackson-de la Vallee Poussin kernels [19].*

### 3.3 An Efficient Implementation of the Fourier Attention

The Fourier kernel is implemented efficiently in the C++/CUDA extension developed by Pytorch [51]. The idea is similar to the function `cdist` [51], which computes the p-norm distance between each pair of the two collections of row vectors. In our case, we aim to compute kernel functions that represent a Fourier attention in Definition 1. The core of this implementation is the following Fourier metric function $d_f$:

$$d_f(\boldsymbol{q}_i, \boldsymbol{k}_j) = \prod_{d=1}^{D} \phi\left(\frac{\sin(R(\boldsymbol{q}_{id} - \boldsymbol{k}_{jd}))}{R(\boldsymbol{q}_{id} - \boldsymbol{k}_{jd})}\right)$$

We directly implement $d_f$ as a `torch.autograd.Function` [51] in which we provide an efficient way to compute forward and backward function ($d_f$ and gradient of $d_f$). While the implementation of the forward function is straight forward, the backward function is more tricky since we need to optimize the code to compute the gradient of $d_f$ w.r.t to variables $\boldsymbol{q}$, $\boldsymbol{k}$, and $R$ all at once. We can develop the backward function with highly parallel computation by exploiting GPU architecture and utilizing the reduction technique. The computational time is comparable to function `cdist`; thus, our FourierFormer implementation is as computationally time-efficient.

## 4 Experimental Results

In this section, we numerically justify the advantage of FourierFormer over the baseline dot-product transformer on two large-scale tasks: language modeling on WikiText-103 [44] (Section 4.1) and image classification on ImageNet [21, 60] (Section 4.2). We aim to show that: (i) FourierFormer achieves better accuracy than the baseline transformer on a variety of practical tasks with different data modalities, and (ii) FourierFormer helps reduce head redundancy compared to the baseline transformer (Section 4.3).

Throughout the section, we compare FourierFormers with the baseline dot-product transformers of the same configuration. In all experiments, we made the constant $R$ in Fourier attention (see

**Table 1.** Perplexity (PPL) on WikiText-103 of FourierFormers compared to the baselines. FourierFormers achieve much better PPL than the baselines.

| Method | Valid PPL | Test PPL |
|---|---|---|
| *Baseline dot-product (small)* | 33.15 | 34.29 |
| FourierFormer (small) | **31.86** | **32.85** |
| *Baseline dot-product (medium)* | 27.90 | 29.60 |
| FourierFormer (medium) | **26.51** | **28.01** |

equation (58)) to be a learnable scalar and set choose the function $\phi(x) = x^4$ (see Remark 2). All of our results are averaged over 5 runs with different seeds. More details on the models and training are provided in Appendix D. We also provide additional experimental results in Appendix E.

### 4.1 Language Modeling on WikiText-103

**Datasets and metrics** WikiText-103 is a collection of articles from Wikipedia, which have long contextual dependencies. The training set consists of about $28K$ articles containing $103M$ running words; this corresponds to text blocks of about 3600 words. The validation and test sets have $218K$ and $246K$ running words, respectively. Each of them contains 60 articles and about $268K$ words. Our experiment follows the standard setting [44, 64] and splits the training data into $L$-word independent long segments. For evaluation, we use a batch size of 1, and process the text sequence with a sliding window of size $L$. The last position is used for computing perplexity (PPL) except in the first segment, where all positions are evaluated as in [1, 64].

**Models and baselines** Our implementation is based on the public code by [64].[1] We use their small and medium models in our experiments. In particular, for small models, the key, value, and query dimension are set to 128, and the training and evaluation context length are set to 256. For medium models, the key, value, and query dimension are set to 256, and the training and evaluation context length are set to 384. In both configurations, the number of heads is 8, the feed-forward layer dimension is 2048, and the number of layers is 16.

**Results** We report the validation and test perplexity (PPL) of FourierFormer versus the baseline transformer with the dot-product attention in Table 1. FourierFormers attain much better PPL than the baselines in both small and medium configurations. For the small configuration, the improvements of FourierFormer over the baseline are 1.29 PPL in validation and 1.44 PPL in test. For the medium configuration, these improvements are 1.39 PPL in validation and 1.59 PPL in test. These results suggest that the advantage of FourierFormer over the baseline dot-product transformer grows with the model's size. This meets our expectation because larger models has larger query and key dimensions, e.g. the language model with medium configuration in this experiment has the query and key dimension of 256 versus 128 as in the language model with small configuration. Since the advantage of FourierFormer results from the property that FourierFormer can capture correlation between features in query and key vectors, the larger the query and key dimensions are, the more advantage FourierFormer has.

### 4.2 Image Classification on ImageNet

**Datasets and metrics** The ImageNet dataset [21, 60] consists of $1.28M$ training images and $50K$ validation images. For this benchmark, the model learns to predict the category of the input image among 1000 categories. Top-1 and top-5 classification accuracies are reported.

