Stat 9911
Principles of AI: LLMs
Large Language Model Architectures 04
Specific LLMs

Edgar Dobriban

Department of Statistics and Data Science, the Wharton School, University of Pennsylvania

February 2, 2025

Penn
University of Pennsylvania

Wharton
University of Pennsylvania

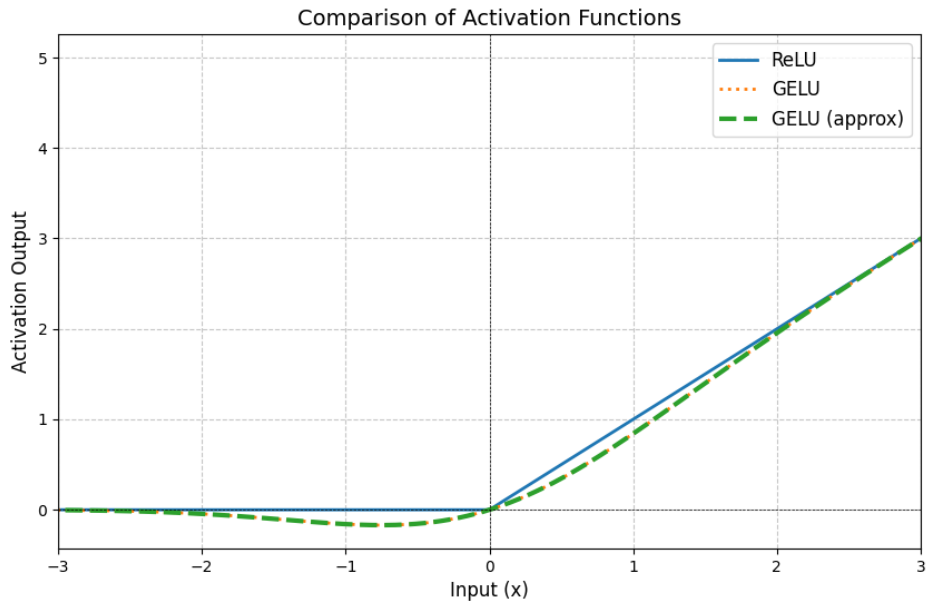
Plan

- We plan to discuss specific LLM families such as GPT, Llama, DeepSeek, LLM360.

Table of Contents

GPT Series

- GPT series (Radford et al., 2018, 2019; Brown et al., 2020; OpenAI, 2023)
- GPT-1: Gaussian Error Linear Unit (GELU) activation (Hendrycks and Gimpel, 2016): $x \mapsto x \cdot \Phi(x)$, where Φ is normal cdf, or approximate $x \mapsto 0.5x\left(1+\tanh\left(\sqrt{\frac{2}{\pi}}\left(x+0.044715x^3\right)\right)\right)$ (Choudhury, 2014).



- GPT-2: Modified initialization: "We scale the weights of residual layers at initialization by a factor of $1/N^{1/2}$ where N is the number of residual layers."

GPT-3 (Brown et al., 2020) Model Details

- "Alternating dense and locally banded sparse attention patterns (similar to the Sparse Transformer (Child et al., 2019))"
- GPT-3 with 175B parameters
  - Context window: $T = 2,048$ tokens
  - Layers: 96
  - Embedding rep: $d = 12,288$
  - Feedforward rep: $d' = 4d$
  - Number of attention heads: $H = 96$, Dimension per head: $d/H = 128$

LLama Series

- LLama 1 (Touvron et al., 2023a):
  - RMSNorm pre-normalization (Zhang and Sennrich, 2019).
  - FFN layer: SwiGLU (Shazeer, 2020):
    $x \mapsto \text{swish}(Wx + b) \odot (Vx + c)$, where $\text{swish}(z) = z/(1 + \exp(-z))$ and W, V, b, c are learnable[1]
  - Rotary Position Embeddings (Su et al., 2024).
- LLama 2 (Touvron et al., 2023b):
  - Grouped-query attention (GQA) (Ainslie et al., 2023).
- LLama 3 (Dubey et al., 2024):
  - "Attention mask that prevents self-attention between different documents within the same sequence."
  - 405-B:
    - Context window: $T = 128K$ tokens
    - Layers: 126
    - Embedding rep: $d = 16,384$
    - Feedforward rep: $d' = 20,480$
    - Number of attention heads: $H = 128$. Key-value heads: 8

[1] Shazeer (2020): "We offer no explanation as to why these architectures seem to work; we attribute their success, as all else, to divine benevolence."

