# Infinite attention: NNGP and NTK for deep attention networks

**Jiri Hron** [1] **Yasaman Bahri** [2] **Jascha Sohl-Dickstein** [2] **Roman Novak** [2]

## Abstract

There is a growing literature on the relationship between wide neural networks (NNs) and Gaussian processes (GPs), identifying an equivalence between the two for a variety of NN architectures. This equivalence enables, for instance, accurate approximation of the behaviour of wide Bayesian NNs without MCMC or variational approximations, or characterisation of the distribution of randomly initialised wide NNs optimised by gradient descent without ever running an optimiser. We provide a rigorous extension of these results to NNs with attention layers, showing that unlike single-head attention which induces non-Gaussian behaviour, multi-head attention architectures behave as GPs as the number of heads tends to infinity. We discuss the effects of positional encodings and layer normalisation, and propose modifications of the attention mechanism which improve performance of both finite and infinitely wide NNs. We evaluate attention kernels empirically, leading to a moderate improvement upon the previous state-of-the-art on CIFAR-10 for GPs without trainable kernels and advanced data preprocessing. Finally, we introduce new features to the Neural Tangents library (Novak et al., 2020) allowing applications of NNGP/NTK models, with and without attention, to variable-length sequences, with an example on the IMDb dataset.

## 1. Introduction

One of the currently most active research directions in theoretical deep learning is the study of NN behaviour as the number of parameters in each layer goes to infinity (e.g., Matthews et al., 2018; Lee et al., 2018; Garriga-Alonso et al., 2019; Novak et al., 2019; Li & Liang, 2018; Allen-Zhu et al., 2019; Du et al., 2019; Arora et al., 2019; Yang, 2019b). Building upon these efforts, we study the asymptotic behaviour of NNs with attention layers (Bahdanau et al., 2015; Vaswani et al., 2017) and derive the corresponding neural network Gaussian proccess (NNGP) and Neural Tangent kernels (NTK, Jacot et al., 2018; Lee et al., 2019).

Beyond their recent empirical successes (e.g., Radford et al., 2019; Devlin et al., 2019), attention layers are also interesting from the theoretical perspective as the standard proof techniques used to establish asymptotic Gaussianity of the input-to-output mappings represented by wide NNs (Matthews et al., 2018; Yang, 2019b) cannot be applied.

To understand why, consider the following simplified attention layer model: let $x \in \mathbb{R}^{d^s \times d'}$ be the input with $d^s$ *spatial* and $d'$ *embedding* dimensions (by spatial, we mean, e.g., the number of tokens in a string or pixels in an image), $W^Q, W^K, W^V \in \mathbb{R}^{d' \times d}$ be weight matrices, and define queries $Q(x) := xW^Q$, keys $K(x) := xW^K$, and values $V(x) := xW^V$ as usual. The attention layer output is then

$$f(x) := \zeta\left(\frac{Q(x)K(x)^\top}{\sqrt{d}}\right)V(x) = \zeta(G(x))V(x)\,, \quad (1)$$

where $\zeta$ is the row-wise softmax function.

Now observe that $\dim G(x) = d^s \times d^s$ where the spatial dimension $d^s$ stays finite even as the number of parameters—here proportional to $d$—goes to infinity. As we will show rigorously in Section 3, this fact combined with the $d^{-1/2}$ scaling causes each column of $f(x)$ to be a linear combination of the *same stochastic matrix* $\zeta(G(x))$, and thus statistically dependent even in the infinite width limit.

Since the exchangeability based arguments (Matthews et al., 2018; Garriga-Alonso et al., 2019) require that certain moment statistics of $f(x)$ asymptotically behave as if its columns were independent (see condition *b* in lemma 10, Matthews et al., 2018), they do not extend to attention layers in a straightforward manner. Similarly, the proofs based on Gaussian conditioning (Novak et al., 2019; Yang, 2019b) require that given the input $x$, the conditional covariance of each column of $f(x)$ converges (in probability) to the same *deterministic* positive semidefinite matrix (see propositions 5.5 and G.4 in Yang, 2019b) which will not be the case due to the aforementioned stochasticity of $\zeta(G(x))$.

Among the many interesting contributions in (Yang, 2019b), the author proposes to resolve the above issue by replacing

[1] University of Cambridge. Work done while interning at Google Brain. [2] Google Brain. Correspondence to: Jiri Hron <jh2084@cam.ac.uk>.

*Proceedings of the $37^{th}$ International Conference on Machine Learning*, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

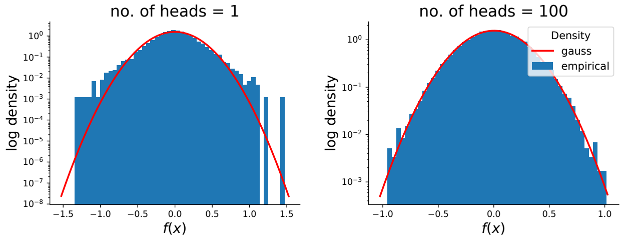

*Figure 1.* **Distribution of an attention layer output for single-head** (left) **and 100-head** (right) architecture at initialisation under the $d^{-1/2}$ scaling when $d$ is large (1000). Red line is the Gaussian density with sample mean and variance substituted for its parameters. Unlike multi-head, the empirical distribution of single-head attention significantly deviates from Gaussian despite $d \gg 0$.

the $d^{-1/2}$ scaling in Equation (1) by $d^{-1}$ which does enable application of the Gaussian conditioning type arguments. However, it also forces the attention layer to only perform computation similar to average pooling in the infinite width limit, and reduces the overall expressivity of attention even if suitable modifications preventing the pooling behaviour are considered (see Section 3.2).

We address the above issues by modifying the exchangability based technique and provide a rigorous characterisation of the infinite width behaviour under both the $d^{-1/2}$ and $d^{-1}$ scalings. We also show that positional encodings (Gehring et al., 2017; Vaswani et al., 2017) can improve empirical performance even in the infinite width limit, and propose modifications to the attention mechanism which results in further gains for both finite and infinite NNs. In experiments, we moderately improve upon the previous state-of-the-art result on CIFAR-10 for GP models without data augmentation and advanced preprocessing (cf. Yu et al., 2020). Finally, since attention is often applied to text datasets, we release code allowing applications of NNGP/NTK models to variable-length sequences, including an example on the IMDb reviews dataset.

## 2. Definitions and notation

**Neural networks:** $f^\ell(x)$ denotes the output of $\ell^{\text{th}}$ layer for an input $x \in \mathcal{X} \subset \mathbb{R}^{d^s \times d^0}$, and $g^\ell(x) := \phi(f^\ell(x))$ the corresponding post-nonlinearity where $\phi \colon \mathbb{R} \to \mathbb{R}$ is the activation function applied elementwise (for convenience, we set $g^0(x) = x$). We assume the network has $L \in \mathbb{N}$ hidden layers, making $f^{L+1}(x)$ the output, and that the input set $\mathcal{X}$ is *countable*. As we will be examining behaviour of sequences of increasingly wide NNs, the variables corresponding to the $n^{\text{th}}$ network are going to be denoted by a subscript $n$ (e.g., $f_n^\ell(x)$ is the output of $\ell^{\text{th}}$ layer of the $n^{\text{th}}$ network in the sequence evaluated at $x$). We also use

$$\begin{aligned} f_{n,\cdot j}^\ell &:= \{f_{n,ij}^\ell(x) \colon x \in \mathcal{X}, i \in [d^s]\} \\ f_n^\ell &:= \{f_{n,\cdot j}^\ell \colon j \in \mathbb{N}\}\,, \end{aligned}$$

with $[d^s] = \{1, 2, \ldots, d^s\}$. To reduce clutter, we omit the $\ell$ index where it is clear from the context or unimportant.

