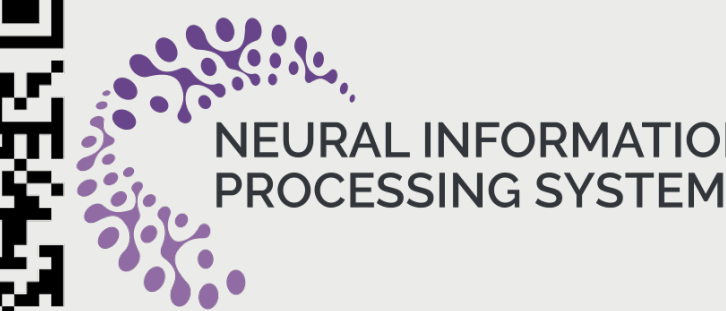
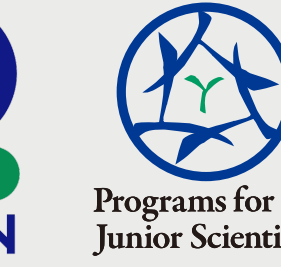


Infinite-Width Limit of a Single Attention Layer: Analysis via Tensor Programs

Mana Sakai^{1,3}, Ryo Karakida^{2,3}, Masaaki Imaizumi^{1,3}

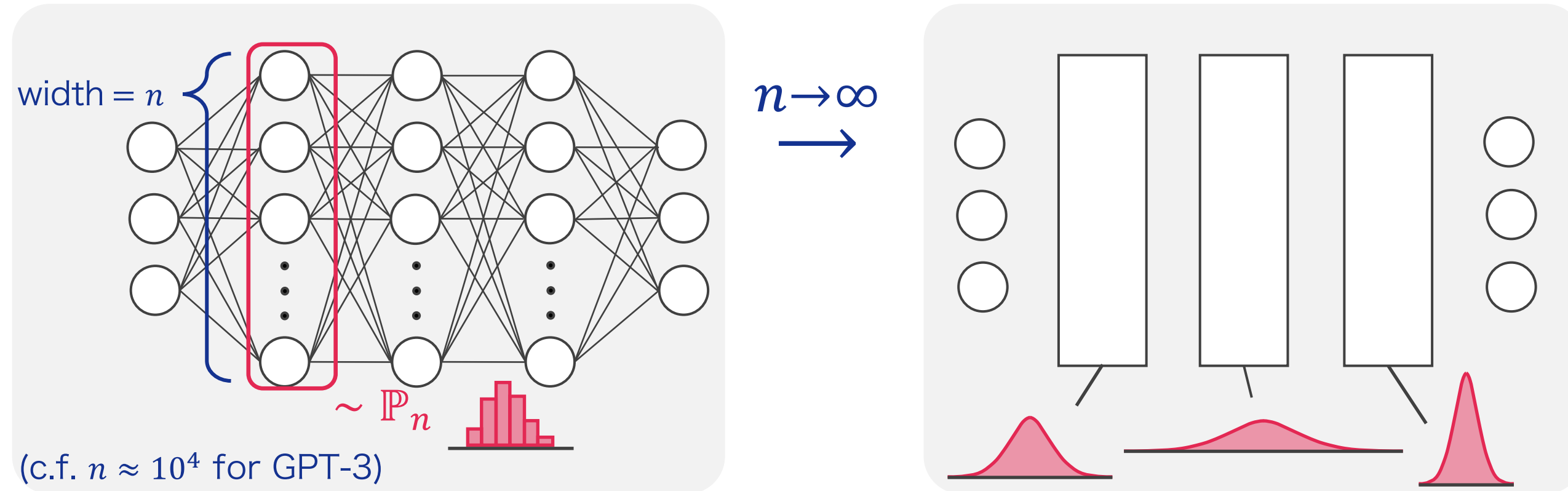
(¹The University of Tokyo ²AIST ³RIKEN AIP)



Message: The infinite-width limit of an attention layer is described by a non-Gaussian distribution

Background: Infinite-Width Limit of NNs

- Goal:** Understand how information propagates in wide NNs
- Approach:** Take the infinite-width limit and observe information propagation; check whether it explodes, vanishes, or stays stable



