Customizing the Inductive Biases of Softmax Attention using Structured Matrices

Yilun Kuang Noah Amsel Sanae Lotfi Shikai Qiu Andres Potapczynski Andrew Gordon Wilson

NYU

Notation

- D embedding dimension
- T sequence length
- $\mathbf{X}$ input matrix ($T \times D$)
- H number of heads
- $r := H / D$ head dimension ("rank")
- $\mathbf{W}$ weight matrix ($D \times r$)
- $\sigma(\cdot)$ row-wise softmax

Our goal: Improve the inductive bias of attention by changing the *structure* of its scoring function

Standard Attention's "Scoring Function" is Low Rank

- Standard attention layer computes:

$$\sum_{i=1}^{H} \sigma\left(\mathbf{X}\mathbf{W}_{Q_i}\mathbf{W}_{K_i}^{\top}\mathbf{X}^{\top}\right)\mathbf{X}\left(\mathbf{W}_{V_i}\mathbf{W}_{O_i}^{\top}\right)$$

- The standard "scoring function" between two tokens is "query dot key"

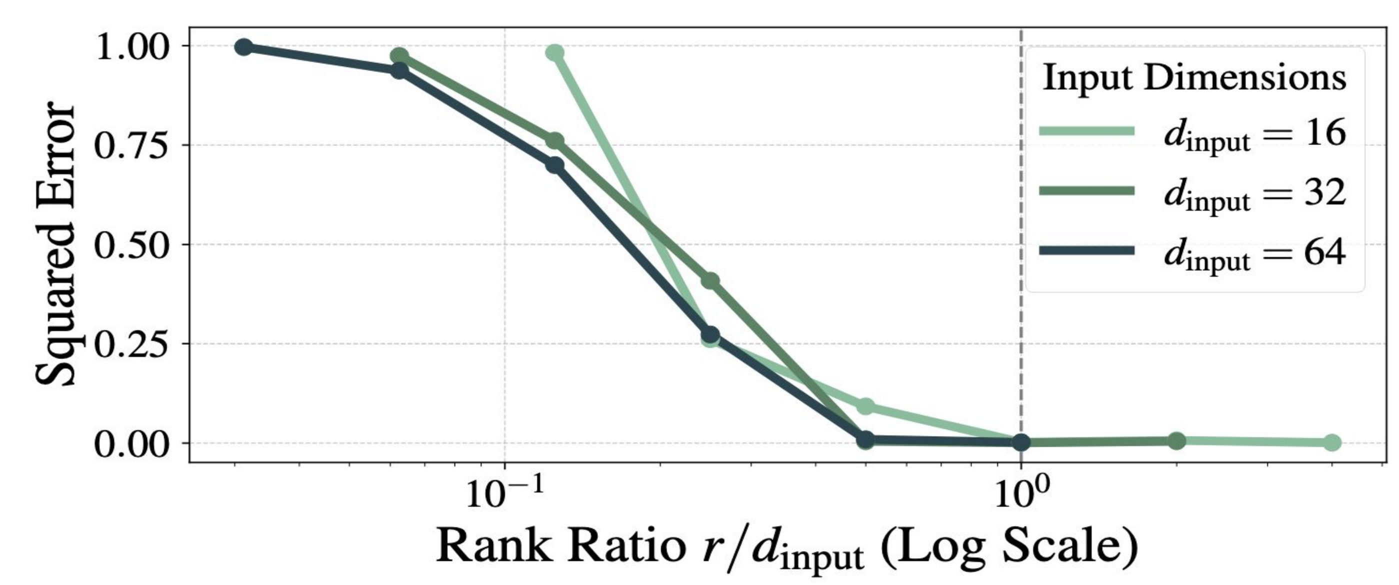
$$s(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\top}\mathbf{W}_{Q_i}\mathbf{W}_{K_i}^{\top}\mathbf{x}'$$

- Bilinear form parameterized by a $D \times D$, rank-r matrix $\mathbf{W}_{Q_i}\mathbf{W}_{K_i}^{\top}$
- Invariant to position of the tokens in the sequence