**Shapes:** $f_n^\ell(x), g_n^\ell(x) \in \mathbb{R}^{d^s \times d_n^\ell}$ with $d^s, d_n^\ell \in \mathbb{N}$ respectively the *spatial* and *embedding* dimensions. If there are multiple spatial dimensions, such as height and width for images, we assume these have been flattened into a single dimension. Finally, we will allow the row space dimension of $W_n^{\ell,Q}, W_n^{\ell,K} \in \mathbb{R}^{d_n^{\ell-1} \times d_n^{\ell,G}}$ to differ from that of $W_n^{\ell,V} \in \mathbb{R}^{d_n^{\ell-1} \times d_n^\ell}$, leading to the modified definition

$$G_n^\ell(x) = \frac{Q_n^\ell(x) K_n^\ell(x)^\top}{\sqrt{d_n^{\ell,G}}} \tag{2}$$

**Multi-head attention:** Equation (1) describes a vanilla version of a *single-head* attention layer. Later in this paper, we examine the *multi-head* attention alternative in which the output $f_n^\ell(x)$ is computed as

$$f_n^\ell(x) = \left[f_n^{\ell 1}(x), \ldots, f_n^{\ell d_n^{\ell,H}}(x)\right] W_n^{\ell,O}\,, \tag{3}$$

i.e., by stacking the outputs of $d_n^{\ell,H} \in \mathbb{N}$ independently parametrised heads into a $d^s \times d_n^{\ell,H} d_n^\ell$ matrix and projecting back into $d^s \times d_n^\ell$ by $W_n^{\ell,O} \in \mathbb{R}^{d_n^{\ell,H} d_n^\ell \times d_n^\ell}$. The embedding dimension of each head $d_n^{\ell,V}$ can optionally differ from $d_n^\ell$. To distinguish the weight matrices corresponding to the individual heads, we will be using a superscript $h$, e.g., $Q_n^{\ell h}(x) = g_n^{\ell-1}(x) W_n^{\ell h,Q}$. Multi-head architectures were popularised by (Vaswani et al., 2017) and are widely used in the literature (for example Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Lee et al., 2020).

**Weight distribution:** As usual, we will assume Gaussian initialisation of the weights, i.e., $W_{n,ij}^{\ell h,Q} \sim \mathcal{N}(0, \sigma_Q^2 / d_n^{\ell-1})$, $W_{n,ij}^{\ell h,K} \sim \mathcal{N}(0, \sigma_K^2 / d_n^{\ell-1})$, $W_{n,ij}^{\ell h,V} \sim \mathcal{N}(0, \sigma_V^2 / d_n^{\ell-1})$, and $W_{n,ij}^{\ell,O} \sim \mathcal{N}(0, \sigma_O^2 / (d_n^{\ell,H} d_n^\ell))$, all i.i.d. over the $i, j$ and $\ell, h$ indices for all $n$. The scaling of variance by inverse of the input dimension is standard and ensures that the asymptotic variances do not diverge (Neal, 1996; LeCun et al., 1998; He et al., 2015). Throughout Sections 3 and 4, we assume all the $\sigma^2$ parameters are equal to one, and only state the results in full generality in the appendix.

**NNGP/NTK:** As discussed in the introduction, randomly initialised NNs induce a distribution over the $f_n^{L+1}$ mappings. For a variety of architectures, this distribution converges (weakly) to that of a GP as $\min_{\ell \in [L]} d_n^\ell \to \infty$, both at initialisation (NNGP), and after continuous gradient descent optimisation of the randomly initialised NN with respect to a mean squared error loss (NTK). Both the NNGP and NTK distributions are typically zero mean, and we use $\kappa^{L+1}$ and $\Theta^{L+1}$ to denote their respective kernel functions.

These kernel functions tend to have a recursive structure where each layer in the underlying NN architecture is asso-

*Table 1.* **Overview of the discussed kernels.** The $d$ column refers to the $d^{-1}$ and $d^{-1/2}$ scaling of the $Q(x)K(x)^\top$. $(\tilde{\kappa}, \widetilde{\Theta})$ denote the input and $(\kappa, \Theta)$ the output NNGP and NTK kernels. NNGP and NTK columns are stated as updates for full $d^s \times d^s$ covariance blocks unless the generic spatial dimension subscripts $ab$ are used. To fit to page width, we use superscripts to denote dependence on inputs, e.g., replacing $\tilde{\kappa}\,(x, x)$ by $\tilde{\kappa}^{xx'}$. $\langle A, B\rangle_F = \sum_{ij} A_{ij} B_{ij}$ is the Frobenius product of matrices $A, B$, with $\|A\|_F^2 = \langle A, A\rangle$. $\mathcal{I}$ denotes interpolation, e.g., $\mathcal{I} \circ \tilde{\kappa}\,(x, x') = \alpha\tilde{\kappa}\,(x, x') + (1-\alpha)R$ with fixed hyperparameters $\alpha \in [0, 1]$ and $R$ (a generic covariance related to initialisation of positional encodings); the special case $R = I$ is denoted by $\mathcal{I}_I$. ‡ is for optional operators (e.g., $\mathcal{I}^{\ddagger} \circ \tilde{\kappa}^{xx'}$ can be replaced with $\tilde{\kappa}^{xx'}$). $W^Q = W^K$ initialisation assumed for all $d^{-1}$, and $\zeta$ = identity for all $d^{-1/2}$ kernels (see Sections 3.2 and 4.1 respectively). See Sections 3 and 4 for derivations, and (Yang, 2019b) for the LayerNorm kernel (stated here for ease of reference).

