Designing Robust Transformers using Robust Kernel Density Estimation

Xing Han[†,⋆] Tongzheng Ren[†,⋆] Tan Minh Nguyen[⋄,⋆] Khai Nguyen[†] Joydeep Ghosh[†] Nhat Ho[†]

University of Texas, Austin[†]; University of California, Los Angeles[⋄]

February 13, 2023

Abstract

Recent advances in Transformer architectures have empowered their empirical success in a variety of tasks across different domains. However, existing works mainly focus on predictive accuracy and computational cost, without considering other practical issues, such as robustness to contaminated samples. Recent work [43] has shown that the self-attention mechanism, which is the center of the Transformer architecture, can be viewed as a non-parametric estimator based on kernel density estimation (KDE). This motivates us to leverage a set of robust kernel density estimation methods for alleviating the issue of data contamination. Specifically, we introduce a series of self-attention mechanisms that can be incorporated into different Transformer architectures and discuss the special properties of each method. We then perform extensive empirical studies on language modeling and image classification tasks. Our methods demonstrate robust performance in multiple scenarios while maintaining competitive results on clean datasets.

1 Introduction

Attention mechanisms and transformers [65] have drawn lots of attention in the machine learning community [29, 59, 25]. Now they are among the best deep learning architectures for a variety of applications, including those in natural language processing [12, 1, 9, 7, 49, 2, 4, 10], computer vision [14, 32, 60, 50, 46, 15, 33], and reinforcement learning [6, 23]. They are also known for their effectiveness in transferring knowledge from various pretraining tasks to different downstream applications with weak supervision or no supervision [47, 48, 12, 71, 31].

Despite these remarkable gains, the robustness of the conventional attention module remains an open question in the literature. Recent works [57, 45, 3] have mostly focused on the robustness of vision transformers (ViT) under various attacks. [36] empirically shows that ViT is vulnerable to white-box adversarial attacks but a simple ensemble defense can achieve unprecedented robustness without sacrificing accuracy on clean data. [37] performs robustness analysis on different building blocks of ViT and proposed position-aware attention scaling and patch-wise augmentation to improve the robustness and accuracy of ViT models. More recently, [73] proposed fully attentional networks to improve self-attention and achieved state-of-the-art accuracy on corrupted images. However, these works focus on improving the architectural design of ViT targeted for task-specific applications and lack a general framework for improving the robustness of transformers. They also introduce extra parameters. In addition, these works largely concentrate on vision-related tasks and cannot be generalized across other data modalities.

[⋆] Xing Han, Tongzheng Ren, and Tan Nguyen contributed equally to this work.

In this paper, to robustify the attention mechanism and build a general framework for robust transformer models, we first revisit the interpretation of the self-attention in transformer as the Nadaraya-Watson (NW) estimator [40] in a non-parametric regression setting. In the context of the transformer, the NW estimator is constructed based on the kernel density estimators (KDE) of the keys and queries. However, such KDEs are not robust to contaminated samples [26]. By viewing KDE as the solution to the kernel regression problem in a reproducing kernel Hilbert space (RKHS), we can adopt multiple state-of-the-art robust KDE methods based on e.g. robust kernel regression and median-of-mean estimator, to design substantially more robust self-attention mechanisms. The resulting family of robust self-attention mechanisms can be tailored to various transformer architectures and tasks in multiple data modalities.

We perform extensive experiments on both vision and language modeling tasks. Results demonstrate that our methods can have comparable accuracy on the clean data, with more favorable performance on the contaminated data over state-of-the-art robust transformer models, without introducing any extra parameters.

2 Self-Attention Mechanism from A Non-parametric Regression Perspective

In this section, we provide background on self-attention mechanism in transformer and its connection to the NW estimator in the non-parametric regression problem, which can be constructed via standard KDE. We then connect KDE to a kernel regression problem in RKHS and demonstrate that it is not robust to the contaminated samples.

2.1 Self-Attention Mechanism

Given an input sequence $\boldsymbol{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]^\top \in \mathbb{R}^{N \times D_x}$ of N feature vectors, the self-attention mechanism transforms it into another sequence $\mathbf{H} := [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N]^\top \in \mathbb{R}^{N \times D_v}$ as follows:

$$\boldsymbol{h}_i = \sum_{j \in [N]} \mathrm{softmax}\Big(\frac{\boldsymbol{q}_i^\top \boldsymbol{k}_j}{\sqrt{D}}\Big) \boldsymbol{v}_j, \text{ for } i = 1, \dots, N. \tag{1}$$

The vectors $\boldsymbol{q}_i$, $\boldsymbol{k}_j$ and $\boldsymbol{v}_j$ are the query, key and value vectors, respectively. They are computed as follows:

$$\begin{aligned} [\boldsymbol{q}_1, \boldsymbol{q}_2, \dots, \boldsymbol{q}_N]^\top &:= \boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}_Q^\top \in \mathbb{R}^{N \times D}, \\ [\boldsymbol{k}_1, \boldsymbol{k}_2, \dots, \boldsymbol{k}_N]^\top &:= \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}_K^\top \in \mathbb{R}^{N \times D}, \\ [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_N]^\top &:= \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}_V^\top \in \mathbb{R}^{N \times D_v}, \end{aligned} \tag{2}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{D \times D_x}$, $\boldsymbol{W}_V \in \mathbb{R}^{D_v \times D_x}$ are the weight matrices. Equation (1) can be written in the following equivalent matrix form:

$$\mathbf{H} = \mathrm{softmax}\Big(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D}}\Big)\boldsymbol{V}, \tag{3}$$

where the softmax function is applied to each row of the matrix $(\boldsymbol{Q}\boldsymbol{K}^\top)/\sqrt{D}$. Equation (3) is also called the “softmax attention”. Throughout this paper, we term a transformer built with softmax attention as the standard Transformer or Transformer.

2.2 A Non-parametric Regression Perspective of Self-Attention

We now briefly discuss the connection between the self-attention mechanism in equation (1) and the non-parametric regression. Assume we have the key and value vectors $\{\boldsymbol{k}_j, \mathbf{v}_j\}_{j\in[N]}$ that is collected from the data generating process

$$\mathbf{v} = f(\boldsymbol{k}) + \varepsilon, \tag{4}$$

where ε is some noise vectors with $\mathbb{E}[\varepsilon] = 0$, and f is the function that we want to estimate. We consider a random design setting where the key vectors $\{\boldsymbol{k}_j\}_{j\in[N]}$ are i.i.d. samples from the distribution $p(\boldsymbol{k})$, and we use $p(\mathbf{v}, \boldsymbol{k})$ to denote the joint distribution of $(\mathbf{v}, \boldsymbol{k})$ defined by equation (4). Our target is to estimate $f(\boldsymbol{q})$ for any new queries $\boldsymbol{q}$.