Weakness of Standard Attention

The Low Rank Bottleneck

- When $r < d$, scoring function loses info about inputs. E.g., it cannot compute $\mathbf{x} \bullet \mathbf{x}'$
- Makes tasks like "in-context regression" impossible:

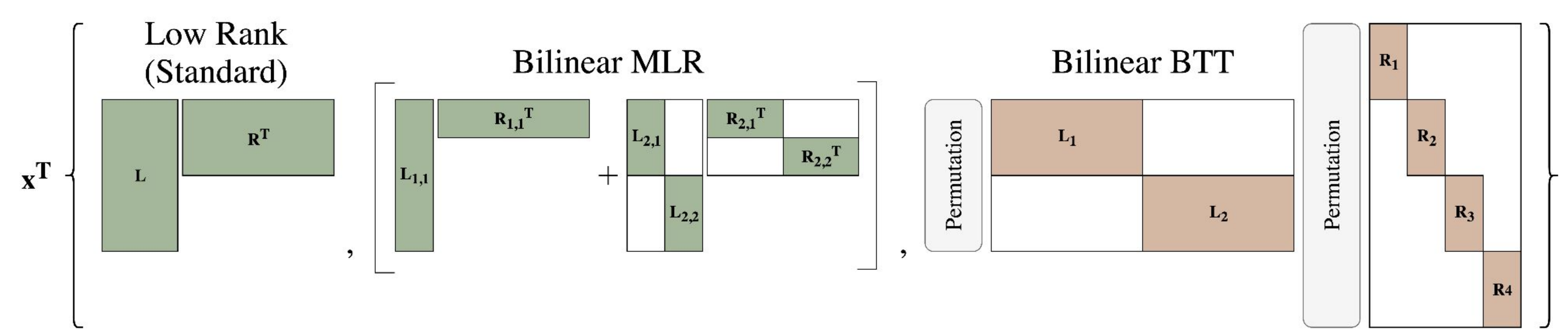


No Locality Bias

- For pairs of nearby tokens, we need fine-grained attention = accurate scores
- For pairs that are far-apart in the sequence, we do not
- Yet we spend the same FLOPs computing attention scores regardless of tokens' distance. That's wasteful

Solution via Structured Matrices

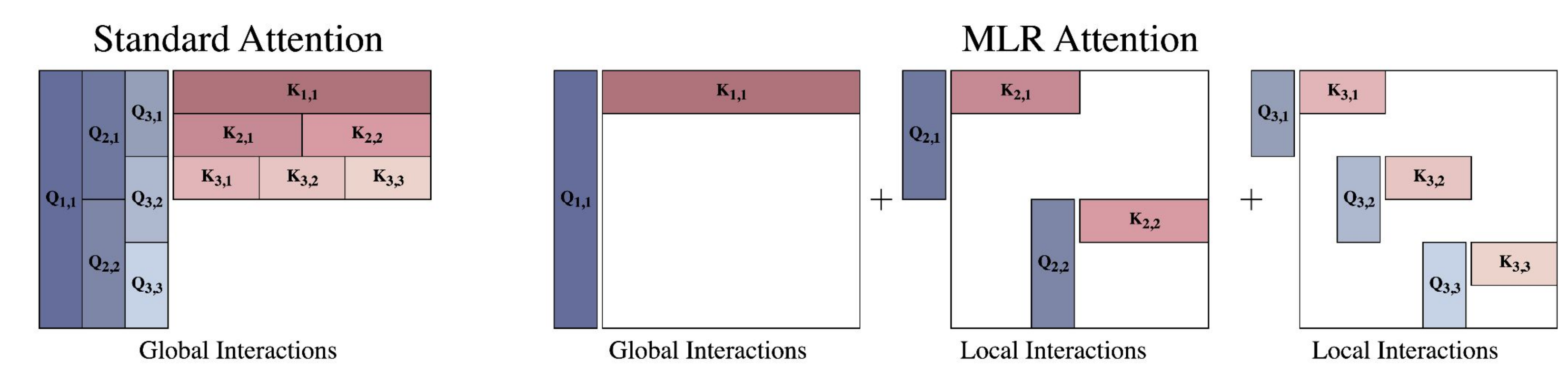
Reparameterizing the Score's Bilinear Form as an MLR or BTT matrix