**Models and baselines** We use the DeiT-tiny model [72] with 12 transformer layers, 4 attention heads per layer, and the model dimension of 192. To train the models, we follow the same setting and configuration as for the baseline [72].[2]

**Results** We summarize our resuls in Table 2. Same as in the language modeling experiment, for this image classification task, the Deit model equipped with FourierFormer significantly outperforms the baseline Deit dot-product transformer in both top-1 and top-5 accuracy. This result suggests that the advantage of FourierFormer over the baseline dot-product transformer holds across different data modalities.

[1]Implementation available at https://github.com/IDSIA/lmtool-fwp.
[2]Implementation available at https://github.com/facebookresearch/deit.

**Table 2.** Top-1 and top-5 accuracy (%) of FourierFormer Deit vs. the baseline Deit with dot-product attention. FourierFormer Deit outperforms the baseline in both top-1 and top-5 accuracy.

| Method | Top-1 Acc | Top-5 Acc |
|---|---|---|
| *Baseline DeiT* | 72.23 | 91.13 |
| FourierFormer DeiT | **73.25** | **91.66** |

**Table 3.** Layer-average mean and standard deviation of $\mathcal{L}_2$ distances between heads of FourierFormer versus the baseline transformer with dot-product attention trained for the WikiText-103 language modeling task. FourierFormer has greater $\mathcal{L}_2$ distance between heads than the baseline and thus captures more diverse attention patterns.

| Method | Train | Test |
|---|---|---|
| *Baseline dot-product* | $6.20 \pm 2.30$ | $6.17 \pm 2.30$ |
| FourierFormer | $\mathbf{7.45 \pm 2.50}$ | $\mathbf{7.37 \pm 2.44}$ |

### 4.3 FourierFormer Helps Reducing Head Redundancy

To study the diversity between attention heads, given the model trained for the WikiText-103 language modeling task, we compute the average $\mathcal{L}_2$ distance between heads in each layer. We show the layer-average mean and variance of distances between heads in Table 3. Results in Table 3 shows that FourierFormer obtains greater $\mathcal{L}_2$ distance between attention heads than the baseline transformer with the dot-product attention and thus helps reduce the head redundancy. Note that we use the small configuration as specified in Section 4.1 for both models.

## 5 Related Work

**Interpretation of Attention Mechanism in Transformers** Recent works have tried to gain an understanding of transformer's attention from different perspectives. [73] considers attention as applying kernel smoother over the inputs. Extending this kernel approach, [33, 14, 82] linearize the softmax kernel in dot-product attention and propose a family of efficient transformers with linear computational and memory complexity. [9] then shows that these linear transformers are comparable to a Petrov-Galerkin projection [57], suggesting that the softmax normalization in the dot-product attention is sufficient but not necessary. Other works provide an understanding of attention in transformers via ordinary/partial differential equation include [43, 62]. In addition, [68, 28, 89] relate attentions in transformers to a Gaussian mixture models. Several works also connect the attention mechanism to graph-structured learning and message passing in graphical models [83, 65, 37]. Our work focuses on deriving the connection between self-attention and nonparametric kernel regression and exploring better regression estimator, such as the generalized Fourier nonparametric regression estimator, to improve the performance of transformers.

**Redundancy in Transformers** [18, 45, 24] show that neurons and attention heads in the pre-trained transformer are redundant and can be removed when applied on a downstream task. By studying the contextualized embeddings in pre-trained networks, it has been demonstrated that the learned representations from these redundant models are highly anisotropic [47, 25]. Furthermore, [63, 67, 79, 61] employ knowledge distillation and sparse approximation to enhance the efficiency of transformers. Our FourierFormer is complementary to these methods and can be combined with them.

## 6 Concluding Remarks

In this paper, we establish the correspondence between the nonparametric kernel regression and the self-attention in transformer. We then develop the generalized Fourier integral estimators and propose the FourierFormer, a novel class of transformers that use the generalized Fourier integral estimators to construct their attentions for efficiently capturing the correlations between features in the query and key vectors. We theoretically prove the approximation guarantees of the generalized Fourier integral estimators and empirically validate the advantage of FourierFormer over the baseline transformer with the dot-product attention in terms of accuracy and head redundancy reduction. It is interesting to incorporate robust kernels into the nonparametric regression framework of FourierFormer to enhance the robustness of the model under data perturbation and adversarial attacks. A limitation of FourierFormer is that it still has the same quadratic computational and memory complexity as the baseline transformer with the dot-product attention. We leave the development of the linear version of FourierFormer that achieves linear computational and memory complexity as future work. It is worth noting that there is no potential negative societal impacts of FourierFormer.

## References


[1] Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3159–3166, 2019.