The NW estimator provides a non-parametric approach to estimate the function f, the main idea is that

$$f(\boldsymbol{k}) = \mathbb{E}[\mathbf{v}|\boldsymbol{k}] = \int_{\mathbb{R}^D} \mathbf{v} \cdot p(\mathbf{v}|\boldsymbol{k}) d\mathbf{v} = \int_{\mathbb{R}^D} \frac{\mathbf{v} \cdot p(\mathbf{v}, \boldsymbol{k})}{p(\boldsymbol{k})} d\mathbf{v}, \tag{5}$$

where the first equation comes from the fact that $\mathbb{E}[\varepsilon] = 0$, the second equation comes from the definition of conditional expectation and the last equation comes from the definition of the conditional density. To provide an estimation of f, we just need to obtain estimations for both the joint density function $p(\mathbf{v}, \boldsymbol{k})$ and the marginal density function $p(\boldsymbol{k})$. One popular approach for the density estimation problem is the kernel density estimation (KDE) [52, 44], which requires a kernel k_σ with the bandwidth parameter σ satisfies $\int_{\mathbb{R}^D} k_\sigma(\boldsymbol{x} - \boldsymbol{x}')d\boldsymbol{x} = 1, \forall \boldsymbol{x}'$, and estimate the density as

$$\hat{p}_\sigma(\mathbf{v}, \boldsymbol{k}) = \frac{1}{N} \sum_{j\in[N]} k_\sigma\left([\mathbf{v}, \boldsymbol{k}] - [\mathbf{v}_j, \boldsymbol{k}_j]\right) \tag{6}$$

$$\hat{p}_\sigma(\boldsymbol{k}) = \frac{1}{N} \sum_{j\in[N]} k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j), \tag{7}$$

where $[\mathbf{v}, \boldsymbol{k}]$ denotes the concatenation of $\mathbf{v}$ and $\boldsymbol{k}$. Specifically, when k_σ is the isotropic Gaussian kernel $k_\sigma(\boldsymbol{x} - \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/(2\sigma^2)\right)$, we have

$$\hat{p}_\sigma(\mathbf{v}, \boldsymbol{k}) = \frac{1}{N} \sum_{j\in[N]} k_\sigma(\mathbf{v} - \mathbf{v}_j) k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j). \tag{8}$$

Given the kernel density estimators of equations (7) and (8), as well as the formulation in equation (5), we can obtain the NW estimator of the function f as

$$\widehat{f}_\sigma(\boldsymbol{k}) = \frac{\sum_{j\in[N]} \mathbf{v}_j k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)}{\sum_{j\in[N]} k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)}. \tag{9}$$

Furthermore, [43] have shown that if the keys $\{\boldsymbol{k}_j\}_{j\in[N]}$ are normalized, the self-attention mechanism $\widehat{f}_\sigma(\boldsymbol{q}_i)$ in equation (9) is exactly the standard transformer

$$\widehat{f}_\sigma(\boldsymbol{q}_i) = \sum_{j\in[N]} \operatorname{softmax}\left(\boldsymbol{q}^\top \boldsymbol{k}_j/\sigma^2\right) \mathbf{v}_j. \tag{10}$$

Such an assumption on the normalized key $\{\boldsymbol{k}_j\}_{j\in[N]}$ can be mild, as in practice we always have an normalization step on the key to stabilize the training of the transformer [55]. If we choose $\sigma^2 = \sqrt{D}$, where D is the dimension of $\boldsymbol{q}$ and $\boldsymbol{k}_j$, then $\widehat{f}_\sigma(\boldsymbol{q}_i) = \boldsymbol{h}_i$. As a result, the self-attention mechanism in fact performs a non-parametric regression with NW-estimator and isotropic Gaussian kernel when the keys are normalized.

2.3 KDE as a Regression Problem in RKHS

We start from the formal definition of the RKHS. The space $\mathcal{H}_k = \{f \mid f : \mathcal{X} \to \mathbb{R}\}$ is called an RKHS associated with the kernel k, where $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, if it is a Hilbert space with inner product $\langle \cdot, \cdot \rangle_{\mathcal{H}_k}$ and following properties:

- $k(\boldsymbol{x}, \cdot) \in \mathcal{H}_k, \forall \boldsymbol{x} \in \mathcal{X}$;
- $\forall f \in \mathcal{H}_k$, $f(\boldsymbol{x}) = \langle f, k(\boldsymbol{x}, \cdot)\rangle_{\mathcal{H}_k}$. This is also known as the reproducing property.

With slightly abuse of notation, we define $k_\sigma(\boldsymbol{x}, \boldsymbol{x}') = k_\sigma(\boldsymbol{x} - \boldsymbol{x}')$. By the definition of the RKHS and the KDE estimator, we know $\hat{p}_\sigma = \frac{1}{N}\sum_{j\in[N]} k_\sigma(\boldsymbol{x}_j, \cdot) \in \mathcal{H}_{k_\sigma}$, and can be viewed as the optimal solution of the following least-square regression problem in RKHS:

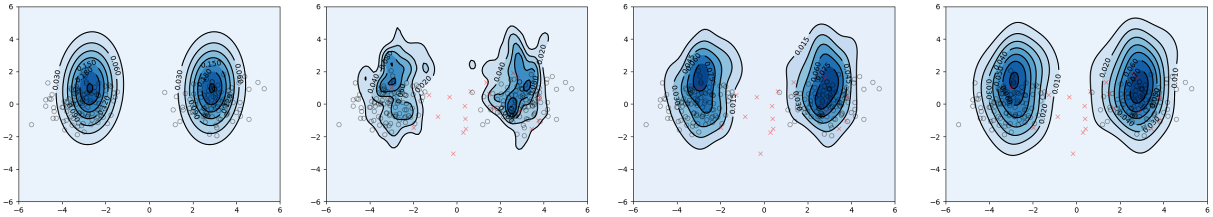
$$\hat{p}_\sigma = \operatorname*{arg\,min}_{p\in\mathcal{H}_{k_\sigma}} \sum_{j\in[N]} \frac{1}{N} \|k_\sigma(\boldsymbol{x}_j, \cdot) - p\|^2_{\mathcal{H}_{k_\sigma}}. \tag{11}$$

Note that, in equation (11), we have the same weight $1/N$ on each of the error $\|k_\sigma(\boldsymbol{x}_j, \cdot) - p\|^2_{\mathcal{H}_{k_\sigma}}$. This works well if there are no outliers in $\{k_\sigma(\boldsymbol{x}_j, \cdot)\}_{j\in[N]}$. However, when we have outliers (e.g., when there exists some j, such that $\|k_\sigma(\boldsymbol{x}_j, \cdot)\|_{\mathcal{H}_{k_\sigma}} \gg \|k_\sigma(\boldsymbol{x}_i, \cdot)\|_{\mathcal{H}_{k_\sigma}}, \forall i \in [N], i \neq j$), the error on the outliers will dominate the whole error and lead to substantially worse estimation on the entire density. We illustrate the robustness issue of the KDE in Figure 1.

Combining the viewpoint that KDE is not robust to outliers with the interpretation of Section 2.2 implies that the transformer is also not robust when there are outliers in the data. The robustness issue of transformer has mostly been studied in the vision domain, such as [36, 37, 73]. These works modify the original architectures of vision transformer and introduces extra parameters. A representative one is [37], which proposed position-based attention by adding on another fully connected layer. However, this approach will cause bi-directional information flow for positional-sensitive dataset such as text or sequences and is therefore limited to image data. We now take a different view of the robustness problem in the RKHS domain and provide a unified framework for different data modalities.

3 Robustify Transformer with Robust Kernel Density Estimation

We observe that variants of the vanilla kernel density estimation such as robust KDE [26], scaled projection KDE [64] and more recently median-of-means [22], can down-weight or filter out the potentially corrupted data and obtain a robust density estimator. We derive the corresponding robust version of the NW-estimator, followed by showing how to use this to strengthen the self-attention mechanism. We propose two types of robust self-attention mechanisms and discuss the properties of each method, which could lead to a more robust Transformer variant.