Table of Contents

DeepSeek-V3 (Liu et al., 2024b): A Preview

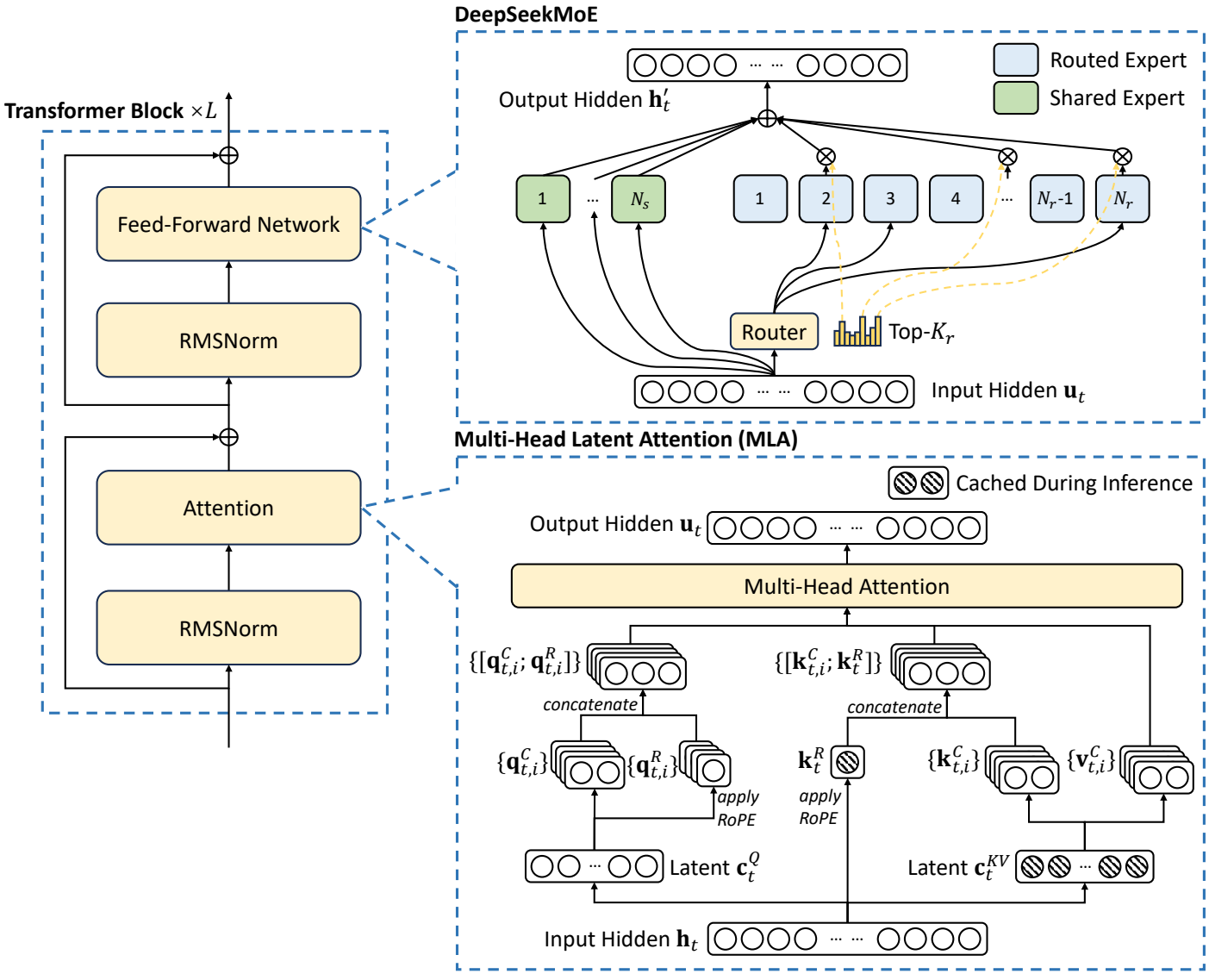


Figure: Our notation: $h \to e$, $c^Q \to e^Q$, $c^{KV} \to e^{KV}$, $o \to \hat{v}$, $u \to \hat{e}$

