Research Article

ARWKV: Pretraining Is Not What We Need – An RNN-Attention-Based Language Model Born From Transformer

Yueyu Lin[1], Zhiyuan Li[1], Peter Yue[1], Xiao Liu[1]

[1]. Independent researcher

As is known, hybrid quadratic and subquadratic attention models in multi-head architectures have surpassed both Transformer and Linear RNN models[1], with these works primarily focusing on reducing KV complexity and improving efficiency. For further research on expressiveness*, we introduce our series of models distilled from Qwen 2.5, based on pure native RWKV-7 attention, which aims to make RNN more expressive and demonstrates state tracking ability beyond transformers. We work with QRWK 32B† based on RWKV-6 architecture, another approach that reduces the entire knowledge processing time to just 8 hours using 16 AMD MI300X GPUs while maintaining Qwen 2.5's performance. In fact, the distillation process can utilize any LLM, not just Qwen, and enables knowledge transfer from larger LLMs to smaller ones with more fewer tokens. We will explain the detailed process and share our insights on building more powerful foundation models. Please note that this is an ongoing work that will be updated continuously. The model checkpoints and source code are available at https://github.com/yynil/RWKVInside, https://huggingface.co/RWKV-Red-Team/ARWKV-7B-Preview-0.1.

1. Introduction

The emergence of Linear RNNs (LRNNs) has grown rapidly, with models like RWKV[2], DeltaNet[3], Mamba-2[4], GLA[5], and others showing strong competitiveness with transformers. However, these models face natural penalties in context learning and long-context retrivel due to subquadratic attention limitations. Recently, RWKV-7 introduced a promising architecture, where a 0.1B parameter model achieved perfect results in 16k passkey retrieval[‡]. With its transition matrix having wider eigenvalues[6], it demonstrates stronger state tracking capabilities than transformers[7]. While Qwen 2.5 was trained on 18 trillion tokens requiring enormous GPU resources, making pretraining impractical for academic use, we bridge this gap by refining the distillation method. Our approach enables training a 7B parameter model on a single A100 80G GPU, while 4x8 A100s can handle a 32B model.

2. Architecture

We know qwen 2.5 dense models is the Transformer-based decoder architecture with group query attention, SwiGLU activation function for non-linear activation, Rotary Positional Embeddings for encoding position information, QKV bias in the attention mechanism and RMSNorm[8]. We only keep RMSnorm and SwiGLU activiation and replace the rest with rwkv-7 attention.In GQA, queries are grouped while keys and values remain separate:

$$\mathrm{GQA}(Q_g, K, V) = \mathrm{concat}[\mathrm{head}_1, \ldots, \mathrm{head}_g]W^O \tag{1}$$

where each head is computed as:

$$\mathrm{head}_i = \mathrm{Attention}(Q_i W_i^Q, K W_i^K, V W_i^V) \tag{2}$$

In RWKV-7 attention which is the time mixing:

$$\mathbf{State}_t = \mathbf{State}_{t-1}(\mathrm{diag}(w_t) - \hat{\kappa}_t^T(a_t \circ \kappa_t)) + v_t^T \odot \tilde{k}_t \tag{3}$$

the state means matrix-valued attention state.where a is the in-context learning rate.

In the comparing with RWKV-6[2]:

$$\mathrm{state}' = \mathrm{diag}(w) \cdot \mathrm{state} + \mathbf{k}^\top \cdot v \tag{4}$$

then we just relpace self-attention in every layer with RWKV-7 time mixing module in Figure 1.