| Kernel | $d$ | NNGP | NTK |
|---|---|---|---|
| Vanilla | $1$ | $\zeta(\tilde{\kappa}^{xx})\tilde{\kappa}^{xx'}\zeta(\tilde{\kappa}^{x'x'})^\top$ | $2\kappa^{xx'} + \zeta(\tilde{\kappa}^{xx})\widetilde{\Theta}^{xx'}\zeta(\tilde{\kappa}^{x'x'})^\top$ |
| | $\frac{1}{2}$ | $\tilde{\kappa}^{xx'}\left\Vert\tilde{\kappa}^{xx'}\right\Vert_F^2$ | $4\kappa_{ab}^{xx'} + \langle\tilde{\kappa}^{xx'}, 2\tilde{\kappa}_{ab}^{xx'}\widetilde{\Theta}^{xx'} + \widetilde{\Theta}_{ab}^{xx'}\tilde{\kappa}^{xx'}\rangle_F$ |
| Random Positional Encoding | $1$ | $\zeta(\mathcal{I}_I \circ \tilde{\kappa}^{xx})[\mathcal{I}_I \circ \tilde{\kappa}^{xx'}]\zeta(\mathcal{I}_I \circ \tilde{\kappa}^{x'x'})^\top$ | $2\kappa^{xx'} + \zeta(\mathcal{I}_I \circ \tilde{\kappa}^{xx})[\mathcal{I}_I \circ \widetilde{\Theta}^{xx'}]\zeta(\mathcal{I}_I \circ \tilde{\kappa}^{x'x'})^\top$ |
| | $\frac{1}{2}$ | $\mathcal{I}_I \circ \tilde{\kappa}^{xx'}\left\Vert\mathcal{I}_I \circ \tilde{\kappa}^{xx'}\right\Vert_F^2$ | $4\kappa_{ab}^{xx'} + \langle\mathcal{I}_I \circ \tilde{\kappa}^{xx'}, 2[\mathcal{I}_I \circ \tilde{\kappa}_{ab}^{xx'}]\mathcal{I}_I \circ \widetilde{\Theta}^{xx'} + [\mathcal{I}_I \circ \widetilde{\Theta}_{ab}^{xx'}]\mathcal{I}_I \circ \tilde{\kappa}^{xx'}\rangle_F$ |
| Structured Positional Encoding | $1$ | $\zeta(\mathcal{I} \circ \tilde{\kappa}^{xx})[\mathcal{I}^{\ddagger} \circ \tilde{\kappa}^{xx'}]\zeta(\mathcal{I} \circ \tilde{\kappa}^{x'x'})^\top$ | $2\kappa^{xx'} + \zeta(\mathcal{I} \circ \tilde{\kappa}^{xx})[\mathcal{I}^{\ddagger} \circ \widetilde{\Theta}^{xx'}]\zeta(\mathcal{I} \circ \tilde{\kappa}^{x'x'})^\top$ |
| | $\frac{1}{2}$ | $\mathcal{I} \circ \tilde{\kappa}^{xx'}\langle\mathcal{I}^{\ddagger} \circ \tilde{\kappa}^{xx'}, \mathcal{I} \circ \tilde{\kappa}^{xx'}\rangle_F$ | $4\kappa_{ab}^{xx'} + \langle\mathcal{I}^{\ddagger} \circ \tilde{\kappa}^{xx'}, [\mathcal{I} \circ \tilde{\kappa}_{ab}^{xx'}]\mathcal{I} \circ \widetilde{\Theta}^{xx'} + [\mathcal{I} \circ \widetilde{\Theta}_{ab}^{xx'}]\mathcal{I} \circ \tilde{\kappa}^{xx'}\rangle_F$<br>$+ \langle\mathcal{I} \circ \tilde{\kappa}^{xx'}\ , [\mathcal{I} \circ \tilde{\kappa}_{ab}^{xx'}]\mathcal{I}^{\ddagger} \circ \widetilde{\Theta}^{xx'}\rangle_F$ |
| Residual | – | $\alpha\tilde{\kappa}^{xx'} + (1-\alpha)R\tilde{\kappa}^{xx'}R^\top$ | $2(1-\alpha)\kappa^{xx'} + \alpha\widetilde{\Theta}^{xx'} + (1-\alpha)R\widetilde{\Theta}^{xx'}R^\top$ |
| LayerNorm | – | $\tilde{\kappa}_{ab}^{xx'}[\tilde{\kappa}_{aa}^{xx}\tilde{\kappa}_{bb}^{xx'}]^{-1/2}$ | $\widetilde{\Theta}_{ab}^{xx'}[\widetilde{\Theta}_{aa}^{xx}\widetilde{\Theta}_{bb}^{x'x'}]^{-1/2}$ |

ciated with a mapping $(\kappa^{\ell-1}, \Theta^{\ell-1}) \mapsto (\kappa^{\ell}, \Theta^{\ell})$ transforming the NNGP and NTK kernels according to the layer's effect on the outputs in the infinite width limit. Since nonlinearities are typically not treated as separate layers, we use $\tilde{\kappa}^{\ell}$ and $\widetilde{\Theta}^{\ell}$ to denote the intermediate transformation $(\kappa^{\ell-1}, \Theta^{\ell-1}) \mapsto (\tilde{\kappa}^{\ell}, \widetilde{\Theta}^{\ell})$ they induce. We generally assume every layer is followed by a nonlinearity, setting $(\tilde{\kappa}^{\ell}, \widetilde{\Theta}^{\ell}) = (\kappa^{\ell-1}, \Theta^{\ell-1})$ if none is used. In the next two sections, we uncover the mappings $(\tilde{\kappa}^{\ell}, \widetilde{\Theta}^{\ell}) \mapsto (\kappa^{\ell}, \Theta^{\ell})$ induced by various attention architectures.

## 3. Attention and Gaussian process behaviour

Throughout the rest of this paper, we restrict our focus to increasingly wide NNs including at least one attention layer. In particular, we consider sequences of NNs such that

$$\lim_{n\to\infty} \min_{\ell\in[L]} d_n^{\ell} = \infty\,, \tag{4}$$

and the reader should thus interpret any statements involving $n \to \infty$ as implicitly assuming Equation (4) holds.

Due to the space constraints, most of the technical discussion *including derivation of the NTK limit* is relegated to Appendix B. In this section, we only focus on the key step in our proof which relies on an inductive argument adapted from (Matthews et al., 2018). On a high level, the induction is applied from $\ell = 1$ to $\ell = L + 1$, and establishes that whenever $f_n^{\ell-1}$ converges in distribution to $\mathcal{GP}(0, \kappa^{\ell-1})$ at initialisation, $f_n^{\ell}$ also converges in distribution to $\mathcal{GP}(0, \kappa^{\ell})$ as $n \to \infty$. Since this fact is known for dense, convolutional, and average pooling layers, and almost all nonlinearities (Matthews et al., 2018; Lee et al., 2018; Garriga-Alonso et al., 2019; Novak et al., 2019; Yang, 2019b), it will be sufficient to show the same for attention layers.