DeepSeek-V2 (Liu et al., 2024a)

- Multi-head Latent Attention
  - Map token emb e into an intermediate *latent* emb $e^{KV} = W^{KV} e$ of much lower dimension. Next, compute keys and values $k = W^K e^{KV}$, $v = W^V e^{KV}$ from this smaller dimensional rep.
  - Reduces size of the KV cache during inference, as only e^{KV} needs to be stored; leading to memory savings.
  - Weight decay can induce low-rank attention layers, see e.g., Kobayashi et al. (2024); so this architectural choice has some principled justification.
  - Same for the query, i.e., $e^Q = W^{Q'} e$, $q = W^Q e^Q$.
  - Compute MHA as usual.
  - Some linear maps become redundant, e.g., W^K and W^Q can be merged; also W^V and output projection W^O
- Decoupled Rotary Position Embedding (Bi et al., 2024)
  - Apply RoPE only to separate key-value projections

MLA + Decoupled RoPE

$$\mathbf{c}_t^Q = W^{DQ}\mathbf{h}_t, \tag{37}$$

$$[\mathbf{q}_{t,1}^C; \mathbf{q}_{t,2}^C; ...; \mathbf{q}_{t,n_h}^C] = \mathbf{q}_t^C = W^{UQ}\mathbf{c}_t^Q, \tag{38}$$

$$[\mathbf{q}_{t,1}^R; \mathbf{q}_{t,2}^R; ...; \mathbf{q}_{t,n_h}^R] = \mathbf{q}_t^R = \mathrm{RoPE}(W^{QR}\mathbf{c}_t^Q), \tag{39}$$

$$\mathbf{q}_{t,i} = [\mathbf{q}_{t,i}^C; \mathbf{q}_{t,i}^R], \tag{40}$$

$$\boxed{\mathbf{c}_t^{KV}} = W^{DKV}\mathbf{h}_t, \tag{41}$$

$$[\mathbf{k}_{t,1}^C; \mathbf{k}_{t,2}^C; ...; \mathbf{k}_{t,n_h}^C] = \mathbf{k}_t^C = W^{UK}\mathbf{c}_t^{KV}, \tag{42}$$

$$\boxed{\mathbf{k}_t^R} = \mathrm{RoPE}(W^{KR}\mathbf{h}_t), \tag{43}$$

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_t^R], \tag{44}$$

$$[\mathbf{v}_{t,1}^C; \mathbf{v}_{t,2}^C; ...; \mathbf{v}_{t,n_h}^C] = \mathbf{v}_t^C = W^{UV}\mathbf{c}_t^{KV}, \tag{45}$$

$$\mathbf{o}_{t,i} = \sum_{j=1}^{t} \mathrm{Softmax}_j\left(\frac{\mathbf{q}_{t,i}^T \mathbf{k}_{j,i}}{\sqrt{d_h + d_h^R}}\right)\mathbf{v}_{j,i}^C, \tag{46}$$

$$\mathbf{u}_t = W^O[\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; ...; \mathbf{o}_{t,n_h}], \tag{47}$$

where the boxed vectors in blue need to be cached for generation. During inference, the naive formula needs to recover $\mathbf{k}_t^C$ and $\mathbf{v}_t^C$ from $\mathbf{c}_t^{KV}$ for attention. Fortunately, due to the associative law of matrix multiplication, we can absorb W^{UK} into W^{UQ}, and W^{UV} into W^O. Therefore, we do not need to compute keys and values out for each query. Through this optimization, we avoid the computational overhead for recomputing $\mathbf{k}_t^C$ and $\mathbf{v}_t^C$ during inference.