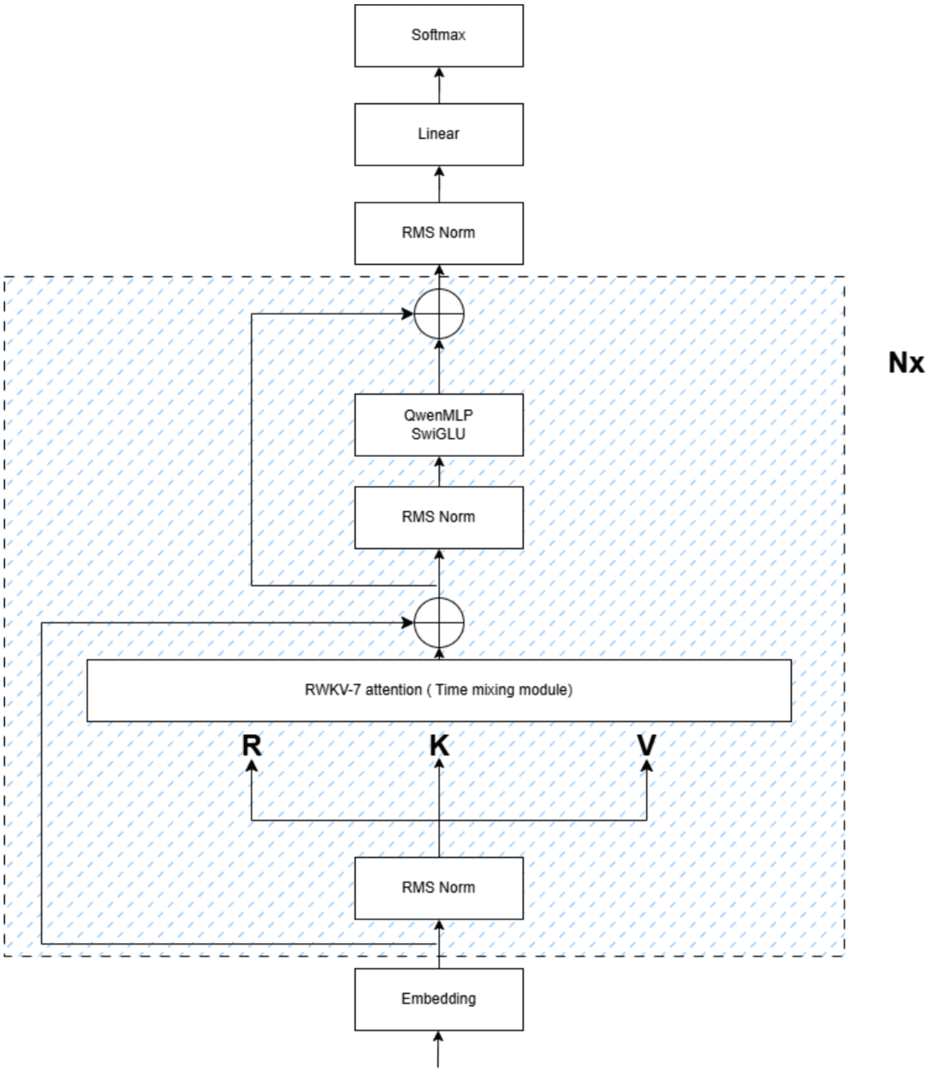

Figure 1. replace self-attention by RWKV-7 time mixing module

3. From Transformer to RNN

Inspired by MambaInLLaMA[4] and Phi-Mamba, we aimed to produce a pure RNN model similar to QRWKV. We tested both layer-wise distillation and one-step distillation methods, finding the latter to be more efficient. Our research revealed that using attention alignment in stage 1 is crucial for maintaining the original model's performance. By mimicking self-attention through RWKV's time mixing module, we can preserve attention expressiveness while transforming the state attention to function more like a meta-learner[9].

3.1. Stage 1 - Time Mixing module replacing Self-Attention

In this stage,we align the hidden state output between the student and teacher attention block[10], frozen the MLP and remove the group normalization for the attention output and make the gate initialized with 1 .[Figure 4] Context length can dramaticaly increase the trainning time, in this work we train 2048 with one h800

$$L_{special} = \|\mathbf{h}_{teacher} - \mathbf{h}_{student}\|_2 \cdot (d_{model})^{-0.5} \tag{5}$$

where d_{model} is dimension size of hidden state.

We found that initializing state attention from teacher's attention is not necessary. The convergence speed and final values of the loss [Figure 5] indicate the target attention ability to capture the teacher model's internal attention representations.[10]

due to the fixed state size in rwkv-7 time mixing module, we can see it as a compression process[11] , or it can be seen attention as a map between probability measures[12],

3.2. Stage 2 - Knowledge Distillation

Divergence-based methods minimize diver gence between the probability distributions of the teacher and student models[13], We adopt word-level KL-Divergence instead of sequence-level knowledge distillation (SeqKD).In practice, we distill from 32B to 7B model.

For the dataset, we balance its distribution[14] based on stage 1 training, achieving fast convergence with only 20M tokens.

We then experiment with whether or not to freeze the MLP layer and introduce a gate-free technique which disables the gate mechanism entirely

3.3. Stage 3 - SFT and DPO

During this stage, we use supervised fine-tuning (SFT) to expand the model's context length and Direct Preference Optimization (DPO) to align with user preferences. Our training process involves 20M tokens in stage 1, 40M tokens in stage 2, and 770M tokens in stage 3 for context length extension.