(a) (b) (c) (d)

Figure 1: Contour plots of density estimation of the 2-dimensional query vector embedding in an attention layer of the transformer when using (b) KDE (equation (11)) and (c) RKDE after one iteration of equation 15 with Huber loss (equation (13)), (d) KDE with median-of-means principle (equation 19), where (a) is the true density function. We draw 1000 samples (gray circles) from a multivariate normal density and 100 outliers (red cross) from a gamma distribution as the contaminating density. RKDE and KDE with median-of-means principle can be less affected by contaminated samples when computing self-attention as nonparametric regression.

3.1 Down-weighting Outliers in RKHS

Robust KDE Motivated by the robust regression [16], [26] proposed a robust version of KDE, by replacing the least-square loss in equation (11) with a robust loss function ρ:

$$\hat{p}_{\text{robust}} = \underset{p \in \mathcal{H}_{k_\sigma}}{\arg\min} \sum_{j \in [N]} \rho\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - p\|_{\mathcal{H}_{k_\sigma}}\right). \tag{12}$$

Examples of the robust loss functions ρ include the Huber loss [21], Hampel loss [19], Welsch loss [69] and Tukey loss [16]. We empirically evaluate different loss functions in our experiments. For simplicity, we use the Huber loss function as the demonstrating example, which is defined as follows:

$$\rho(x) := \begin{cases} x^2/2, & 0 \le x \le a \\ ax - a^2/2, & a < x, \end{cases} \tag{13}$$

where a is a constant. The solution of this robust regression problem has the following form:

Proposition 1. *Assume the robust loss function ρ is non-decreasing in $[0, \infty]$, $\rho(0) = 0$ and $\lim_{x \to 0} \frac{\rho(x)}{x} = 0$. Define $\psi(x) := \frac{\rho'(x)}{x}$ and assume $\psi(0) = \lim_{x \to 0} \frac{\rho'(x)}{x}$ exists and finite. Then the optimal $\hat{p}_{robust}$ can be written as*

$$\hat{p}_{robust} = \sum_{j \in [N]} \omega_j k_\sigma(\boldsymbol{x}_j, \cdot),$$

where $\omega = (\omega_1, \cdots, \omega_N) \in \Delta_N$, with each $\omega_j \propto \psi\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - \hat{p}_{robust}\|_{\mathcal{H}_{k_\sigma}}\right)$. Here Δ_n denotes the n-dimensional probability simplex.

The proof of this proposition can be found in Appendix A. For the Huber loss function, we have that

$$\psi(x) := \begin{cases} 1, & 0 \le x \le a \\ a/x, & a < x. \end{cases}$$

Hence, when the error $\|k_\sigma(\boldsymbol{x}_j,\cdot), \cdot - \hat{p}_{\text{robust}}\|_{\mathcal{H}_{k_\sigma}}$ is over the threshold a, the final estimator will down-weight the importance of $k_\sigma(\boldsymbol{x}_j,\cdot)$. This is in sharp contrast with the standard KDE method, which will assign uniform weights to all of the $k_\sigma(\boldsymbol{x}_j,\cdot)$. One additional issue is that, the estimator provided in Proposition 1 is circularly defined, as $\hat{p}_{\text{robust}}$ is defined via ω, and ω depends on $\hat{p}_{\text{robust}}$. Such an issue can be addressed by estimating ω with an iterative algorithm termed as kernelized iteratively re-weighted least-squares (KIRWLS) algorithm. The algorithm starts with some randomly initialized $\omega^{(0)} \in \Delta_n$, and perform the following iterative updates between two steps:

$$\hat{p}_{\text{robust}}^{(k)} = \sum_{j\in[N]} \omega_i^{(k-1)} k_\sigma(\boldsymbol{x}_j,\cdot), \tag{14}$$

$$\omega_j^{(k)} = \frac{\psi\left(\left\|k_\sigma(\boldsymbol{x}_j,\cdot) - \hat{p}_{\text{robust}}^{(k)}\right\|_{\mathcal{H}_{k_\sigma}}\right)}{\sum_{j\in[N]} \psi\left(\left\|k_\sigma(\boldsymbol{x}_j,\cdot) - \hat{p}_{\text{robust}}^{(k)}\right\|_{\mathcal{H}_{k_\sigma}}\right)}. \tag{15}$$

Note that, the optimal $\hat{p}_{\text{robust}}$ is the fixed point of this iterative update, and the KIRWLS algorithm converges under standard regularity conditions. Furthermore, one can directly compute the term $\left\|k_\sigma(\boldsymbol{x}_j,\cdot) - \hat{p}_{\text{robust}}^{(k)}\right\|_{\mathcal{H}_{k_\sigma}}$ via the reproducing property:

$$\begin{aligned}\left\|k_\sigma(\boldsymbol{x}_j,\cdot) - \hat{p}_{\text{robust}}^{(k)}\right\|_{\mathcal{H}_{k_\sigma}}^2 &= -2\sum_{m\in[N]} \omega_m^{(k-1)} k_\sigma(\boldsymbol{x}_m,\boldsymbol{x}_j) \\ + k_\sigma(\boldsymbol{x}_j,\boldsymbol{x}_j) + &\sum_{m\in[N],n\in[N]} \omega_m^{(k-1)}\omega_n^{(k-1)} k_\sigma(\boldsymbol{x}_m,\boldsymbol{x}_n).\end{aligned}$$

Therefore, the weights can be updated without mapping the data to the Hilbert space.

Scaled Projection KDE Scaled and Projected KDE (SPKDE) [64] is one other option of robust KDE in the RKHS space. It essentially scales the original KDE and projects it to its nearest weighted KDE in the L_2 norm. The resulting weighted KDE can allocate more weight to high-density regions and truncate the weights for anomalous samples. Specifically, given the scaling factor $\beta > 1$, and let $\mathcal{C}_\sigma^N$ be the convex hull of $k_\sigma(\boldsymbol{x}_1,\cdot),\ldots,k_\sigma(\boldsymbol{x}_N,\cdot) \in \mathcal{H}_{k_\sigma}$, i.e., the space of weighted KDEs, the optimal density $\hat{p}_{\text{robust}}$ is given by

$$\hat{p}_{\text{robust}} = \arg\min_{p\in\mathcal{C}_\sigma^N} \left\|\frac{\beta}{N}\sum_{j\in[N]} k_\sigma(x_j,\cdot) - p\right\|_{\mathcal{H}_{k_\sigma}}^2, \tag{16}$$

which is guaranteed to have a unique minimizer since we are projecting in a Hilbert space and $\mathcal{C}_\sigma^N$ is closed and convex. Notice that, by definition, $\hat{p}_{\text{robust}}$ can also be represented as $\hat{p}_{\text{robust}} = \sum_{j\in[N]} \omega_j k_\sigma(x_j,\cdot)$, $\omega \in \Delta^N$, which is same as the formulation in Proposition 1. Then equation (16) can be written as a quadratic programming (QP) problem over ω. Let G be the Gram matrix of $\{\boldsymbol{x}_j\}_{j\in[N]}$ with k_σ and $q = \boldsymbol{G}\mathbf{1}\frac{\beta}{N}$, then the QP can be written as follows

$$\min_\omega \omega^\top G\omega - 2q^\top\omega, \quad \text{subject to } \omega \in \Delta^N. \tag{17}$$

Since k_σ is a positive-definite kernel and each $\boldsymbol{x}_i$ is unique, the Gram matrix G is also positive-definite. As a result, this QP problem is convex, and we can leverage commonly used solvers to efficiently obtain the solution and the optimal density $\hat{p}_{\text{robust}}$.