Figure: Our notation: $h \to e$, $c^Q \to e^Q$, $c^{KV} \to e^{KV}$, $o \to \hat{v}$, $u \to \hat{e}$

Mixtures of Experts in a LLM

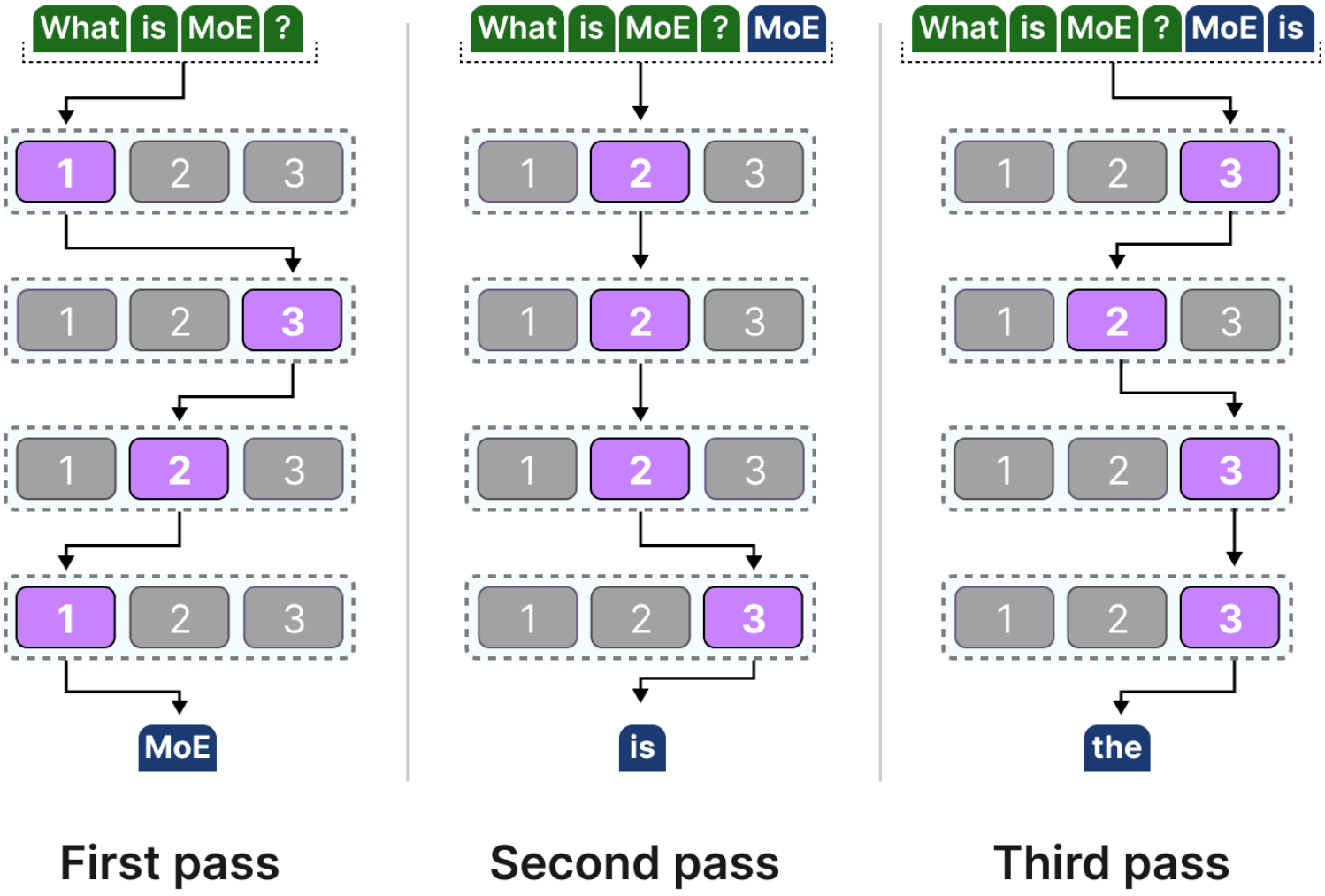


Figure: Source: A Visual Guide to Mixture of Experts (MoE). The blocks represent FFNs.