4. Evaluation

We conducted ablation experiments between 7B models distilled using different approaches in stage 2. By controlling the presence of Gate, freeze MLP, size of teacher model, we distill model not-freezing MLP(ARWKV-M), model with GATE and not freezing MLP(ARWKV-G-M), model without Gate nor freezing MLP(ARWKV) and model with teacher model as QWEN2.5-32B-Instruct(ARWKV-from32B). All models are distilled from QWEN2.5-7B-Instruct except ARWKV-from32B. After stage-2 training, all models are tested with several benchmarks to demonstrate the impact of different training factors on the final model performance.

Moreover, we found that although we trained the model using bfloat16 (BF16), performing inference with float16 (FP16) significantly improved the performance, This differs from the original RWKV implementation, which required careful tuning of the layer scaling parameters to avoid overflow.

Our analysis of Table 1 reveals that knowledge distillation from the 32B parameter model, when performed without gating mechanisms and with frozen MLPs, yields suboptimal results. This performance degradation may be attributed to the limited capacity of the 7B model's MLP layers to effectively accommodate and adapt to the more sophisticated attention patterns learned by the 32B model's architecture. This observation suggests a potential architectural mismatch in the direct transfer of attention mechanisms between models of significantly different scales.

doi.org/10.32388/MBSRAR

	Qwen2.5-7B-Instruct	ARWKV	active MLP	w/ gate & active MLP	ARWKV-from32B
MMLU	71.72	62.41	58.22	64.77	61.78
Squad	47.89	40.05	40.35	38.74	39.01
GPQA(Diamond)	49.0	45.5	51.1		
WinoGrande	71.35	68.67	69.67	68.98	68.35
GSM8K	82.34	39.95	51.93	47.99	43.44
IfEval	73.62	52.16	48.68	52.16	44.12
Arc-c	54.86	52.22	53.52	52.22	50.77

Table 1. This is a ongoing work, benchmark based on stage-2 , currently we limit the context length to 2048 and use same datasets

5. Conclusions

We demonstrate that attention alignment combined with knowledge distillation can effectively transform transformer attention patterns into RNN-based attention in a straightforward manner.In some hybrid architecture, this attention module serves as a memory component[1], different kind of attention introduce unique inductive bias[§], bring more expressive power.

6. Future Work

For our subsequent phase of investigation, we will implement Stage 3 post-training to replicate the reasoning capabilities demonstrated by deepseek-R1[15] models.

Furthermore, we propose to generalize this methodology across diverse architectural paradigms, encompassing Mixture-of-Experts (MoE) frameworks, multimodal architectures, hybrid architectures and model compression scenarios. This expansion aims to validate the robustness and transferability of our approach across different computational paradigms.

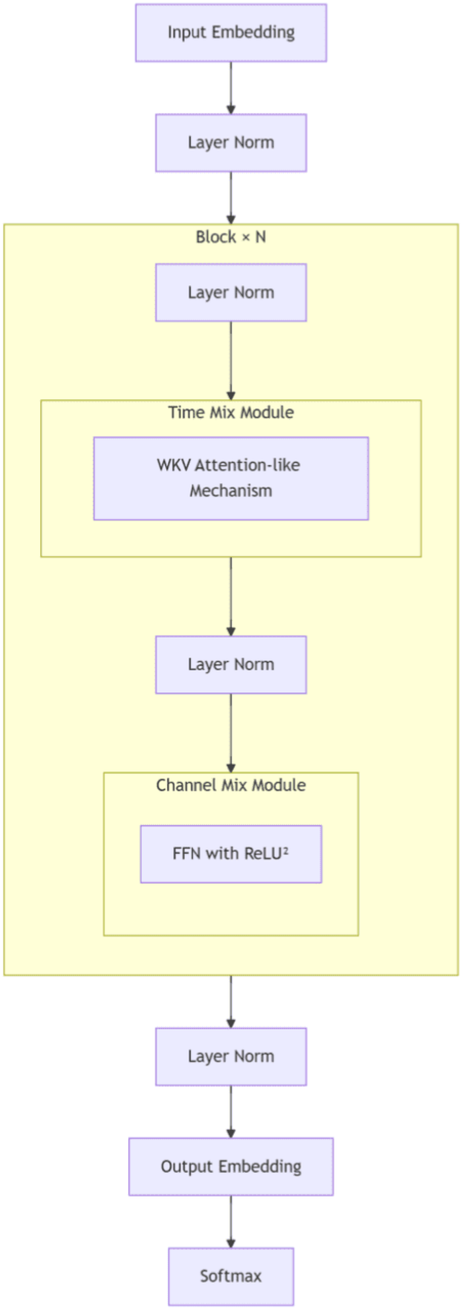

Figure 2. RWKV-7 architecture.capability of attention is the key for RNN-based LLMs, which in this case is Time mixing module