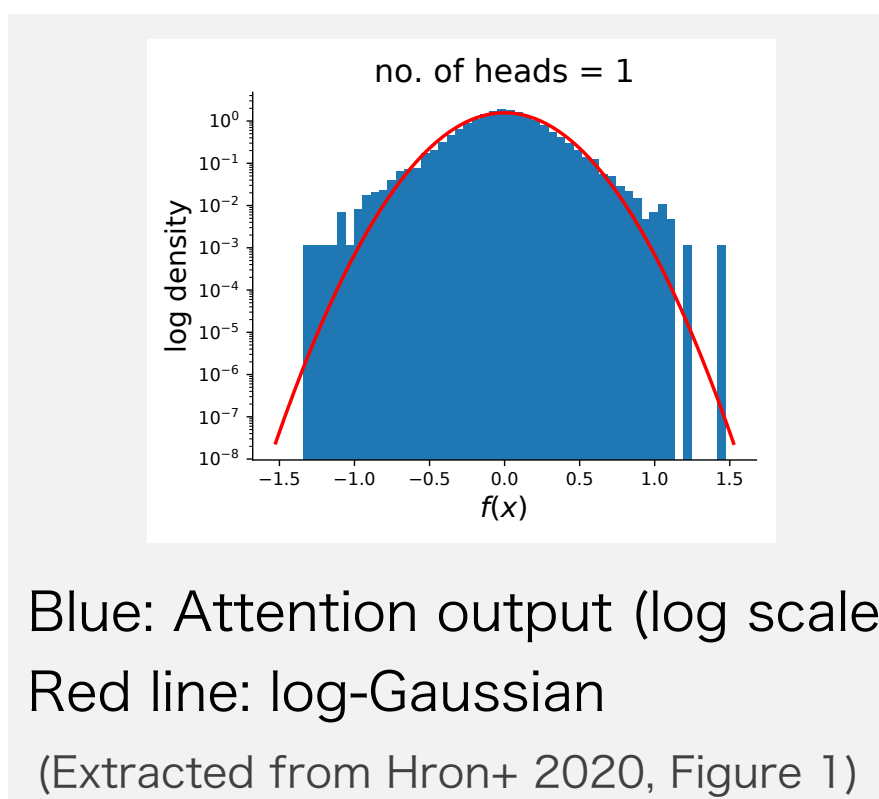
- For standard NN architectures, $\mathbb{P}_n \xrightarrow{n \rightarrow \infty} \text{Gaussian (+correction term)}$
- Notable frameworks:** **NNGP** (Lee+ 2017; Matthews+ 2018), **TP** (Yang 2019)

Attention Cannot Be Approximated by a Gaussian

- Attention output $\text{Attn}: \mathbb{R}^{s \times n} \rightarrow \mathbb{R}^{s \times n}$ is

$$\text{Attn}(X) = \frac{1}{\sqrt{H}} \sum_{a=1}^H \text{SoftMax} \left(\frac{1}{\sqrt{n}} (XW^{Q,a})(XW^{K,a})^\top \right) (XW^{V,a})W^{O,a}$$

- Attention outputs are empirically known to be non-Gaussian
- Prior work employs tailored assumptions** (e.g., infinite heads, $1/n$ -scaling of attention scores) **to make NNGP and TP applicable**
→ We derive the limit distribution of attention without relying on such assumptions



Blue: Attention output (log scale)
Red line: log-Gaussian
(Extracted from Hron+ 2020, Figure 1)

Result: Non-Gaussian Limit of Attention

i th row vector of the attention output $\text{Attn}(X) \in \mathbb{R}^{s \times n}$ is

$$y^i = \frac{1}{\sqrt{H}} \sum_{a=1}^H \sum_{j=1}^s \text{SoftMax}_j \left(p_{i,1}^{(a)}, \dots, p_{i,s}^{(a)} \right) W^{O,a} W^{V,a} x^i, \quad p_{i,j}^{(a)} = \frac{1}{\sqrt{n}} (W^{Q,a} x^i)^\top (W^{K,a} x^j) \in \mathbb{R}$$