### 3.1. Infinite width limit under the $d^{-1}$ scaling

As illustrated in Figure 1, use of the $d^{-1/2}$ scaling within a *single-head* architecture leads to a scale mixture behaviour of the attention layer outputs as the number of parameters goes to infinity. To obtain a Gaussian limit, Yang (2019b, appendix A) proposes to replace the definition in Equation (2) by $G_n(x) = (d_n^G)^{-1}Q_n(x)K_n(x)^\top$, i.e., the use of $d^{-1}$ scaling. The desired result then follows:

**Theorem 1** ($d^{-1}$ limit (Yang, 2019b))**.** *Under the $d^{-1}$ scaling and the assumptions stated in (Yang, 2019b):*

**(I)** *For any $(x, x') \in \mathcal{X} \times \mathcal{X}$ and $a, b, i, j \in [d^s]$, there exist constants $(\bar{\zeta}_{ai}^{x}, \bar{\zeta}_{bj}^{x'}) \in \mathbb{R} \times \mathbb{R}$ such that*

$$(\zeta(G_n(x))_{ai}, \zeta(G_n(x'))_{bj}) \xrightarrow{\mathbb{P}} (\bar{\zeta}_{ai}^{x}, \bar{\zeta}_{bj}^{x'})\,. \tag{5}$$

**(II)** *$f_n$ converges in distribution to $f \sim \mathcal{GP}(0, \kappa)$ with $f_{.k}$ and $f_{.l}$ independent for any $k \neq l$, and*

$$\kappa_{ab}\,(x, x') = \mathbb{E}[f_{a1}(x)f_{b1}(x')] = \sum_{i,j=1}^{d^s} \tilde{\kappa}_{ij}\,(x, x')\bar{\zeta}_{ai}^{x}\bar{\zeta}_{bj}^{x'}\,.$$

An analogous result also holds for multi-head attention architectures which follows by the usual argument for fully connected layers as long as either the number of embedding dimensions per head or the number of heads goes to infinity.

### 3.2. Limitations of the $d^{-1}$ scaling

While Theorem 1 is a good starting point, several issues have to be resolved before using the attention kernel in practice. Firstly, since $W_n^Q$ and $W_n^K$ are initialised independently, the $d^{-1}$ scaled inner products of keys and queries converges to zero, and thus for any $a, i$ and $x$, $\bar{\zeta}_{ai}^x = (d^s)^{-1}$ by the continuous mapping theorem. This issue was noted by Yang in appendix A but not discussed further as the main focus of the paper lies elsewhere. In any case, substituting $(d^s)^{-1}$ for all the $\bar{\zeta}$ coefficients will make $\kappa_{ab}(x, x') = \kappa_{ij}(x, x')$ for any $a, b, i, j \in [d^s]$, and in fact all of these entries will be equivalent to output of a simple global average pooling kernel (Novak et al., 2019, equation 17).[1]

Perhaps the simplest way to address the above issue is by drawing the initial weights such that $W_n^Q = W_n^K$. This will ensure that the key and query for a particular spatial dimension will point in the same direction and thus the attention weight corresponding to itself will be large with high probability. The resulting formula for $\kappa_{ab}(x, x')$ is

$$\sum_{i,j=1}^{d^s} \tilde{\kappa}_{ij}(x, x') \zeta(\tilde{\kappa}_{ai}(x, x)) \zeta(\tilde{\kappa}_{bj}(x', x')) \, . \tag{6}$$

Since Equation (6) resolves the issue of reduction to average pooling, a natural question is whether swapping $d^{-1/2}$ for $d^{-1}$ has any undesirable consequences in the infinite width limit. As we will see, this question can be answered in affirmative. In particular, we start by a proposition inspired by (Cordonnier et al., 2020) in which the authors show that an attention layer with a sufficient number of heads is at least as expressive as a standard convolutional layer, and that attention layers often empirically learn to perform computation akin to convolution. In contrast, Proposition 2 proves that there is no initial distribution of $W_n^Q$ and $W_n^K$ which would recover the convolutional kernel (Novak et al., 2019; Garriga-Alonso et al., 2019) in the infinite width limit.

**Proposition 2.** *There is no set of attention coefficients $\{\bar{\zeta}_{ai}^x \in \mathbb{R} \colon a, i \in [d^s], x \in \mathcal{X}\}$ such that for* all *positive semidefinite kernels $\tilde{\kappa}$ simultaneously*

$$\sum_{i,j=1}^{d^s} \tilde{\kappa}_{ij}(x, x') \bar{\zeta}_{ai}^x \bar{\zeta}_{bj}^{x'} = \sum_{i=1}^{d_f} \tilde{\kappa}_{N_a(i) N_b(i)}(x, x') \frac{1}{d_f} \, ,$$

*where $d_f$ is the dimension of the (flattened) convolutional filter, $N_a, N_b \subset [d^s]$ are the ordered subsets of pixels which are used to compute the new values of pixels $a$ and $b$, respectively, and $N_a(i), N_b(i)$ are the $i^{th}$ pixels in $N_a, N_b$.*

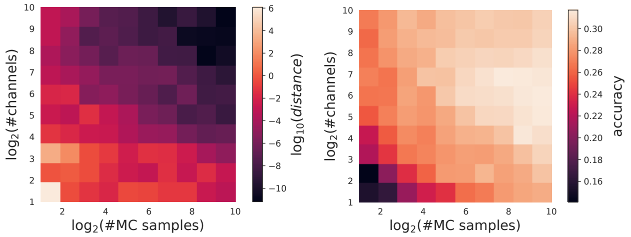

*Figure 2.* **Convergence** (left) **and validation accuracy** (right) **plots for an empirical NNGP kernel** estimated by Monte Carlo on a 2K/4K train/validation subset of 8x8-downsampled CIFAR-10, as the number weight samples averaged over (x-axis) and the number of parameters (y-axis) grows. Architecture: Convolution + ReLU, 2x Attention + ReLU, Flatten, Dense. For attention layers, $d_n^{\ell,G}$ = #channels but $d_n^{\ell,H} = d_n^{\ell,V} = \lfloor \sqrt{\text{\#channels}} \rfloor$ to reduce the memory footprint. Details in Appendix A.1.2.

In the next section, we will see that the convolutional kernel can be recovered under the $d^{-1/2}$ scaling (Proposition 4). However, we need to establish convergence scaling first.

### 3.3. Infinite width limit under the $d^{-1/2}$ scaling

As discussed in Section 1, single-head attention architectures can exhibit non-Gaussian asymptotic behaviour under the $d^{-1/2}$ scaling. This is inconvenient for our purposes as many modern NN architectures combine attention with fully connected, convolutional, and other layer types, all of which have Gaussian NNGP and NTK limits (e.g., Novak et al., 2019; Garriga-Alonso et al., 2019; Yang, 2019b). This Gaussianity simplifies derivation of the infinite width behaviour of many architectures and allows for easy integration with existing software libraries (Novak et al., 2020). Fortunately, the output of an attention layer becomes asymptotically Gaussian when the number of heads becomes large.