doi.org/10.32388/MBSRAR

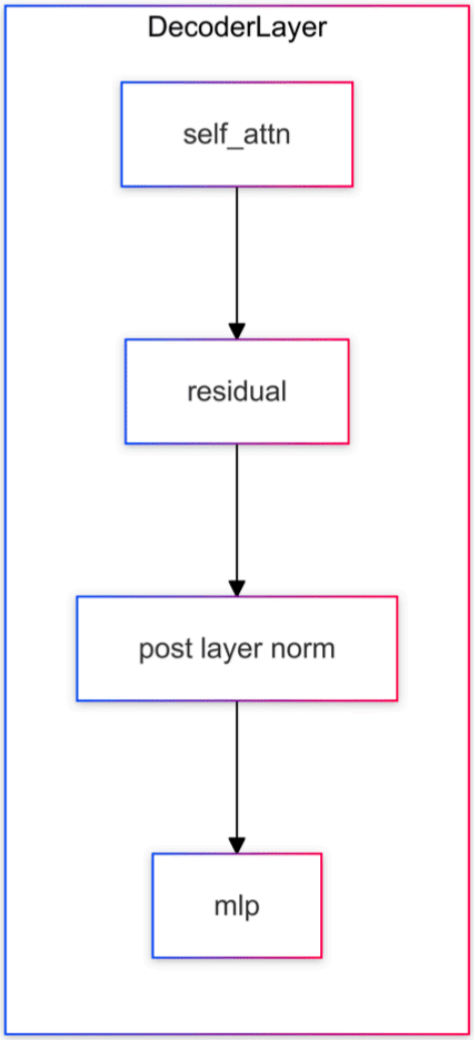

Figure 3. General Decoder Layer in transformer

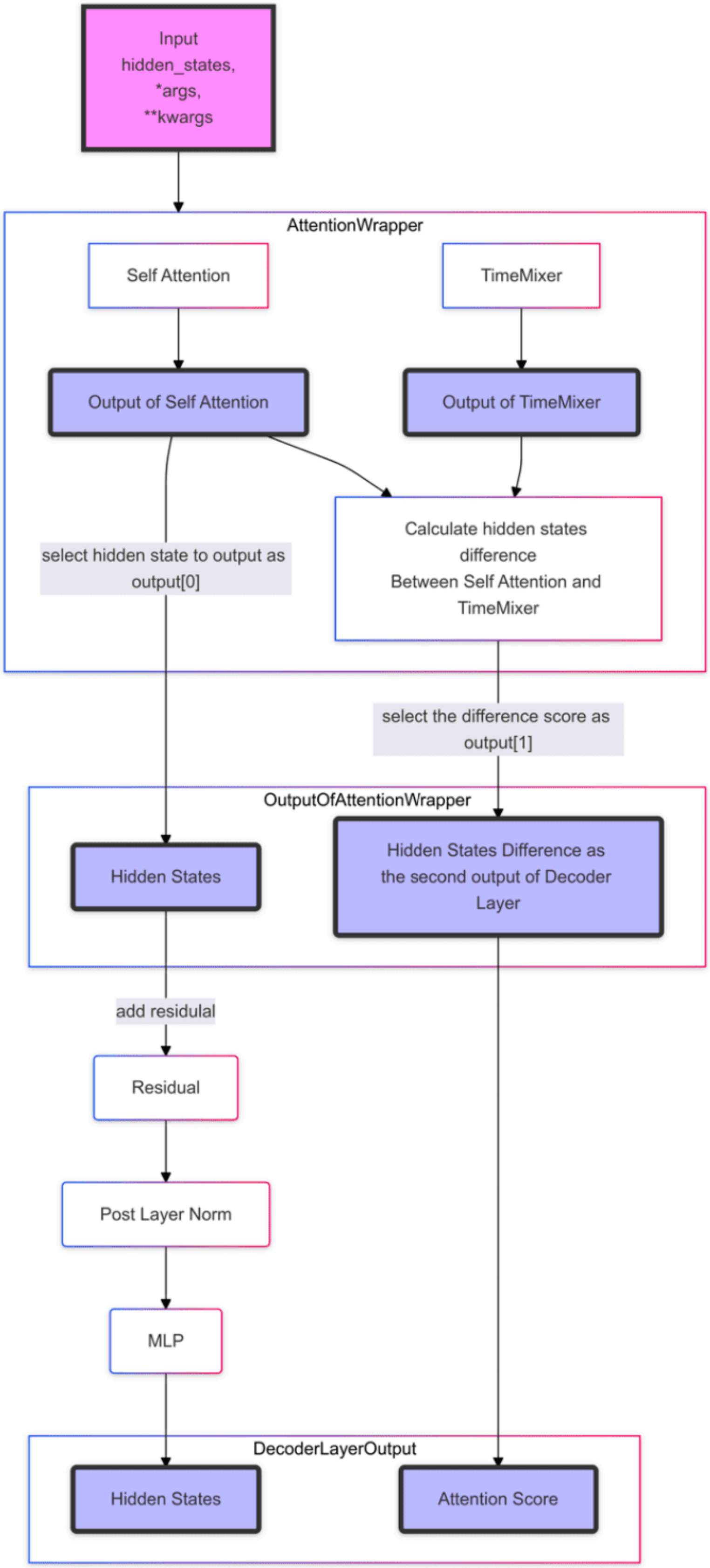
Input
hidden_states,
*args,
**kwargs
AttentionWrapper
Self Attention
TimeMixer
Output of Self Attention
Output of TimeMixer
select hidden state to output as output[0]
Calculate hidden states difference Between Self Attention and TimeMixer
select the difference score as output[1]
OutputOfAttentionWrapper
Hidden States
Hidden States Difference as the second output of Decoder Layer
add residual
Residual
Post Layer Norm
MLP
DecoderLayerOutput
Hidden States
Attention Score