Robust Self-Attention Mechanism We now propose the robust self-attention mechanism via weighting samples. We consider the density estimator of the joint distribution and the marginal distribution from the robust KDE:

$$\hat{p}_{\text{robust}}(\mathbf{v}, \boldsymbol{k}) = \sum_{j \in [N]} \omega_j^{\text{joint}} k_\sigma([\mathbf{v}_j, \boldsymbol{k}_j], [\mathbf{v}, \boldsymbol{k}]),$$
$$\hat{p}_{\text{robust}} = \sum_{j \in [N]} \omega_j^{\text{marginal}} k_\sigma(\boldsymbol{k}_j, \boldsymbol{k}).$$

With a similar computation, the robust self-attention mechanism we use is defined as

$$\widehat{\boldsymbol{h}}_i = \frac{\sum_{j \in [N]} \mathbf{v}_j \omega_j^{\text{joint}} k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j)}{\sum_{j \in [N]} \omega_j^{\text{marginal}} k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j)}, \tag{18}$$

where ω^{joint} and ω^{marginal} are obtained via either the KIRWLS algorithm or results from the QP solver. We term the transformer models that employ robust KDE and SPKDE as Transformer-RKDE and Transformer-SPKDE.

Remark. *Note that, the computation of* $\{\omega_j^{marginal}\}_{j \in [N]}$ *and* $\{\omega_j^{joint}\}_{j \in [N]}$ *are separate as* ω_j^{joint} *involves both keys and values vectors. During the empirical evaluation, we concatenate the keys and values along the head dimension to obtain the weights for the joint density* $\hat{p}_{robust}(\mathbf{v}, \boldsymbol{k})$ *and only use the key vectors for obtaining the set of weights for the marginal* $\hat{p}_{robust}(\boldsymbol{k})$*. In addition,* $\omega^{marginal}, \omega^{joint} \in \mathbb{R}^{j \times i}$ *for* $i, j = 1, \ldots, N$ *are 2-dimensional matrices that include the pairwise weights between each position of the sequence and the rest of the positions. The weights are initialized uniformly across a certain sequence length dimension. For experiments related to language modeling, we can leverage information from the attention mask to initialize the weights on the unmasked part of the sequence.*

3.2 Median-of-Means Principle

Methods to down-weight outliers are effective, but they require iterative algorithms to compute the set of weights, which increases the overall complexity. The Median-of-Means (MoM) method is one other way to construct robust estimators while addressing the drawback of the above approaches [22]. Specifically, we randomly divide the keys $\{\boldsymbol{k}_j\}_{j=1}^N$ into B subsets $I_1, \ldots, I_B$ of equal size, namely, $|I_1| = |I_2| = \ldots = |I_B| = \mathcal{S}$. Then, the robust estimator of $p(\boldsymbol{k})$ takes the following form:

$$\hat{p}_{\text{robust}}(\boldsymbol{k}) \propto \text{median}\left\{\hat{p}_{\sigma, I_1}(\boldsymbol{k}), \ldots, \hat{p}_{\sigma, I_B}(\boldsymbol{k})\right\}, \tag{19}$$

where we define $\hat{p}_{\sigma, I_l}(\boldsymbol{k}) = \frac{1}{\mathcal{S}} \sum_{j \in I_l} k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)$ for any $l \in [B]$. KDE with the MoM principle has demonstrated its statistical performance under a less restrictive outlier framework and can be easily adapted to self-attention.

Similarly, the robust estimator of $p(\mathbf{v}, \boldsymbol{k})$ is as follows:

$$\hat{p}_{\text{robust}}(\mathbf{v}, \boldsymbol{k}) \propto \text{median}\left\{\hat{p}_{\sigma, I_1}(\mathbf{v}, \boldsymbol{k}), \ldots, \hat{p}_{\sigma, I_B}(\mathbf{v}, \boldsymbol{k})\right\}, \tag{20}$$

where $\hat{p}_{\sigma, I_l}(\mathbf{v}, \boldsymbol{k}) = \frac{1}{\mathcal{S}} \sum_{j \in I_l} k_\sigma(\mathbf{v} - \mathbf{v}_j) k_\sigma(\boldsymbol{k} - \boldsymbol{k}_j)$ for any $l \in [B]$. We now propose the self-attention mechanism utilizing the median-of-means principle.

Algorithm 1 Procedure of Computing Attention Vector of Transformer-RKDE/SPKDE/MoM

```
**Input**: $\mathbf{Q} = \{\boldsymbol{q}_i\}_{i\in[N]}$, $\mathbf{K} = \{\boldsymbol{k}_j\}_{j\in[N]}$, $\mathbf{V} = \{\mathbf{v}_l\}_{l\in[N]}$, initial weights $\omega^{(0)}$
Normalize $\mathbf{K} = \{\boldsymbol{k}_j\}_{j\in[N]}$ along the head dimension.
Compute kernel function between each pair of sequence: $k_\sigma(\mathbf{Q}, \mathbf{K}) = \{k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j)\}_{i,j\in[N]}$.
(Optional) apply attention mask on $k_\sigma(\mathbf{Q}, \mathbf{K})$.
**[MoM]** Randomly sample $B$ subsets $I_1, \ldots, I_B$ of size $\mathcal{S}$, obtain the median block $I_l$ such that $\frac{1}{\mathcal{S}}\sum_{j\in I_l} k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j) = \text{median}\{\frac{1}{\mathcal{S}}\sum_{j\in I_1} k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j), \ldots, \frac{1}{\mathcal{S}}\sum_{j\in I_B} k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j)\}$
**[RKDE]** Update weights $\omega^{(0)}$ for marginal/joint density by $\omega_j^{(1)} = \frac{\psi\left(\left\|k_\sigma(\boldsymbol{k}_j,\cdot)-\hat{p}^{(k)}_{\text{robust}}(\boldsymbol{k})\right\|_{\mathcal{H}_{k_\sigma}}\right)}{\sum_{j\in[N]}\psi\left(\left\|k_\sigma(\boldsymbol{k}_j,\cdot)-\hat{p}^{(k)}_{\text{robust}}(\boldsymbol{k})\right\|_{\mathcal{H}_{k_\sigma}}\right)}$.
**[SPKDE]** Obtain optimal weights $\omega$ for marginal/joint density via solving equation (17).
**[RKDE, SPKDE]** Obtain robust self-attention vector $\widehat{\boldsymbol{h}}_i = \frac{\sum_{j\in[N]}\mathbf{v}_j\omega_j^{\text{joint}}k_\sigma(\boldsymbol{q}_i-\boldsymbol{k}_j)}{\sum_{j\in[N]}\omega_j^{\text{marginal}}k_\sigma(\boldsymbol{q}_i-\boldsymbol{k}_j)}$.
**[MoM]** Obtain attention vector $\widehat{\boldsymbol{h}}_i = \frac{\sum_{j\in I_l} v_j k_\sigma(\boldsymbol{q}_i-\boldsymbol{k}_j)}{\sum_{j\in I_l} k_\sigma(\boldsymbol{q}_i-\boldsymbol{k}_j)}$.
```

Median-of-Means Self-Attention Mechanism Given the robust estimators in equations (19) and (20), we can consider the following robust estimation of the attention:

$$\widehat{\boldsymbol{h}}_i = \frac{\frac{1}{\mathcal{S}}\sum_{j\in I_l} v_j k_\sigma(\boldsymbol{q}_i - \boldsymbol{k}_j)}{\text{median}\left\{\hat{p}_{\sigma,I_1}(\boldsymbol{q}_i - \boldsymbol{k}), \ldots, \hat{p}_{\sigma,I_B}(\boldsymbol{q}_i - \boldsymbol{k})\right\}}, \tag{21}$$

where I_l is the block such that $\hat{p}_\sigma(\boldsymbol{q}_i - \boldsymbol{k})$ achieves its median value in equation (20). Note that, the reason that we choose the block I_l for the numerator instead of considering the median over all possible blocks is due to computational efficiency.