**Theorem 3** ($d^{-1/2}$ limit)**.** *Let $\ell \in \{2, \ldots, L+1\}$, and $\phi$ be such that $|\phi(x)| \leq c + m|x|$ for some $c, m \in \mathbb{R}_+$. Assume $f_n^{\ell-1}$ converges in distribution to $f^{\ell-1} \sim \mathcal{GP}(0, \kappa^{\ell-1})$, such that $f_{\cdot j}^{\ell-1}$ and $f_{\cdot k}^{\ell-1}$ are independent for any $j \neq k$, the variables $\{f_{n,\cdot j}^{\ell-1} \colon j \in \mathbb{N}\}$ are exchangeable over $j$.*

*Then as $\min\{n, d_n^{\ell,H}, d_n^{\ell,G}\} \to \infty$:*

**(I)** *$G_n^\ell = \{G_n^{\ell h}(x) \colon x \in \mathcal{X}, h \in \mathbb{N}\}$ converges in distribution to $G^\ell \sim \mathcal{GP}(0, \kappa^{\ell,G})$ with*

$$\mathbb{E}[G_{ai}^{\ell h}(x) G_{bj}^{\ell h'}(x')] = \delta_{h=h'} \tilde{\kappa}_{ab}^\ell(x, x') \, \tilde{\kappa}_{ij}^\ell(x, x') \, .$$

**(II)** *$f_n^\ell$ converges in distribution to $f^\ell \sim \mathcal{GP}(0, \kappa^\ell)$ with $f_{\cdot k}^\ell$ and $f_{\cdot l}^\ell$ independent for any $k \neq l$, and*

$$\begin{aligned} \kappa_{ab}^\ell(x, x') &= \mathbb{E}[f_{a1}^\ell(x) f_{b1}^\ell(x')] \qquad (7) \\ &= \sum_{i,j=1}^{d^s} \tilde{\kappa}_{ij}^\ell(x, x') \, \mathbb{E}[\zeta(G^{\ell 1}(x))_{ai} \zeta(G^{\ell 1}(x'))_{bj}] \, . \end{aligned}$$

[1] In fact, the asymptotic distribution induced by such an attention layer followed by flatten and dense layers is the same as that induced by global average pooling followed by a dense layer.

We can now revisit our argument from the previous section, and prove that unlike in Proposition 2, $d^{-1/2}$ scaling ensures a convolutional kernel can in principle be recovered.

**Proposition 4.** *Under the $d^{-1/2}$ scaling, there exists a distribution over $G$ such that for any $x, x'$ and $a, b, i, j$*

$$\mathbb{E}[\zeta(G(x))_{ai}\zeta(G(x'))_{bj}] = \begin{cases} \frac{1}{d_f}\,, & \exists k \in [d] \text{ s.t. } i = N_a(k)\,, j = N_b(k)\,, \\ 0\,, & \text{otherwise.} \end{cases} \tag{8}$$

## 4. Beyond the vanilla attention definition

Before progressing to empirical evaluation of infinitely wide attention architectures, two practical considerations have to be addressed: (i) the $d^{-1/2}$ scaling induced kernel in Equation (7) involves an analytically intractable integral $\mathbb{E}[\zeta(G^{\ell 1}(x))\zeta(G^{\ell 1}(x'))]$; (ii) incorporation of positional encodings (Gehring et al., 2017; Vaswani et al., 2017).

### 4.1. Alternatives to softmax in attention networks

We propose to resolve the analytical intractability of the $\mathbb{E}[\zeta(G^{\ell 1}(x))\zeta(G^{\ell 1}(x'))]$ in Equation (7) by substituting functions other than softmax for $\zeta$. In particular, we consider two alternatives: (i) $\zeta(x) = \text{ReLU}(x)$, and (ii) $\zeta(x) = x$, both applied elementwise. Besides analytical tractability of the expectation, our motivation for choosing (i) and (ii) is that ReLU removes the normalisation while still enforcing positivity of the attention weights, while the identity function allows the attention layer to learn an arbitrary linear combination of the values without constraints.

To see if either is a sensible modification, we evaluated performance of *finite* attention networks on CIFAR-10 for different choices of $\zeta$. Since softmax typically dampens the marginal variance of attention layer outputs (variance of a convex combination of random variables is upper bounded by the maximum of the individual variances), and both ReLU and identity can also significantly affect scale of the outputs, we optionally add layer normalisation as is common in attention architectures. We consider no normalisation (`none`),[2] normalisation applied after each head prior to multiplication by $W_n^{\ell,O}$ (`per_head`), and normalisation applied to the output after $W_n^{\ell,O}$ (`at_output`).

Figure 3 shows the results across varying hyperparameters and random seeds, and Table 8 (Appendix A.1.3) reports accuracies attained under optimal hyperparameter settings. As you can see, both the replacement of softmax and addition of layer normalisation significantly increases the performance of the NN, with $\zeta(x) = x$ and `at_output` normalisation being the best across variety of hyperparameter choices.

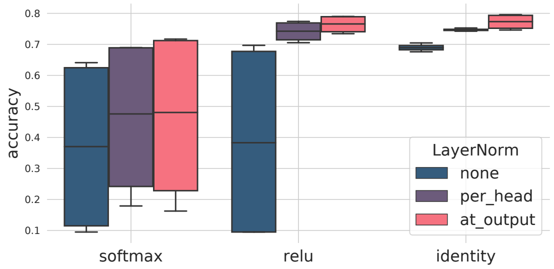

*Figure 3.* **Comparison of $\zeta$ alternatives.** Architecture: 4x Convolution + ReLU, Attention, Flatten, Dense. The captured variability is due to multiple random seeds, varying learning rate and network width, illustrating robustness of the reported results. Softmax significantly underperforms other $\zeta$ alternatives whenever attention is followed by layer normalisation. Details in Appendix A.1.3.

In light of the above, we will restrict our attention to the identity function alternative for $\zeta$ in the rest of the paper, and contrast its performance with the standard softmax choice where possible (finite NNs, and infinite attention NNs under the $d^{-1}$ scaling—see Theorem 1). Similarly, we will also leverage the `at_output` layer normalisation over the embedding dimension in our experiments. As shown by Yang (2019b, appendix A), layer normalisation does not prevent Gaussianity of the infinite width limit (see Table 1 for the associated NNGP and NTK kernel transformations).

### 4.2. Positional encodings

While substituting the identity function for $\zeta$ as suggested in Section 4.1 would technically allow us to move on to the experimental evaluation already, we found that positional encodings are as important in the infinite width limit as they are for the finite attention layers (Vaswani et al., 2017). Since there are many possible variants of the positional encoding implementation, we focus only on the major points here and provide more detail in Appendix C.

In finite networks, some of the most common ways to implement positional encodings is to modify the attention layer input by either `add`-ing $g_n^{\ell-1}(x) + E_n^{\ell}$ or `append`-ing $[g_n^{\ell-1}(x)\,, E_n^{\ell}]$ a matrix $E_n^{\ell}$ which may be either fixed or a trainable parameter. The purpose of $E_n^{\ell}$ is to provide the attention layer with information about the relationships between individual spatial dimensions (e.g., position of a particular pixel in an image, or of a token in a string).