Figure 4. We replace the standard Attention with an AttentionWrapper that contains both the original self-attention mechanism and a TimeMixer. The TimeMixer is trained to minimize the gap between its output and that of the self-attention module. The final output combines the hidden states from the original self-attention with the residual difference between self-attention and TimeMixer outputs. This architecture enables the model to optimize the TimeMixer to progressively reduce the discrepancy between self-attention and TimeMixer outputs.

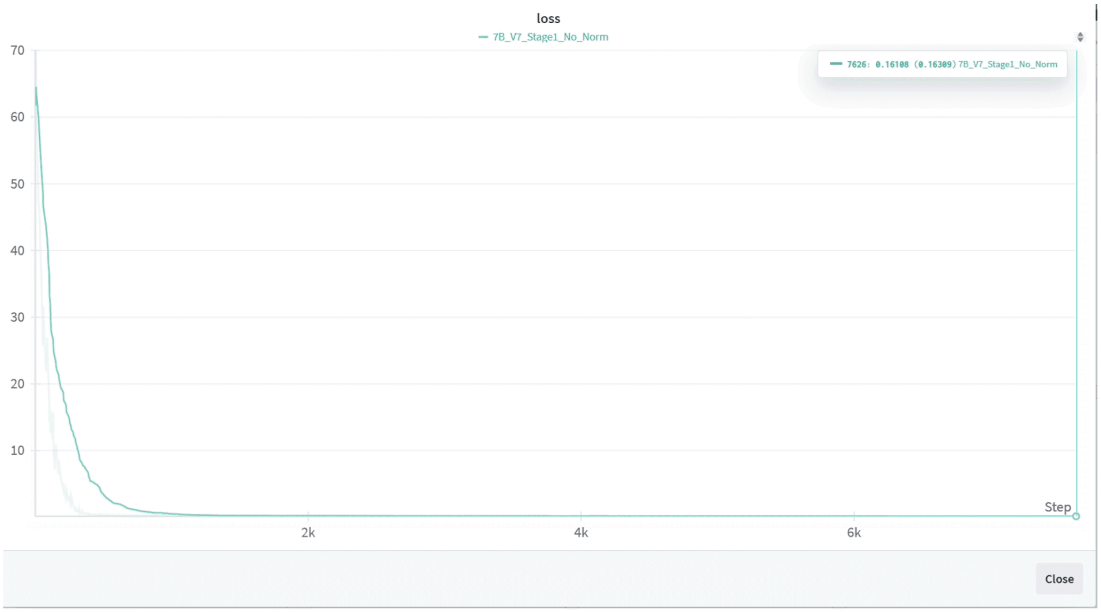

Figure 5. Stage-1 loss, 18 hours with one 8*h800 80G , context length 2048, 4B tokens

Acknowledgements

We would like to express our gratitude to Yuhang He for his endorsement.