"Specialization": Tokens Routed in LLMs

Expert specialization	Expert position	Routed tokens
Punctuation	Layer 2	, , , , , , , , ,- , , , , ,).)
	Layer 6	, , , , , : . : , & , & & ? &-,,? ,,, .
Conjunctions and articles	Layer 3	The the the the the the the the The...
	Layer 6	a and and and and and and or and ...
Verbs	Layer 1	died falling identified fell closed left posted lost felt left said read miss place struggling falling signed died...
Visual descriptions *color, spatial position*	Layer 0	her over her know dark upper dark outer center upper blue inner yellow raw mama bright bright over open your dark blue
Counting and numbers *written and numerical forms*	Layer 1	after 37 19. 6. 27 I I Seven 25 4, 54 I two dead we Some 2012 who we few lower

Figure: Source: A Visual Guide to Mixture of Experts (MoE)

DeepSeek-V3 MoE

- Mixture of Experts (MoE): Shared and routed experts for efficiency (Dai et al., 2024). Compute the FFN output h' as

$$\tilde{e}_t = \hat{e}_t + \sum_{i=1}^{N_s} \phi_i^{(s)}(\hat{e}_t) + \sum_{i=1}^{N_r} g_i \phi_i^{(r)}(\hat{e}_t),$$

$$g_i = \begin{cases} s_i, & s_i \in \mathrm{TopK}(\{s_j \mid 1 \leqslant j \leqslant N_r\}, K), \\ 0, & \text{otherwise}, \end{cases}$$

$$s_i' = \mathrm{Sigmoid}\left(\hat{e}_t^\top \mu_i\right), \quad s_i = s_i' / (\sum_{j=1}^{N_r} s_j')$$

where

  - $\hat{e}_t$ is the FFN input of token t (after attention, residual update, and normalization)
  - $\phi_i^{(s)}(\cdot)$, $i \in [N_s]$ and $\phi_i^{(r)}(\cdot)$, $i \in [N_r]$ denote the i-th shared and routed experts (FFNs), resp;
  - K denotes the number of activated routed experts; (TopK idea from Shazeer et al. (2017))
  - g_i is the gate value for the i-th expert; s_i is token-to-expert affinity;
  - μ_i is a learnable "centroid" for the i-th routed expert (softmax routing idea from Jordan and Jacobs (1994)).