Remark. *Here, the random subsets are over input sequences instead of data points, which is different from that of stochastic batches. The original MoM principle requires each subset to be non-overlapped: i.e. $I_{l_1} \cap I_{l_2} = \emptyset$ for any $1 \le l_1 \ne l_2 \le B$. However, for structured data in high-dimension, dividing into non-overlapping blocks will result in the model only having a partial view of the dataset, leading to sub-optimal performance. We therefore construct each subset by sampling with replacement from the original dataset and retain the sequential relationship after that. Notice that, our proposed attention mechanism assumes key and query vectors achieve their median on the same block and therefore applies the median block obtained from the denominator into the numerator, which is faster than computing median blocks on both sides.*

3.3 Practical Implementation

The two types of robust attention mechanisms we proposed have their respective strengths. To speed up the computation for Transformer-RKDE, we use a single-step iteration on equation (15) to approximate the optimal set of weights. Empirical results have shown that this one-step iteration can achieve sufficiently accurate results. For Transformer-SPKDE, since the optimal set of weights is obtained via the QP solver, it requires longer computation time but leads to better performance on both clean and adversarial data. As an alternative to weight-based methods, Transformer-MoM is much more efficient and demonstrates competitive performance, especially on text data. The full procedure of computing the attention vector for Transformer-RKDE, Transformer-SPKDE, and Transformer-MOM can be found at Algorithm 1.

Table 1: Perplexity (PPL) and negative likelihood loss (NLL) of our methods and baselines on WikiText-103 dataset. The best results are highlighted in bold font and the second best results are highlighted in underline. Transformer-MoM and Transformer-SPKDE achieve competitive performance to the baseline methods while shows much better PPL and NLL under random swap with outlier words.

Method	Clean Data		Word Swap	
	Valid PPL/Loss	Test PPL/Loss	Valid PPL/Loss	Test PPL/Loss
Transformer [66]	33.52/3.51	34.59/3.54	72.28/4.45	74.56/4.53
Performer [8]	33.35/3.51	34.49/3.54	69.78/4.38	71.03/4.41
Transformer-MGK [42]	33.28/3.51	34.21/3.53	71.64/4.42	73.48/4.49
Transformer-KDE	33.34/3.51	34.37/3.54	71.94/4.43	73.75/4.49
Transformer-RKDE (Huber)	33.22/3.50	34.29/3.54	52.14/3.92	55.68/3.99
Transformer-RKDE (Hampel)	33.24/3.50	34.35/3.54	55.61/3.98	57.92/4.03
Transformer-SPKDE	**33.05/3.49**	**34.18/3.53**	51.36/3.89	54.97/3.96
Transformer-MoM	33.56/3.51	34.68/3.55	**48.29/3.82**	**52.14/3.92**

4 Experimental Results

In this section, we empirically validate the advantage of our proposed transformer integrated with robust KDE attention (Transformer-RKDE/SPKDE/MoM) over the standard softmax transformer and its nonparametric regression variant (Transformer-KDE in equation (9)) on two large-scale datasets: language modeling on WikiText-103 dataset [38] (Section 4.1) and image classification on Imagenet [53, 11] and Imagenet-C [20] (Section 4.2). We also compare the proposed series of robust transformers with state-of-the-art models, including Performer [8], MGK [41], RVT [37], and FourierFormer [43]. Our experiments have shown that: (1) Transformer with robust KDE attention can reach competitive performance with baseline methods on a variety of tasks with different data modalities, this can be achieved without modifying the model architecture or introducing extra parameters; (2) the advantage of Transformer with robust KDE attention is more prominent when there is contamination of samples in either text or image data. All of our experiments are performed on the NVIDIA A-100 GPUs. For each experiment, we compare Transformer-RKDE/SPKDE/MoM with other baselines under the same hyper-parameter configurations.

4.1 Robust Language Modeling

Dataset: WikiText-103 is a language modeling dataset that contains collection of tokens extracted from good and featured articles from Wikipedia, which is suitable for models that can leverage long-term dependencies. The dataset contains around $268K$ words and its training set consists of about $28K$ articles with $103M$ tokens, this corresponds to text blocks of about 3600 words. The validation set and test sets consist of 60 articles with $218K$ and $246K$ tokens respectively. We follow the standard configurations in [38, 55] and splits the training data into L-word independent long segments. During evaluation, we process the text sequence using a sliding window of size L and feed into the model with a batch size of 1. The last position of the sliding window is used for computing perplexity except in the first segment, where all positions are evaluated as in [1, 55].

Implementation Details: We used the small version of language models developed by [55]

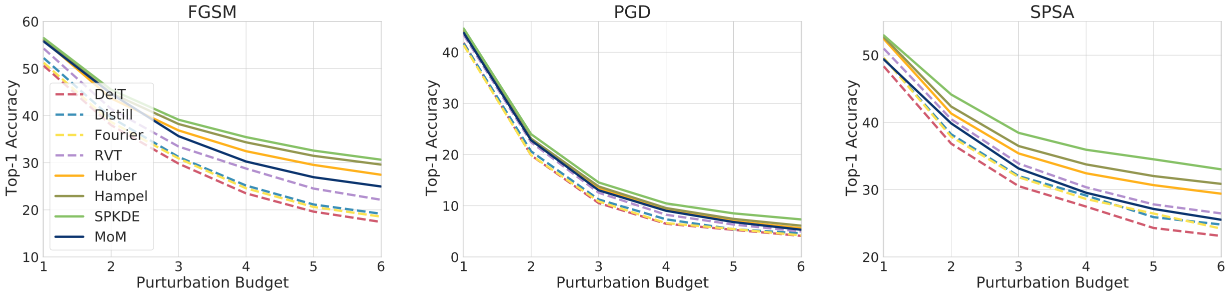
Table 2: Top-1, top-5 accuracy (%) and mean corruption error (mCE) of DeiT with different attention mechanisms. The best results are highlighted in bold font and the second best are highlighted in underlines. RVT [37] and DeiT with Distillation [61] achieves better results on clean data and corrupted imagenet; the proposed DeiT with robust KDE attention hold stronger defense under different adversarial attacks while still achieve competitive performance on clean imagenet.

Method	Clean Data		FGSM		PGD		SPSA		Imagenet-C	
	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	Top 5	Top 1	mCE$\downarrow$
DeiT [13]	72.23	91.13	52.61	82.26	41.84	76.49	48.34	79.36	42.38	71.14
Distill [61]	74.32	93.72	53.24	84.07	41.72	76.43	49.56	80.14	43.29	70.26
FourierFormer [43]	73.25	91.66	53.08	83.95	41.34	76.19	48.79	79.57	42.47	71.07
RVT [37]	**74.37**	**93.89**	53.67	84.11	43.39	77.26	51.43	80.98	**45.64**	**68.57**
DeiT-KDE	72.58	91.34	52.25	81.52	41.38	76.41	48.61	79.68	42.63	70.78
DeiT-RKDE (Huber)	72.83	91.44	55.83	85.89	44.15	79.06	52.42	82.03	45.58	68.69
DeiT-RKDE (Hampel)	72.94	91.63	55.92	85.97	44.23	79.16	52.48	82.07	45.61	68.67
DeiT-SPKDE	73.22	91.95	**56.03**	**86.12**	**44.51**	**79.47**	**52.64**	**82.33**	44.76	69.34
DeiT-MoM	71.94	91.08	55.76	85.23	43.78	78.85	49.38	80.02	45.16	69.11

in our experiments. The dimensions of key, value, and query are set to 128, and the training and evaluation context length is set to 256. We compare our methods with Performer [8] and Transformer-MGK [41], which have recently achieved state-of-the-art performance on this task. As for self-attention, we set the number of heads as 8 for our methods and Performer, and 4 for Transformer-MGK. We set the dimension of the feed-forward layer as 2048, and the number of layers as 16. To avoid numerical instability, we apply the `log-sum-exp` trick in equation (9) when computing the attention probability vector through the Gaussian kernel. We apply similar tricks when computing the weights of the KIRWLS algorithm, where we first obtain the weights in `log` space, followed by the `log-sum-exp` trick to compute robust self-attention as in equation (18). For Transformer-MoM, the sampled subset sequences account for 80% length of the original sequence.