Footnotes

[*] https://github.com/Jellyfish042/Sudoku-RWKV, https://github.com/Jellyfish042/RWKV_Othello

[†] https://huggingface.co/recursal/QRWKV6-32B-Instruct-Preview-v0.1

[‡] https://x.com/BlinkDL_AI/status/1869433254425833487

[§] https://medium.com/@felixhill/the-agreeable-lesson-9766382c6d83

References

1. [a, b]*Dong X, Fu Y, Diao S, Byeon W, Chen Z, Mahabaleshwarkar AS, Liu S-Y, Van Keirsbilck M, Chen M-H, Suhara Y, et al. Hymba: A Hybrid-head Architecture for Small Language Models. arXiv preprint arXiv:2411.13676. 2024.*
2. [a, b]*Peng B, Goldstein D, Anthony Q, Albalak A, Alcaide E, Biderman S, Cheah E, Du X, Ferdinan T, Hou H, et al. Eagle and finch: Rwkv with matrix-valued states and dynamic recurrence. arXiv preprint arXiv:2404.05892. 2024.*
3. [^]*Yang S, Wang B, Zhang Y, Shen Y, Kim Y (2024). "Parallelizing Linear Transformers with the Delta Rule over Sequence Length". arXiv preprint arXiv:2406.06484. Available from: https://arxiv.org/abs/2406.06484.*
4. [a, b]*Wang J, Paliotta D, May A, Rush AM, Dao T (2024). "The mamba in the llama: Distilling and accelerating hybrid models". arXiv preprint arXiv:2408.15237.*
5. [^]*Yang S, Wang B, Shen Y, Panda R, Kim Y (2023). "Gated linear attention transformers with hardware-efficient training". arXiv preprint arXiv:2312.06635.*
6. [^]*Grazzi R, Siems J, Franke JK, Zela A, Hutter F, Pontil M (2024). "Unlocking State-Tracking in Linear RNNs Through Negative Eigenvalues". arXiv preprint arXiv:2411.12537.*
7. [^]*Merrill W, Petty J, Sabharwal A (2024). "The illusion of state in state-space models". arXiv preprint arXiv:2404.08819. Available from: https://arxiv.org/abs/2404.08819.*
8. [^]*Yang A, Yang B, Zhang B, Hui B, Zheng B, Yu B, Li C, Liu D, Huang F, Wei H, et al. (2024). "Qwen2. 5 Technical Report". arXiv preprint arXiv:2412.15115.*
9. [^]*Hospedales T, Antoniou A, Micaelli P, Storkey A (2021). "Meta-learning in neural networks: A survey". IEEE Transactions on Pattern Analysis and Machine Intelligence.* **44** *(9): 5149–5169.*
10. [a, b]*Bick A, Li KY, Xing EP, Kolter JZ, Gu A (2024). "Transformers to ssms: Distilling quadratic knowledge to subquadratic models". arXiv preprint arXiv:2408.10189.*
11. [^]*Skean O, Arefin MR, LeCun Y, Shwartz-Ziv R (2024). "Does Representation Matter? Exploring Intermediate Layers in Large Language Models". arXiv preprint arXiv:2412.09563.*

12. ^*Castin V, Ablin P, Peyr\u{e} G (2024). "How Smooth Is Attention?" In: ICML 2024.*
13. ^*Xu X, Li M, Tao C, Shen T, Cheng R, Li J, Xu C, Tao D, Zhou T (2024). "A survey on knowledge distillation of large language models". arXiv preprint arXiv:2402.13116.*
14. ^*Xu Z, Jiang F, Niu L, Deng Y, Poovendran R, Choi Y, Lin BY (2024). "Magpie: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing". arXiv preprint arXiv:2406.08464.*
15. ^*Guo D, Yang D, Zhang H, Song J, Zhang R, Xu R, Zhu Q, Ma S, Wang P, Bi X, et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv preprint arXiv:2501.12948. 2025.*

Declarations

Funding: No specific funding was received for this work.

Potential competing interests: No potential competing interests to declare.