#### 4.2.1. Effect on the infinite width limit

If we assume $E_n^{\ell}$ is trainable and each of its columns is initialised independently from $\mathcal{N}(0, R)$, $R$ positive semi-

[2] Despite the similarity between attention with ReLU or identity for $\zeta$ and dense layers with cubic nonlinearities, which are known to be hard to train, we found that layer normalisation was not strictly necessary. We believe this is partly because we only used a single attention layer, and partly because the weights for keys, queries, and values are initialised independently which leads to relatively better behaved distribution of gradients at initialisation.

definite, it can be shown that both in the `add` and `append` case, the attention layer output converges (in distribution) to a Gaussian infinite width limit (see Appendix C). The corresponding kernels can be stated in terms of an operator $\mathcal{I}$ which interpolates any given kernel $\kappa$ with $R$

$$\mathcal{I}\colon \kappa\,(x,x') \mapsto \alpha\kappa\,(x,x') + (1-\alpha)R\,, \tag{9}$$

where $\alpha \in [0,1]$ is a hyperparameter,[3] yielding the following modification of the kernel induced by the $d^{-1}$ scaling and $W^Q = W^K$ initialisation (Equation (6)):

$$\kappa_{ab}\,(x,x') = \bar{\zeta}_a^x[\mathcal{I}\circ\tilde{\kappa}\,(x,x')](\bar{\zeta}_b^{x'})^\top\,, \tag{10}$$

where $\bar{\zeta}_a^x := \zeta(\mathcal{I}\circ\tilde{\kappa}\,(x,x))_{a\cdot}$ and similarly for $\bar{\zeta}_b^{x'}$. The modification of the kernel induced by the $d^{-1/2}$ scaling, $W^Q, W^K$ initialised independently, and $\zeta$ replaced by the identity function (Equation (7)), then leads to:

$$\kappa_{ab}\,(x,x') = \mathcal{I}\circ\tilde{\kappa}_{ab}\,(x,x')\sum_{i,j=1}^{d^s}[\mathcal{I}\circ\tilde{\kappa}_{ij}\,(x,x')]^2. \tag{11}$$

Several comments are in order. Firstly, the typical choice of the initialisation covariance for $E_n^\ell$ is $R = \rho I$, $\rho > 0$. This may be reasonable for the $\bar{\zeta}_a^x = \zeta(\mathcal{I}\circ\tilde{\kappa}\,(x,x))_{a\cdot}$ in Equation (10) when $\zeta$ is the softmax function as it increases attention to the matching input spatial dimension, but does not seem to have any "attention-like" interpretation in Equation (11) where the effect of applying $\mathcal{I}$ to $\tilde{\kappa}$ with $R = \rho I$ is essentially analogous to that of just adding i.i.d. Gaussian noise to each of the attention layer inputs.

Secondly, the right hand side of Equation (11) is just a scaled version of the discussed $\mathcal{I}\circ\tilde{\kappa}$ kernel, with the scaling constant disappearing when the attention layer is followed by layer normalization (Table 1). Both of these call into question whether the performance of the corresponding finite NN architectures will translate to its infinite width equivalent. We address some of these issues next.

### 4.2.2. Structured positional encodings

As mentioned, the main purpose of positional encodings is to inject structural information present in the inputs which would be otherwise ignored by the attention layer. A natural way to resolve the issues discussed in previous section is thus to try to incorporate similar information directly into the $R$ covariance matrix. In particular, we propose

$$R_{ab} = \rho\begin{cases}\exp\{-\varphi[r_h(a,b)^2 + r_v(a,b)^2]\} & \text{(image)}\\ \exp\{-\varphi\, r_s(a,b)^2\} & \text{(string)}\end{cases} \tag{12}$$

where $\rho, \varphi > 0$ are hyperparameters, $r_h(a,b)$ and $r_v(a,b)$ are the absolute horizontal and vertical distances between the pixels $a$ and $b$ divided by the image width and height respectively, and $r_s(a,b)$ is the absolute distance between the relative position of tokens $a$ and $b$, e.g., if $a$ is the 4th token out of 7 in the first, and $b$ is the 2nd token out of 9 in the second string, then $r_s(a,b) = |\frac{4}{7} - \frac{2}{9}|$.

To motivate the above definition, let us briefly revisit Equation (10). Intuitively, the $d^{-1}$ kernel $\bar{\zeta}_a^x[\mathcal{I}\circ\tilde{\kappa}\,(x,x')](\bar{\zeta}_b^{x'})^\top$ is a result of multiplying the asymptotically Gaussian values $V \sim \mathcal{GP}(0, \mathcal{I}\circ\tilde{\kappa})$ by matrices of row-wise stacked $\bar{\zeta}^x = [\bar{\zeta}_1^x; \ldots; \bar{\zeta}_{d^s}^x]$ vectors, e.g., $f(x) = \bar{\zeta}^x V(x)$,[4] meaning that the $\bar{\zeta}$ vectors serve the role of attention weights in the infinite width limit. This in turn implies that the greater the similarity under $\tilde{\kappa}_{ab}\,(x,x)$ the higher the attention paid by $a$ to $b$. Thus, if we want to inject information about the relevance of neighbouring pixels in an image or tokens in a string, we need to increase the corresponding entries of $\mathcal{I}\circ\tilde{\kappa}\,(x,x) = \alpha\tilde{\kappa}\,(x,x') + (1-\alpha)R$ which can be achieved exactly by substituting the $R$ from Equation (12).

The above reasoning only provides the motivation for modifying the attention weights using positional encodings but not necessarily for modifying the asymptotic distribution of the values $V$. Adding positional encodings only inside the $\zeta$ is not uncommon (e.g., Shaw et al., 2018), and thus we will also experiment with kernels induced by adding positional encodings only to the inputs of $Q_n$ and $K_n$, leading to

$$\kappa_{ab}\,(x,x') = \bar{\zeta}_a^x\tilde{\kappa}\,(x,x')(\bar{\zeta}_b^{x'})^\top\,, \tag{13}$$

under the $d^{-1}$ scaling (cf. Equation (10)), and

$$\kappa_{ab}\,(x,x') = \mathcal{I}\circ\tilde{\kappa}_{ab}\,(x,x')\sum_{i,j=1}^{d^s}\tilde{\kappa}_{ij}\,(x,x')\mathcal{I}\circ\tilde{\kappa}_{ij}\,(x,x')\,,$$

under the $d^{-1/2}$ scaling (cf. Equation (11)).

Finally, note that the last kernel remains a scaled version of the aforementioned $\mathcal{I}\circ\tilde{\kappa}$ kernel, albeit now with $R$ as in Equation (12). In our experience, using just $\mathcal{I}\circ\tilde{\kappa}$ without the scaling leads to improved empirical performance, and further gains can be obtained with the related kernel

$$\kappa_{ab}\,(x,x') = \alpha\tilde{\kappa}_{ab}\,(x,x') + (1-\alpha)R_{a\cdot}\tilde{\kappa}\,(x,x')R_{b\cdot}^\top\,. \tag{14}$$

We call Equation (14) the *residual* attention kernel, as it can be obtained as a limit of architecture with a skip connection, $f_n^\ell(x) = \sqrt{\alpha}g_n^{\ell-1}(x) + \sqrt{1-\alpha}\tilde{f}_n^\ell(x)$, where $\tilde{f}_n^\ell(x)$ is output of an attention layer (details in Appendix D).

[3] If $E_n^\ell$ is `append`-ed, $\alpha = \lim_{n\to\infty} d_n^{\ell-1}/(d_n^{\ell,E} + d_n^{\ell-1})$ with $d_n^{\ell,E}$ the row space dimension of $E_n^\ell$. When $E_n^\ell$ is `add`-ed, we replace $g_n^{\ell-1}(x)$ by $\sqrt{\alpha}g_n^{\ell-1}(x) + \sqrt{1-\alpha}E_n^\ell$ so as to prevent increase of the layer's input variance (see Appendix C).