DeepSeek-V3

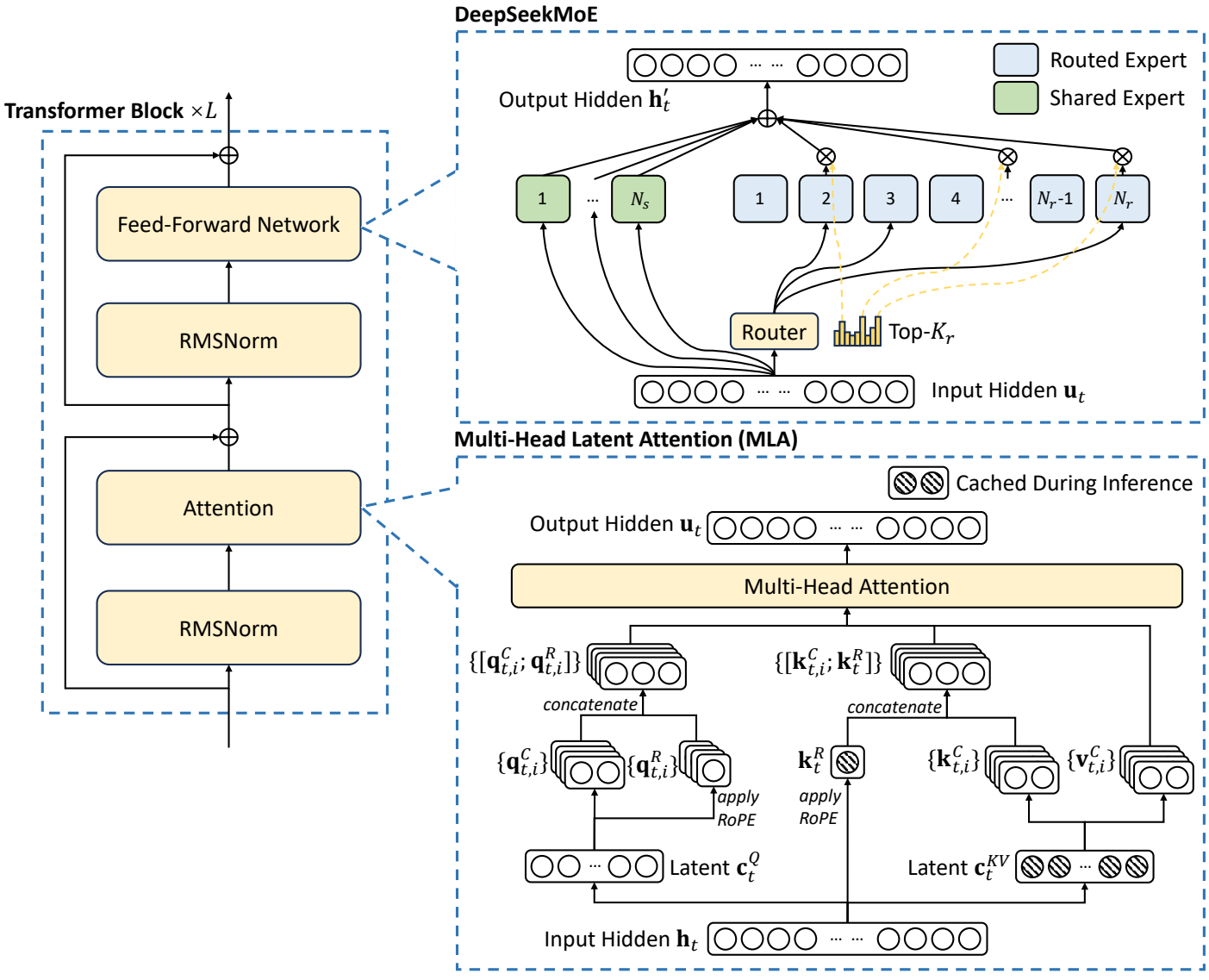


Figure: Our notation: $h \to e$, $c^Q \to e^Q$, $c^{KV} \to e^{KV}$, $o \to \hat{v}$, $u \to \hat{e}$

DeepSeek-V3 MoE

- Auxiliary-Loss-Free Load Balancing (Wang et al., 2024).
  - Add a constant c_i to the affinities when determining which experts to choose:

    $$s_i + c_i \in \text{Topk}\left(\{s_j + c_j \mid 1 \leqslant j \leqslant N_r\}, K\right).$$

    These values are constant across tokens.
  - Update them heuristically during training to balance loads.

Loss-based balancing in MoE

- Loss-based balancing (Fedus et al., 2022), with a small weight: For a batch of B tokens, define auxiliary loss $\mathcal{L}_{\text{Balance}} = \alpha N \sum_{i=1}^{N} f_i \bar{s}_i / K$, where

$$f_i = \frac{1}{B} \sum_{b=1}^{B} I(\text{Token } b \text{ selects expert } i), \quad \bar{s}_i = \frac{1}{B} \sum_{b=1}^{B} s_{i,b}.$$

  Here:

  - $N = N_r + N_s$ is the total number of experts.
  - K is the number of experts selected for each token.
  - $s_{i,b}$ is the routing score of expert i for token t.
  - f_i represents the fraction of tokens routed to expert i.
  - $\bar{s}_i$ denotes the average gating scores of expert i.
  - α is a hyper-parameter controlling the strength of the auxiliary loss.

Intuition for loss-based balancing (Fedus et al., 2022)

- Consider $B = 1$. Then $\bar{s} = (\bar{s}_1, \ldots, \bar{s}_N)^\top$ and $f = (f_1, \ldots, f_N)^\top$, where

$$f_i = I(i \in \arg\max(\bar{s}))/|\arg\max(\bar{s})|.$$

- Since $\bar{s}^\top f = \max_i(\bar{s}_i)$, the loss is minimized for a uniform distribution $\bar{s} = (1/N, \ldots, 1/N)^\top$
- Now suppose p is parametrized by parameters w. While max is not differentiable, we can still heuristically pick $i \in \arg\max(\bar{s})$ and use $\nabla_w \bar{s}_i$ as a gradient in backpropagation.
- Intuitively, this loss promotes balance, since f_i is correlated with $\bar{s}_i$ across tokens: larger average scores (across tokens) for an expert correspond to larger selection frequencies (across tokens) of that specific expert.