[2] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[3] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6816–6826, 2021.

[4] Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019.

[5] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] S. Bochner. *Lectures on Fourier Integrals*. Princeton University Press, 1959.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Shuhao Cao. Choose a transformer: Fourier or galerkin. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, volume 57, 2014.

[11] J.E. Chacón and T. Duong. *Multivariate Kernel Smoothing and its Applications*. CRC Press, 2018.

[12] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

[13] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[14] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021.

[15] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[16] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy, August 2019. Association for Computational Linguistics.

[17] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

[18] Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. Analyzing redundancy in pretrained transformer models. *arXiv preprint arXiv:2004.04010*, 2020.

[19] Kathryn Bullock Davis. Mean square error properties of density estimates. *The Annals of Statistics*, 3(4):1025–1030, 1975.

[20] Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

[21] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[22] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[23] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[24] Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. Analyzing individual neurons in pre-trained language models. *arXiv preprint arXiv:2010.02695*, 2020.

[25] Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *arXiv preprint arXiv:1909.00512*, 2019.

[26] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021.

[27] J. Fan. On the optimal rates of convergence for nonparametric deconvolution problems. *Annals of Statistics*, 19(3):1257–1272, 1991.

[28] Prasad Gabbur, Manjot Bilkhu, and Javier Movellan. Probabilistic attention for interactive segmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[29] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021.

[30] John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics.

[31] N. Ho and S.G. Walker. Multivariate smoothing via the Fourier integral theorem and Fourier kernel. *Arxiv preprint Arxiv:2012.14482*, 2021.

[32] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[33] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020.

[34] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[35] Hyunjik Kim, George Papamakarios, and Andriy Mnih. The lipschitz constant of self-attention. In *International Conference on Machine Learning*, pages 5562–5571. PMLR, 2021.

[36] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. Structured attention networks. *arXiv preprint arXiv:1702.00887*, 2017.

[37] Devin Kreuzer, Dominique Beaini, Will Hamilton, Vincent Létourneau, and Prudencio Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021.

[38] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021.

[39] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[41] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[42] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[43] Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

[44] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[45] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[46] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018.

[47] Jiaqi Mu and Pramod Viswanath. All-but-the-top: Simple and effective postprocessing for word representations. In *International Conference on Learning Representations*, 2018.

[48] E.A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[49] Ankur Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, Austin, Texas, November 2016. Association for Computational Linguistics.

[50] E. Parzen. On estimation of a probability density function and mode. *Annals of Mathematical Statistics*, 33:1065–1076, 1962.

[51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[52] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[53] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *OpenAI report*, 2018.

[54] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[55] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[56] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[57] JN Reddy. *An introduction to the finite element method*, volume 1221. McGraw-Hill New York, 2004.

[58] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[59] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics*, 27:832–837, 1956.

[60] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[61] Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. Poor man's bert: Smaller and faster transformer models. *arXiv e-prints*, pages arXiv–2004, 2020.

[62] Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

[63] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[64] Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pages 9355–9366. PMLR, 2021.

[65] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[66] J.G. Staniswalis, K. Messer, and D.R. Finston. Kernel estimators for multivariate regression. *Journal of Nonparametric Statistics*, 3:103–121, 1993.

[67] Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*, 2019.

[68] Binh Tang and David S. Matteson. Probabilistic transformer for time series analysis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[69] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020.

[70] Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics.

[71] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[72] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[73] Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: An unified understanding for transformer's attention via the lens of kernel. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4344–4353, Hong Kong, China, November 2019. Association for Computational Linguistics.

[74] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-margin training: Scalable certification of perturbation invariance for deep neural networks. *Advances in neural information processing systems*, 31, 2018.

[75] A. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[77] Jesse Vig and Yonatan Belinkov. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy, August 2019. Association for Computational Linguistics.

[78] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics.

[79] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

[80] M.P. Wand. Error analysis for general multivariate kernel estimators. *Journal of Nonparametric Statistics*, 2:1–15, 1992.

[81] M.P. Wand and M.C. Jones. Comparison of smoothing parameterizations in bivariate kernel density estimation. *Journal of the American Statistical Association*, 88:520–528, 1993.

[82] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[83] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[84] L. Wasserman. *All of Nonparametric Statistics*. Springer, 2006.

[85] N. Wiener. *The Fourier Integral and Certain of its Applications*. Cambridge University Press, 1933.

[86] Haixu Wu, Jialong Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Flowformer: Linearizing transformers with conservation flows. In *International Conference on Machine Learning*, 2022.

[87] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019.

[88] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2114–2124, 2021.

[89] Shaolei Zhang and Yang Feng. Modeling concentrated cross-attention for neural machine translation with Gaussian mixture model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1401–1411, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[90] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021.

# Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [N/A]
    (b) Did you mention the license of the assets? [N/A]
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]