[4] By standard Gaussian identities, if $Z \sim \mathcal{N}(0,\Sigma)$, and $A$ is a deterministic matrix, then $AZ \sim \mathcal{N}(0, A\Sigma A^\top)$.

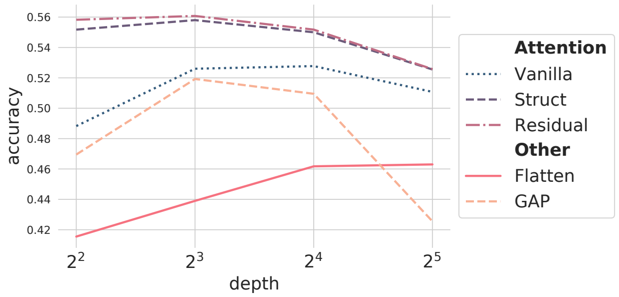

*Figure 4.* **Validation accuracy as a function of depth for various NNGP kernels** on a 2K/4K train/validation split of CIFAR-10 (no pixel downsampling). Architecture: `[depth]` x Convolution + ReLU, followed by a single instance of the kernel specified in the legend (attention kernels combined with additional Flatten), and Dense. See Table 1 for attention, and (Novak et al., 2019; Garriga-Alonso et al., 2019) for Convolutional, Flatten, and Global Average Pooling (GAP) kernel descriptions. Results reported for best hyperparameters ($d^{-1}$ scaling generally resulted in better performance for the `Struct` kernel). More experimental details in Appendix A.1.4. Notice the improved performance of attention kernels with positional embeddings and layer normalisation (i.e., `Struct`, `Residual`) over their `Vanilla` counterpart.

## 5. Experiments

We evaluate the attention NNGP/NTK kernels on the CIFAR-10 (Krizhevsky, 2009) and IMDb reviews (Maas et al., 2011) datasets. While IMDb is a more typical setting for attention models (Section 5.2), we included CIFAR-10 experiments (Section 5.1) due to desire to compare with other NNGPs/NTKs on an established benchmark (e.g., Novak et al., 2019; Du et al., 2019; Yu et al., 2020), and the recent successes of attention on vision tasks (e.g., Wang et al., 2017; 2018; Hu et al., 2018; Woo et al., 2018; Chen et al., 2018; Ramachandran et al., 2019; Bello et al., 2019). Our experimental code utilises the JAX (Bradbury et al., 2018) and Neural Tangents (Novak et al., 2020) libraries.

### 5.1. CIFAR-10

We have run two types of experiments on CIFAR-10: (i) smaller scale experiments focused on understanding how different hyperparameters of the attention kernel affect empirical performance; (ii) a larger scale experiment comparing attention kernels to existing NNGP/NTK benchmarks. The smaller scale experiments were run on a randomly selected subset of six thousand observations from the *training* set, with the 2K/4K train/validation split. This subset was used in Figures 2 and 4, and for hyperparameter tuning. Selected hyperparameters were then employed in the larger scale experiment with the usual 50K/10K train/test split.

*Table 2.* **CIFAR-10 test accuracies** of attention kernels and existing NNGP/NTK alternatives. The standard 50K/10K train/test split is used (no pixel downsampling). Best hyperparemeters from the 2K/4K subset experiments used for each kernel, $d^{-1}$ scaling for the `Struct` kernel (see Table 1). Details in Appendix A.1.5.

| KERNEL | NNGP | NTK |
|---|---|---|
| FLATTEN | 65.54 | 66.27 |
| GAP (YU ET AL., 2020) | 77.85 | 77.39 |
| LAP (YU ET AL., 2020) | 80.36 | 79.71 |
| STRUCT | 80.55 | 79.93 |
| RESIDUAL | **80.72** | **80.10** |

All kernels evaluated in this section correspond to NN architectures composed of multiple stacked convolutional layers with ReLU activations, followed by either simple flattening, global average pooling (GAP), or one of our attention kernels itself followed by flattening and, except for the `Vanilla` attention case (see Table 1), also by layer normalisation; the output is then computed by a single dense layer placed on top. The choice to use only one attention layer was made to facilitate comparison with (Novak et al., 2019; Du et al., 2019; Yu et al., 2020) where the same set-up with a stack of convolutional layers was considered. Adding more attention layers did not result in significant gains during hyperparameter search though. Exact details regarding data normalisation, hyperparameter tuning, and other experimental settings can be found in Appendix A.

The most important observations from the smaller scale experiments are captured in Figure 4 which shows the validation accuracy of various NNGP models as a function of kernel choice and number of convolutional layers (`depth`) preceding the final flatten/GAP/attention plus dense block. Firstly, notice that except for the `Flatten` model, all other kernel choices achieve their best performance at smaller depths which is consistent with existing literature (Arora et al., 2019; Yu et al., 2020).

Secondly, observe that both the `Struct` and `Residual` attention kernels significantly outperform the `Vanilla` one, demonstrating that the use of positional embeddings and layer normalisation is helpful even in the infinite width limit as claimed in Section 4.2. In contrast, we did not find significant evidence for $\zeta(x) = x$ outperforming the standard softmax choice as was the case for finite networks (see Figure 3), with the best set of hyperparameters for `Struct` $d^{-1}$ with softmax being only marginally better than the best results with the identity function (recall that no $d^{-1/2}$ kernels use $\zeta$ = softmax due to the intractability discussed in Section 4). This finding provides hope that the $d^{-1/2}$ kernels also do not sacrifice much in terms of performance by using identity for $\zeta$, but also points to salient differences between the qualitative effects of individual hyperparameter choices in finite and infinite attention layers.

Using the insights from the smaller scale experiments, we ran the larger scale experiment on the full dataset using

*Table 3.* **IMDb sentiment classification, test accuracies** of simple NNGP/NTK models on the 25K/25K train/test split using GloVe word embeddings (Pennington et al. (2014); 840B.300d). `GAP-only` corresponds to a single global average pooling layer followed by a linear fully connected readout. `GAP-FCN` has 2 ReLU fully connected layers after GAP. `Struct` has an attention layer preceding GAP, followed by one (NNGP) or two (NTK) fully connected layers. Models selected on a validation set of 10K reviews. Details in Appendix A.2.2.

| Kernel | NNGP | NTK |
|---|---|---|
| GAP-only | – 84.98 | – |
| GAP-FCN | 85.82 | 85.80 |
| Struct | **86.09** | **86.09** |

eight layer models and the `Struct` and `Residual` attention kernels. We used the positional embedding covariance matrix defined in Equation (12) in both cases, and $d^{-1}$ with softmax for the `Struct` kernel (further details in Appendix A.1.5). The results can be found in Table 2. As you can see, attention performs significantly better than the `GAP` kernel (Arora et al., 2019), and also provides a moderate improvement over the recent local average pooling (`LAP`) results (Yu et al., 2020). Since we used the validation accuracy from smaller scale experiments to determine our hyperparameters, we are comparing against the best cross-validation results from (Yu et al., 2020) for fairness.