Structure	Definition	Parameters	Rank
Dense	$\mathbf{W}$	D^2	D
Low Rank	$\mathbf{L}\mathbf{R}^{\top}$	$2Dr$	r
Multi-Level Low Rank (MLR)	$\sum_{l=1}^{L}\bigoplus_{k=1}^{p_l}\mathbf{L}_{l,k}\mathbf{R}_{l,k}^{\top}$	$2D\sum_l r_l$	$\sum_l r_l p_l$
Block Tensor Train (BTT)	$\mathbf{P}_L(\bigoplus_{k'=1}^{b}\mathbf{L}_{k'})\mathbf{P}_R(\bigoplus_{k=1}^{b}\mathbf{R}_k^{\top})$	$2D^{\frac{3}{2}}s$	D

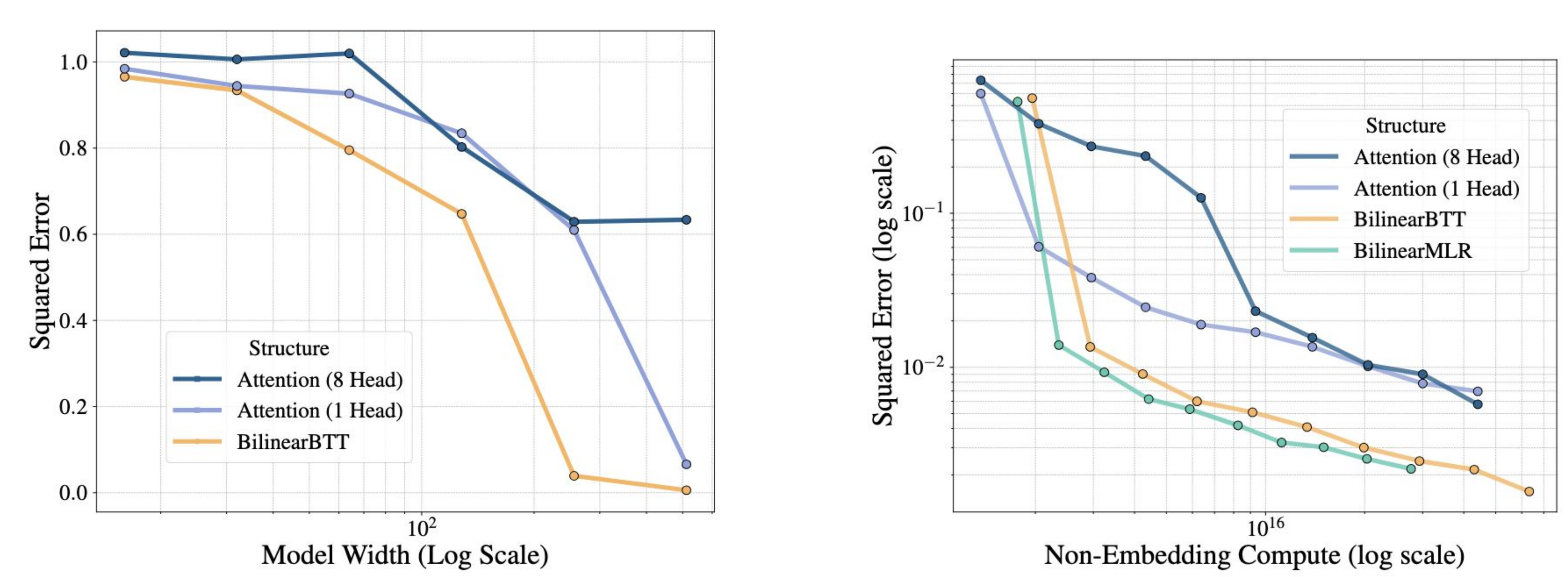
Distance-Dependent Scoring Function

- Combine queries and keys by making them the left and right factors of an MLR matrix
- Effectively use a smaller head dimension r when the tokens are far apart. Saves FLOPs