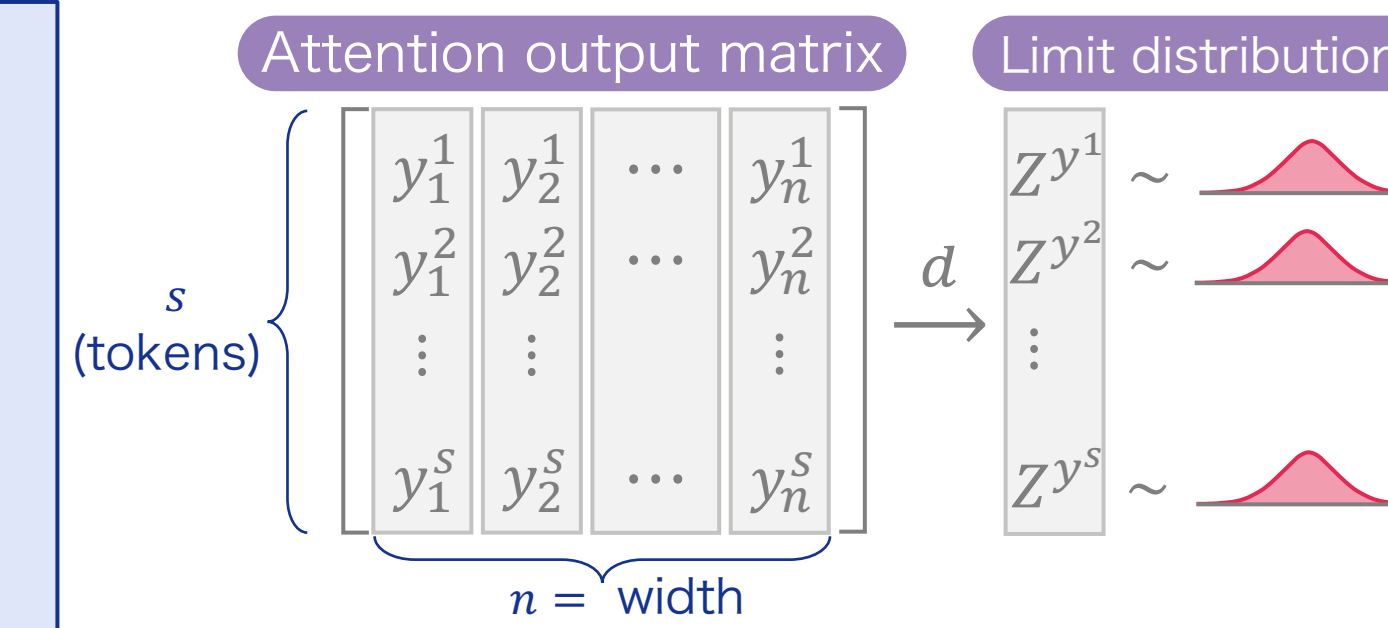
Limit Distribution of an Attention Output

Suppose the network does not contain an attention layer prior to $\{y^i\}_{i \in [s]}$. Then,

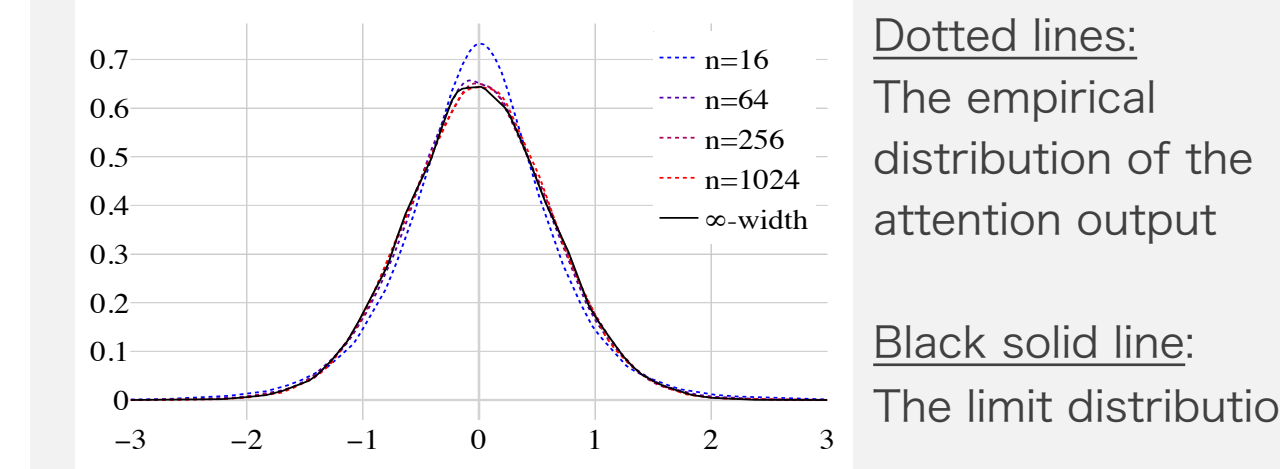
$$(y_\alpha^1, \dots, y_\alpha^s) \xrightarrow{d} (Z^{y^1}, \dots, Z^{y^s}) \quad (n \rightarrow \infty),$$
$$Z^{y^i} = \frac{1}{\sqrt{H}} \sum_{a=1}^H \sum_{j=1}^s \text{SoftMax}_j \left(p_{i,1}^{(a)}, \dots, p_{i,s}^{(a)} \right) Z^{\tilde{v}^{a,j}}$$

where

- $\{Z^{\tilde{v}^{a,j}}\}_{j \in [s], a \in [H]}$ is independent of $\{p_{i,j}^{(a)}\}_{i,j \in [s], a \in [H]}$
- $\{Z^{\tilde{v}^{a,j}}\}_{j \in [s], a \in [H]} \sim N(0, \kappa)$
- $\{p_{i,j}^{(a)}\}_{i,j \in [s], a \in [H]} \sim N(0, \kappa')$



Simulation



Implications

- $(Z^{y^1}, \dots, Z^{y^s})$ is Gaussian conditional on $\{p_{i,j}^{(a)}\}$
→ It follows a conditionally Gaussian (**hierarchical Gaussian**) structure
→ Marginally, it is inherently **non-Gaussian** and generally has heavier tails than a standard Gaussian
- A new framework distinct from existing Gaussian-based approaches is essential**

Architectures with Gaussian-based limit distribution

MLP CNN LSTM etc.