Results: In Table 1, we report the validation and test PPL of Transformer-RKDE (with Huber and Hampel loss functions), Transformer-SPKDE and Transformer-MoM versus the above mentioned baselines. Based on the derivation in equation (10), we would expect Transformer-KDE to have similar performance with softmax transformer. Meanwhile, Transformer-RKDE and SPKDE is able to improve baselines PPL and NLL in both validation and test sets, while Transformer-MoM shows slightly higher perplexity due to the fact that only part of the sequence is used.

We can observe more obvious improvement when the dataset is under a word swap attack, which randomly replaces selected keywords of input data with a generic token "*AAA*" during evaluation. Our method, particularly MoM-based robust attention, achieves much better results for filter out rare words, where the median trick has demonstrated its effectiveness. We also observed more robust performance from RKDE/SPKDE-based robust attention than other baseline methods that have not been protected from the attack. Our implementation of word swap is based on the public



(a) (b) (c)

Figure 2: The top-1 classification *accuracy v.s. perturbation budget* × *255* curves on ImageNet against three untargeted attack methods under the l_∞ norm. Among all the competing methods, the proposed set of DeiT with robust KDE attention mechanisms shows stronger defense under all attack methods with different perturbation budgets.

code TextAttack by [39], while we use the greedy search method with the constraints on stop-words modification from the TextAttack library.

4.2 Image Classification under Adversarial Attack

Dataset: We use the full ImageNet dataset that contains $1.28M$ training images and $50K$ validation images. The model learns to predict the class of the input image among 1000 categories. We report the top-1 and top-5 accuracy on all experiments. For robustness on common image corruptions, we use ImageNet-C [20] which consists of 15 types of algorithmically generated corruptions with five levels of severity. ImageNet-C uses the mean corruption error (mCE) as metric: the smaller mCE means the more robust the model under corruption.

Implementation Details: Our method uses the same training configurations as DeiT-Tiny [61]. Given that all our approaches do not modify the model architecture, each employed model has $5.7M$ parameters. We also implemented state-of-the-art methods including DeiT with hard distillation [61], FourierFormer [43] and robust vision transformer (RVT) model [37] as our baselines. Note that, for a fair comparison with RVT, we only implemented its position-aware attention scaling without further modifications to the model architecture. The resulting model has around $7.2M$ parameters. To evaluate adversarial robustness, we apply adversarial examples generated by untargeted white-box attacks including single-step attack method FGSM [18], multi-step attack method PGD [35] and score-based black-box attack method SPSA [63]. The attacks are applied on 100% of the validation set of ImageNet. Both these attacks perturb the input image with perturbation budget $\epsilon = 1/255$ under l_∞ norm; while PGD attack uses 20 steps with step size $\alpha = 0.15$.

Results: We summarize the results in Table 2. Corresponding to the original papers, RVT and DeiT-Distillation achieve better performance on clean imagenet. The proposed series of DeiT with robust KDE attention can also obtain very close results with RVT and DeiT-Distillation under these settings. They can also evidently improve RVT under multiple types of adversarial attacks, especially for DeiT-SPKDE method. Figure 2 shows the relationship between accuracy versus perturbation budget using three attack methods. We observe that, the series of transformers with

Implementation available at github.com/QData/TextAttack robust self-attention mechanism have distinctly stronger defense under different perturbation budgets and exhibits greater advantage with higher perturbation strength. We provide more ablation studies in Appendix B regarding to different design choices of each of the proposed robust KDE attention.

5 Related Works

Robustness of Transformer: Vision Transformer (ViT) models [13, 61] recently achieved exemplary performance on a variety of vision tasks that can be used as a strong alternative to CNNs. To ensure its generalization ability on different datasets, many works [36, 37, 73] have proposed solutions to improve the defense of common adversarial attacks on image data, including ensemble defense by [36], position-aware attention scaling and patch-wise augmentation by [37], and fully attentional networks by [73]. Apart from this line of work, robust transformers have also been studied in domains such as text analysis and social media. [70] investigated table understanding and proposed a robust and structurally aware table-text encoding architecture to avoid row and column order perturbations. [30] proposed a robust end-to-end transformer-based model for crisis detection and crisis recognition. In addition, [28] designed a novel attention mechanism to construct a robust neural text-to-speech model to synthesize both natural and stable audios. Despite their strong performance, these works focus on architecture design for application-specific tasks and cannot generalize to all situations.

Theoretical Frameworks of Attention Mechanisms: Attention mechanisms in transformers have been recently studied from different perspectives. [62] shows that attention can be derived from smoothing the inputs with appropriate kernels. [24, 8, 67] further linearize the softmax kernel in attention to attain a family of efficient transformers with both linear computational and memory complexity. These linear attentions are proven in [5] to be equivalent to a Petrov-Galerkin projection [51], thereby indicating that the softmax normalization in dot-product attention is sufficient but not necessary. Other frameworks for analyzing transformers that use ordinary/partial differential equations include [34, 54]. In addition, the Gaussian mixture model and graph-structured learning have been utilized to study attentions and transformers [58, 17, 72, 68, 56, 27].

6 Conclusion and Future Work

In this paper, via the connection between the dot-product self-attention mechanism used in transformers with nonparametric kernel regression, we developed a family of robustified transformers by leveraging robust kernel density estimation as a replacement for dot-product attention to alleviate the effects of contaminated samples. We proposed two types of robust self-attention mechanisms that either down-weight or filter out the potentially corrupted data. The procedure requires iteratively computing a set of weights or obtaining the median block over subsets of sequences: both approaches can be flexibly integrated into commonly used transformer models. Empirical evaluations show that these robust transformer models can improve performance on clean data while demonstrating robust results under various attacks for both vision and language modeling tasks. The robust KDE attention we have developed generalizes to the whole family of transformer models. We are currently investigating potentially more efficient approaches to estimating the set of weights for robust kernel density estimation for large models.

References

[1] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones. Character-level language modeling with deeper self-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3159–3166, 2019. (Cited on pages 1 and 9.)

[2] A. Baevski and M. Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019. (Cited on page 1.)

[3] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10231–10241, 2021. (Cited on page 1.)

[4] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. (Cited on page 1.)

[5] S. Cao. Choose a transformer: Fourier or galerkin. *Advances in Neural Information Processing Systems*, 34, 2021. (Cited on page 12.)

[6] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021. (Cited on page 1.)

[7] R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. (Cited on page 1.)

[8] K. M. Choromanski, V. Likhosherstov, D. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. B. Belanger, L. J. Colwell, and A. Weller. Rethinking attention with performers. In *International Conference on Learning Representations*, 2021. (Cited on pages 9, 10, and 12.)

[9] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. (Cited on page 1.)

[10] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019. (Cited on page 1.)

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. (Cited on page 9.)