### 5.2. IMDb reviews

Although there has been interest in applying attention in vision, to date it has been predominantly recognized for performance on language tasks. However, most of available NNGP/NTK kernel implementations (Matthews et al., 2018; Lee et al., 2018; Garriga-Alonso et al., 2019; Arora et al., 2019; Yang, 2019b; Yu et al., 2020) are hard-coded for the specific experiments performed in the respective paper. Neural Tangents (Novak et al., 2020) allows for some flexibility, yet still accepts only inputs of fixed length and having exactly zero (i.e. inputs to fully connected networks) or two (images for CNNs) spatial dimensions.

We release code allowing use of NNGP/NTK models (with or without attention) on inputs of variable spatial extent and arbitrary dimensionality (e.g., one spatial dimension for texts and time series, three spatial dimensions for videos). Our implementation seamlessly extends the Neural Tangents library, enabling research and application of NNGP and NTK models to new domains with almost no extra effort.

As an example, we present the first benchmarks of simple NNGP and NTK models on the IMDb sentiment classification dataset in Table 3. We observe that `Struct` kernels outperform the `GAP-only` kernel (corresponding to linear regression on the word embeddings mean), but provides marginal benefit compared to a fully connected model on top of the pooling layer (`GAP-FCN`). We conjecture this is due to high-quality word embeddings partially incorporating the inductive bias of the considered model. Indeed, we further demonstrate this effect by contrasting the gaps in performance between different kernel families on high- and low-quality word embeddings in Table 4.

*Table 4.* **IMDb sentiment classification, test accuracies** on a 3.2K/1.6K train/test split. When high-quality word embeddings are used (300-dimensional GloVe trained on 840B tokens), complex models yield diminishing returns. Contrarily, simple embeddings (50-dimensional GloVe trained on 6B tokens) lead to significant gaps in model performance due to respective inductive biases (`GAP-only` $<$ `GAP-FCN` $<<$ `CNN-GAP` $\approx$ `Struct`). Models selected on a validation set of 1.6K reviews. Details in Appendix A.2.3.

| Embeddings:<br>(dimension) | | GloVe 840B<br>(300) | GloVe 6B<br>(50) |
|---|---|---|---|
| GAP only | | 83.81 | 73.00 |
| NNGP | GAP-FCN | 83.75 | 74.44 |
| | CNN-GAP | 84.69 | 81.00 |
| | Struct | 83.56 | 80.88 |
| NTK | GAP-FCN | 83.81 | 74.88 |
| | CNN-GAP | **84.88** | 80.31 |
| | Struct | 84.00 | **81.06** |

Naturally, our sample IMDb results are not competitive with the state-of-the-art, which achieve up to 97.4% (Thongtan & Phienthrakul, 2019, Table 4). However, we hope they will be a useful baseline for future research in infinite width sequence models, and that our codebase will substantially facilitate the process by enabling variable-length, arbitrary-dimensional input processing.

## 6. Conclusion

Unlike under the $d^{-1}$ scaling of $Q(x)K(x)^\top$ proposed in (Yang, 2019b), the standard $d^{-1/2}$ scaling may lead to non-Gaussian asymptotic behaviour of attention layer outputs. Gaussianity of the limit can however be obtained by taking the number of heads to infinity. We explored the effect of positional encodings and replacements for the softmax function in attention layers, leading to improved performance for both finite and infinite attention architectures. On CIFAR-10, attention improves moderately upon the previous state-of-the-art for GPs without trainable kernels and advanced data preprocessing (Yu et al., 2020). We further released code allowing application of NNGP/NTK kernels to variable-length sequences and demonstrated its use on the IMDb reviews dataset. While caution is needed in extrapolation of any results, we hope that particularly Figure 3 and Table 2 inspire novel NN architectures and kernel designs.

## Acknowledgements

We thank Jaehoon Lee for frequent discussion, help with scaling up the experiments, and feedback on the manuscript. We thank Prajit Ramachandran for frequent discussion about attention architectures. We thank Greg Yang, Niki Parmar, and Ashish Vaswani, for useful discussion and feedback on the project. Finally, we thank Sam Schoenholz for insightful code reviews.

## References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 242–252, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R. R., and Wang, R. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems 32*, pp. 8139–8148. Curran Associates, Inc., 2019.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3286–3295, 2019.

Billingsley, P. *Probability and Measure*. John Wiley and Sons, second edition, 1986.

Blum, J. R., Chernoff, H., Rosenblatt, M., and Teicher, H. Central limit theorems for interchangeable processes. *Canadian Journal of Mathematics*, 10:222–229, 1958.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., and Wanderman-Milne, S. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Chen, Y., Kalantidis, Y., Li, J., Yan, S., and Feng, J. Aˆ 2-nets: Double attention networks. In *Advances in Neural Information Processing Systems*, pp. 352–361, 2018.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *International Conference on Learning Representations*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. 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)*, pp. 4171–4186. Association for Computational Linguistics, 2019.

Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pp. 1675–1685, 2019.

Dudley, R. M. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2nd edition, 2002.

Garriga-Alonso, A., Rasmussen, C. E., and Aitchison, L. Deep convolutional networks as shallow gaussian processes. In *International Conference on Learning Representations*, 2019.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8571–8580. Curran Associates, Inc., 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. Efficient backprop. *Lecture notes in computer science*, pp. 9–50, 1998.

Lee, J., Sohl-dickstein, J., Pennington, J., Novak, R., Schoenholz, S., and Bahri, Y. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pp. 8570–8581, 2019.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*, pp. 8157–8166, 2018.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Matthews, A. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271v2*, 2018.

Neal, R. M. *Bayesian Learning for Neural Networks*. Springer, 1996.

Novak, R., Xiao, L., Bahri, Y., Lee, J., Yang, G., Abolafia, D. A., Pennington, J., and Sohl-dickstein, J. Bayesian deep convolutional networks with many channels are gaussian processes. In *International Conference on Learning Representations*, 2019.

Novak, R., Xiao, L., Hron, J., Lee, J., Alemi, A. A., Sohl-Dickstein, J., and Schoenholz, S. S. Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

Pennington, J., Socher, R., and Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014. URL https://nlp.stanford.edu/projects/glove/.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proc. of NAACL*, 2018. URL https://allennlp.org/elmo.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. In *Advances in Neural Information Processing Systems 32*, pp. 68–80. 2019.

Shaw, P., Uszkoreit, J., and Vaswani, A. 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)*, 2018.

Thongtan, T. and Phienthrakul, T. Sentiment classification using document embeddings trained with cosine similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pp. 407–414, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2057. URL https://www.aclweb.org/anthology/P19-2057.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., and Tang, X. Residual attention network for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3156–3164, 2017.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7794–7803, 2018.

Woo, S., Park, J., Lee, J.-Y., and So Kweon, I. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019a.

Yang, G. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In *Advances in Neural Information Processing Systems 32*, pp. 9947–9960. Curran Associates, Inc., 2019b.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.

Yu, D., Wang, R., Li, Z., Hu, W., Salakhutdinov, R., Arora, S., and Du, S. S. Enhanced convolutional neural kernels, 2020. URL https://openreview.net/forum?id=BkgNqkHFPr.