Architectures with non-Gaussian-based limit distribution

attention

General Result Using Tensor Programs

NETSOR Program

Consider a feed-forward neural network as a finite set of random vectors $h^1, \dots, h^J \in \mathbb{R}^n$, which is **inductively generated** as follows:

- $\mathcal{V}_0 \subset \{h^1, \dots, h^J\}$: fixed set of initial vectors (input layer)
- Each $h^k \notin \mathcal{V}_0$ is generated either by:
 - MatMul**: matrix multiplication $h^k = W h^j$, $W \in \mathbb{R}^{n \times n}$
 - Nonlin**: coordinatewise nonlinearity $h^k = \phi(h^{j_1}, \dots, h^{j_m})$, $h_\alpha^k = \phi(h_\alpha^{j_1}, \dots, h_\alpha^{j_m})$ ($\alpha \in [n]$)

Setup & Assumptions

- $r, m \in \mathbb{N}$ satisfy $m \geq 2r$
- Consider a **NETSOR program**, and suppose all nonlinearities used in Nonlin are pseudo-Lipschitz
- $g^1, \dots, g^m \in \mathbb{R}^n$ are vectors in NETSOR generated by MatMul
- A subset $\{g^{i,j}\}_{i \in [r], j \in [2]} \subset \{g^1, \dots, g^m\}$ is defined by $g^{i,j} = W^{i,j} x^{i,j}$, $x^{i,j} = \phi^{i,j}(g^1, \dots, g^m)$
- Each $\phi^{i,j}$ is bounded and pseudo-Lipschitz
- The weight matrices $W^{i,j} \in \mathbb{R}^{n \times n}$ satisfy:
 - $\{W^{i,j}\}_{i \in [r], j \in [2]}$ is not used for any $g \in \{g^1, \dots, g^m\} \setminus \{g^{i,j}\}_{i \in [r], j \in [2]}$
 - $W^{i,j}$ may be the same matrix as $W^{i',j'}$ unless $i = i', j \neq j'$
- Define the scalar dot-products by $p_i = n^{-1/2} (g^{i,1})^\top g^{i,2}$ ($i \in [r]$)

Theorem (informal)

Let $h^1, \dots, h^k \in \mathbb{R}^n$ be vectors whose elements are given by

$$h_\alpha^j = \varphi^j(g_\alpha^1, \dots, g_\alpha^m, p_1, \dots, p_r) \quad (\alpha \in [n], j \in [k]),$$

where each φ^j is pseudo-Lipschitz. Then, for any bounded and pseudo-Lipschitz function $\psi: \mathbb{R}^k \rightarrow \mathbb{R}$, we have

$$\frac{1}{n} \sum_{\alpha=1}^n \psi(h_\alpha^1, \dots, h_\alpha^k) \xrightarrow{d} \mathbb{E} \left[\psi(Z^{h^1}, \dots, Z^{h^k}) \mid p_1^*, \dots, p_r^* \right] \quad (n \rightarrow \infty),$$

where

- $Z^{h^j} = \varphi^j(Z^{g^1}, \dots, Z^{g^m}, p_1^*, \dots, p_r^*)$ ($j \in [k]$).
- $(Z^{g^1}, \dots, Z^{g^m})$ is statistically independent of (p_1^*, \dots, p_r^*) .
- $(Z^{g^1}, \dots, Z^{g^m})$ is defined as in Yang (2019).

- (p_1^*, \dots, p_r^*) is Gaussian with $\mathbb{E}(p_i^*) = 0$ and $\text{Cov}(p_k^*, p_i^*) = E \left[Z^{g^{i,1}} Z^{g^{i,2}} Z^{g^{k,1}} Z^{g^{k,2}} \right]$, where $\{Z^{g^{i,j}}\}_{i \in [r], j \in [2]}$ is defined as in Yang (2019).

As a corollary, we have $(h_\alpha^1, \dots, h_\alpha^k) \xrightarrow{d} (Z^{h^1}, \dots, Z^{h^k})$ ($n \rightarrow \infty$).

References

> Lee et al. (2017). Deep Neural Networks as Gaussian Processes.
> Matthews et al. (2018). Gaussian Process Behaviour in Wide Deep Neural Networks.
> Hron et al. (2020). Infinite attention: NNGP and NTK for deep attention networks.
> Yang (2019). Tensor Programs I: Wide Feedforward or Recurrent Neural Networks of Any Architecture are Gaussian Processes.