[12] J. Devlin, M.-W. Chang, K. Lee, and K. 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. (Cited on page 1.)

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. (Cited on pages 10 and 12.)

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. (Cited on page 1.)

[15] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. (Cited on page 1.)

[16] J. Fox and S. Weisberg. Robust regression. *An R and S-Plus companion to applied regression*, 91, 2002. (Cited on page 5.)

[17] P. Gabbur, M. Bilkhu, and J. Movellan. Probabilistic attention for interactive segmentation. *Advances in Neural Information Processing Systems*, 34, 2021. (Cited on page 12.)

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. (Cited on page 11.)

[19] F. R. Hampel, E. M. Ronchetti, P. Rousseeuw, and W. A. Stahel. *Robust statistics: the approach based on influence functions*. Wiley-Interscience; New York, 1986. (Cited on page 5.)

[20] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. (Cited on pages 9 and 11.)

[21] P. J. Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. (Cited on page 5.)

[22] P. Humbert, B. Le Bars, and L. Minvielle. Robust kernel density estimation with median-of-means principle. In *International Conference on Machine Learning*, pages 9444–9465. PMLR, 2022. (Cited on pages 4 and 7.)

[23] M. Janner, Q. Li, and S. Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021. (Cited on page 1.)

[24] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR, 2020. (Cited on page 12.)

[25] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021. (Cited on page 1.)

[26] J. Kim and C. Scott. Robust kernel density estimation. *The Journal of Machine Learning Research*, 13:2529–2565, 2012. (Cited on pages 2, 4, 5, 19, and 20.)

[27] D. Kreuzer, D. Beaini, W. Hamilton, V. Létourneau, and P. Tossou. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34, 2021. (Cited on page 12.)

[28] N. Li, Y. Liu, Y. Wu, S. Liu, S. Zhao, and M. Liu. Robutrans: A robust transformer-based text-to-speech model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8228–8235, 2020. (Cited on page 12.)

[29] T. Lin, Y. Wang, X. Liu, and X. Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021. (Cited on page 1.)

[30] J. Liu, T. Singhal, L. T. Blessing, K. L. Wood, and K. H. Lim. Crisisbert: a robust transformer for crisis classification and contextual crisis embedding. In *Proceedings of the 32nd ACM Conference on Hypertext and Social Media*, pages 133–141, 2021. (Cited on page 12.)

[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. (Cited on page 1.)

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. (Cited on page 1.)

[33] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. (Cited on page 1.)

[34] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019. (Cited on page 12.)

[35] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. (Cited on page 11.)

[36] K. Mahmood, R. Mahmood, and M. Van Dijk. On the robustness of vision transformers to adversarial examples. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7838–7847, 2021. (Cited on pages 1, 4, and 12.)

[37] X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, and H. Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. (Cited on pages 1, 4, 9, 10, 11, and 12.)

[38] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. (Cited on page 9.)

[39] J. X. Morris, E. Lifland, J. Y. Yoo, J. Grigsby, D. Jin, and Y. Qi. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. *arXiv preprint arXiv:2005.05909*, 2020. (Cited on page 11.)

[40] E. A. Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964. (Cited on page 2.)

[41] T. Nguyen, T. Nguyen, H. Do, K. Nguyen, V. Saragadam, M. Pham, K. Nguyen, N. Ho, and S. Osher. Improving transformer with an admixture of attention heads. In *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 9 and 10.)

[42] T. Nguyen, T. Nguyen, D. Le, K. Nguyen, A. Tran, R. Baraniuk, N. Ho, and S. Osher. Improving transformers with probabilistic attention keys. In *International Conference on Machine Learning*, 2022. (Cited on page 9.)

[43] T. Nguyen, M. Pham, T. Nguyen, K. Nguyen, S. J. Osher, and N. Ho. Fourierformer: Transformer meets generalized Fourier integral theorem. *Advances in Neural Information Processing Systems*, 2022. (Cited on pages 1, 3, 9, 10, and 11.)

[44] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962. (Cited on page 3.)

[45] S. Paul and P.-Y. Chen. Vision transformers are robust learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2071–2081, 2022. (Cited on page 1.)

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. (Cited on page 1.)

[47] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. *OpenAI report*, 2018. (Cited on page 1.)

[48] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. (Cited on page 1.)

[49] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. 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. (Cited on page 1.)

[50] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021. (Cited on page 1.)

[51] J. Reddy. *An introduction to the finite element method*, volume 1221. McGraw-Hill New York, 2004. (Cited on page 12.)

[52] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837, 1956. (Cited on page 3.)

[53] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. (Cited on page 9.)

[54] M. E. Sander, P. Ablin, M. Blondel, and G. Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022. (Cited on page 12.)

[55] I. Schlag, K. Irie, and J. Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pages 9355–9366. PMLR, 2021. (Cited on pages 4 and 9.)

[56] P. Shaw, J. Uszkoreit, and A. 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. (Cited on page 12.)

[57] A. Subramanya, A. Saha, S. A. Koohpayegani, A. Tejankar, and H. Pirsiavash. Backdoor attacks on vision transformers. *arXiv preprint arXiv:2206.08477*, 2022. (Cited on page 1.)

[58] B. Tang and D. S. Matteson. Probabilistic transformer for time series analysis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. (Cited on page 12.)

[59] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*, 2020. (Cited on page 1.)

[60] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. (Cited on page 1.)

[61] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. (Cited on pages 10, 11, and 12.)

[62] Y.-H. H. Tsai, S. Bai, M. Yamada, L.-P. Morency, and R. 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, Nov. 2019. Association for Computational Linguistics. (Cited on page 12.)

[63] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018. (Cited on page 11.)

[64] R. A. Vandermeulen and C. Scott. Robust kernel density estimation by scaling and projection in hilbert space. *Advances in Neural Information Processing Systems*, 27, 2014. (Cited on pages 4 and 6.)

[65] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. (Cited on page 1.)

[66] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. (Cited on page 9.)

[67] S. Wang, B. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. (Cited on page 12.)

[68] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. (Cited on page 12.)

[69] R. E. Welsch and R. A. Becker. Robust non-linear regression using the dogleg algorithm. Technical report, National Bureau of Economic Research, 1975. (Cited on page 5.)

[70] J. Yang, A. Gupta, S. Upadhyay, L. He, R. Goel, and S. Paul. Tableformer: Robust transformer modeling for table-text encoding. *arXiv preprint arXiv:2203.00274*, 2022. (Cited on page 12.)

[71] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. (Cited on page 1.)

[72] S. Zhang and Y. 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, Nov. 2021. Association for Computational Linguistics. (Cited on page 12.)

[73] D. Zhou, Z. Yu, E. Xie, C. Xiao, A. Anandkumar, J. Feng, and J. M. Alvarez. Understanding the robustness in vision transformers. In *International Conference on Machine Learning*, pages 27378–27394. PMLR, 2022. (Cited on pages 1, 4, and 12.)

Supplementary Material of "Designing Robust Transformers using Robust Kernel Density Estimation"

A Proof of Proposition

Proposition 2. *Assume the robust loss function ρ is non-decreasing in $[0, \infty]$, $\rho(0) = 0$ and $\lim_{x\to 0} \frac{\rho(x)}{x} = 0$. Define $\psi(x) := \frac{\rho'(x)}{x}$ and assume $\psi(0) = \lim_{x\to 0} \frac{\rho'(x)}{x}$ exists and finite. Then the optimal $\hat{p}_{robust}$ can be written as*

$$\hat{p}_{robust} = \sum_{j\in[N]} \omega_j k_\sigma(\boldsymbol{x}_j, \cdot),$$

where $\omega = (\omega_1, \cdots, \omega_N) \in \Delta_N$, and $\omega_j \propto \psi\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - \hat{p}_{robust}\|_{\mathcal{H}_{k_\sigma}}\right)$. Here Δ_n denotes the n-dimensional simplex.