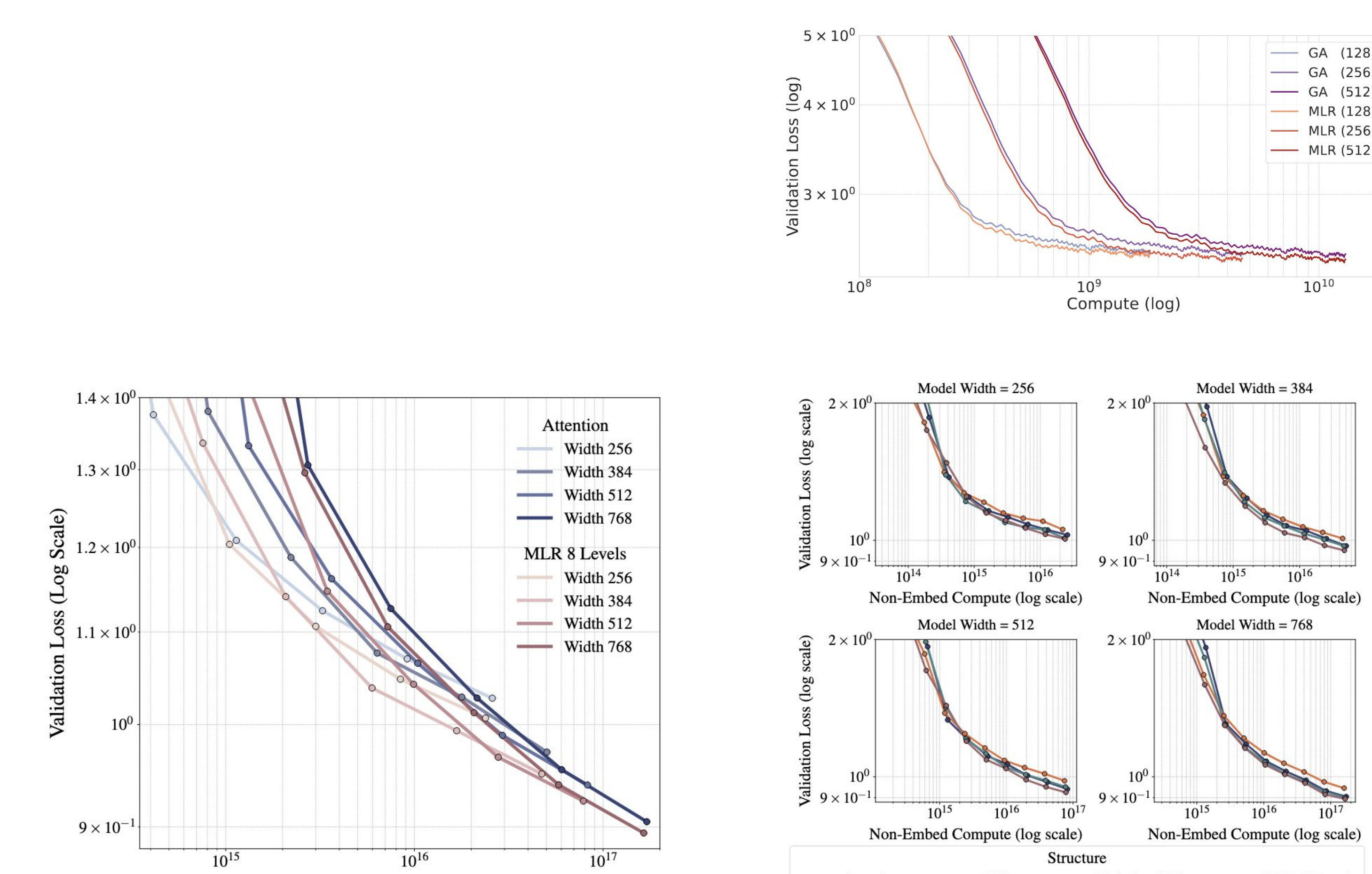
Experimental Results

Better Accuracy on In-Context Regression



(a) Input Dimension (d_{input}) = 128. (b) Input Dimension (d_{input}) = 64.

Better Accuracy on Language Modeling and Time-Series Forecasting



(a) Scaling Law on OpenWebText (b) Different Forms of Distance-Dependent Compute