Proof. The proof of Proposition 2 is mainly adapted from the proof in [26]. Here, we provide proof of completeness. For any $p \in \mathcal{H}_{k_\sigma}$, we denote

$$J(p) = \frac{1}{N} \sum_{j\in[N]} \rho\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - p\|_{\mathcal{H}_{k_\sigma}}\right).$$

Then we have the following lemma regarding the Gateaux differential of J and a necessary condition for $\hat{p}_{\text{robust}}$ to be optimal solution of the robust loss objective function in equation (12).

Lemma 1. *Given the assumptions on the robust loss function ρ in Proposition 2, the Gateaux differential of J at $p \in \mathcal{H}_{k_\sigma}$ with incremental $h \in \mathcal{H}_{k_\sigma}$, defined as $\delta J(p; h)$, is*

$$\delta J(p; h) := \lim_{\tau\to 0} \frac{J(p + \tau h) - J(p)}{\tau} = -\langle V(p), h\rangle_{\mathcal{H}_{k_\sigma}},$$

where the function $V : \mathcal{H}_{k_\sigma} \to \mathcal{H}_{k_\sigma}$ is defined as:

$$V(p) = \frac{1}{N} \sum_{j\in[N]} \psi\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - p\|_{\mathcal{H}_{k_\sigma}}\right) (k_\sigma(\boldsymbol{x}_j, \cdot) - p).$$

A necessary condition for $\hat{p}_{robust}$ is $V(\hat{p}_{robust}) = 0$.

The proof of Lemma 1 can be found in Lemma 1 of [26]. Based on the necessary condition for $\hat{p}_{\text{robust}}$ in Lemma 1, i.e., $V(\hat{p}_{\text{robust}}) = 0$, we have

$$\frac{1}{N} \sum_{j\in[N]} \psi\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - \hat{p}_{\text{robust}}\|_{\mathcal{H}_{k_\sigma}}\right) (k_\sigma(\boldsymbol{x}_j, \cdot) - \hat{p}_{\text{robust}}) = 0.$$

Direct algebra indicates that $\hat{p}_{\text{robust}} = \sum_{j\in[N]} \omega_j k_\sigma(\boldsymbol{x}_j, \cdot)$ where $\omega = (\omega_1, \cdots, \omega_N) \in \Delta_N$, and $\omega_j \propto \psi\left(\|k_\sigma(\boldsymbol{x}_j, \cdot) - \hat{p}_{\text{robust}}\|_{\mathcal{H}_{k_\sigma}}\right)$. As a consequence, we obtain the conclusion of the proposition. □

B Ablation Studies

Table 3: Text PPL/NLL loss versus the parameter a of Huber loss function defined in equation 13 (upper) and Hampel loss function [26] (lower; we use $2 \times a$ and $3 \times a$ as parameters b and c) on original and word-swapped Wiki-103 dataset. The best results are highlighted in bold font and the second best are highlighted in underline. We choose $a = 0.4$ in rest of the experiments.

Robust Loss Parameter	0.1	0.2	0.4	0.6	0.8	1
Clean Data	34.92/3.57	34.87/3.56	**34.29/3.54**	34.38/3.54	34.46/3.54	34.48/3.54
Word Swap	56.82/4.01	55.97/3.99	**55.68/3.99**	57.89/4.03	58.26/4.04	58.37/4.04
Clean Data	34.67/3.55	**34.32/3.54**	34.35/3.54	34.47/3.54	34.53/3.54	34.58/3.54
Word Swap	58.02/4.03	**57.86/4.03**	57.92/4.03	58.24/4.04	58.37/4.04	58.43/4.04

Table 4: Top-1 classification accuracy on ImageNet versus the parameter a of Huber loss function defined in equation 13 under different settings. The best results are highlighted in bold font and the second best are highlighted in underline. We choose $a = 0.2$ in rest of the experiments.

Huber Loss Parameter	0.1	0.2	0.4	0.6	0.8	1
Clean Data	71.45	**72.83**	71.62	71.07	70.65	70.34
FGSM	**56.72**	55.83	55.34	54.87	54.02	52.98
PGD	**46.37**	44.15	43.87	43.25	42.69	41.96
SPSA	52.38	**52.42**	51.69	51.34	50.97	48.22
Imagenet-C	45.37	45.58	**45.63**	45.26	44.63	43.76

Table 5: Top-1 classification accuracy on ImageNet versus the parameter a of Hampel loss function defined in [26] under different settings. We use $2 \times a$ and $3 \times a$ as parameters b and c. The best results are highlighted in bold font and the second best are highlighted in underline. We choose $a = 0.2$ in rest of the experiments.

Hampel Loss Parameter	0.1	0.2	0.4	0.6	0.8	1
Clean Data	71.63	**72.94**	71.84	71.23	70.87	70.41
FGSM	**56.42**	55.92	55.83	55.66	54.97	53.68
PGD	**45.18**	44.23	43.89	43.62	43.01	42.34
SPSA	**52.96**	52.48	52.13	51.46	50.92	50.23
Imagenet-C	44.76	45.61	46.04	**46.13**	45.82	45.31

Table 6: Top-1 classification accuracy on ImageNet versus the parameter β of SPKDE defined in equation 16 under different settings. $\beta = \frac{1}{1-\varepsilon} > 1$, where ε is the percentage of anomalous samples. A larger β indicates a more robust model. The best results are highlighted in bold font and the second best are highlighted in underline. We choose $\beta = 1.4$ in rest of the experiments.

β	1.05	1.2	1.4	1.6	1.8	2
Clean Data	**74.25**	73.56	73.22	73.01	72.86	72.64
FGSM	53.69	55.08	**56.03**	55.37	54.21	53.86
PGD	42.31	43.68	**44.51**	44.32	44.17	43.71
SPSA	51.29	52.02	52.64	**52.84**	52.16	51.39
Imagenet-C	44.68	**45.49**	44.76	44.21	43.96	43.33

Table 7: Top-1 classification accuracy on ImageNet versus the number of iterations of the KIRWLS algorithm in equation 15 employed in Transformer-RKDE. Since the increased number of iterations does not lead to significant improvements of performance while the computational cost is much higher, we use the single-step iteration of the KIRWLS algorithm in Transformer-RKDE.

	Huber Loss				Hampel Loss			
Iteration #	1	2	3	5	1	2	3	5
Clean Data	72.83	72.91	72.95	72.98	72.94	72.99	73.01	73.02
FGSM	55.83	55.89	55.92	55.94	55.92	55.96	55.97	55.99
PGD	44.15	44.17	44.17	44.18	44.23	44.26	44.28	44.31
SPSA	52.42	52.44	52.45	52.45	52.48	52.53	52.55	52.56
Imagenet-C	45.58	45.61	45.62	45.62	45.61	45.66	45.68	45.71

Table 8: Computation time (measured by seconds per iteration) of baseline methods, Transformer-SPKDE, Transformer-MoM and Transformer-RKDE with different number of KIRWLS iterations. Transformer-SPKDE requires longer time since it directly obtains the optimal set of weights via the QP solver.

	Iterations of KIRWLS				DeiT	RVT	SPKDE	MoM-KDE
	1	2	3	5				
Time (s/it)	0.43	0.51	0.68	0.84	0.35	0.41	1.45	0.37