DeepSeek-V3

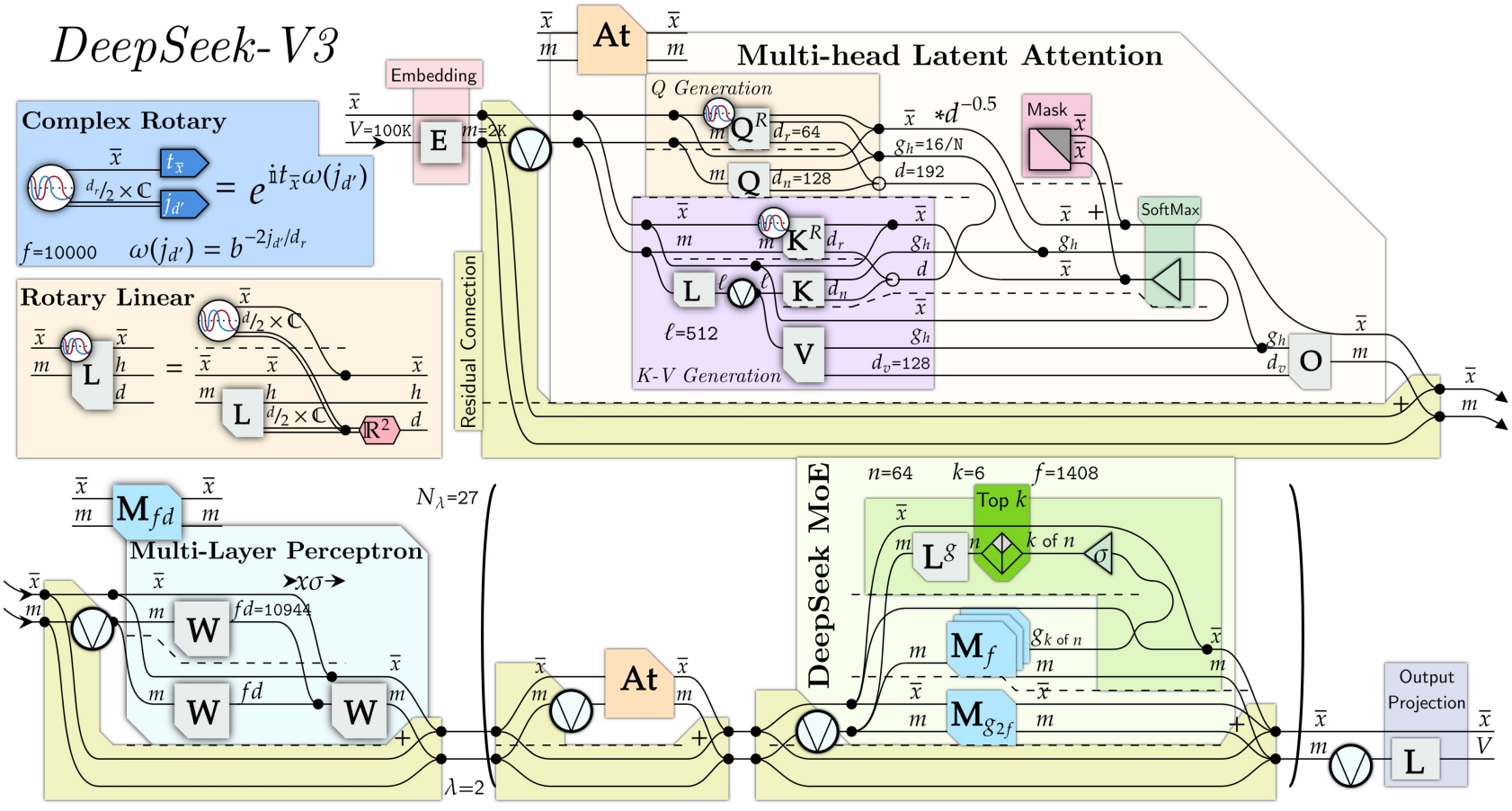


Figure: Via X (suspended account)

DeepSeek-V3 Long context extension

- Long context extension: YaRN (Peng et al., 2023).
- RoPE location embedding for token j in context of length T:

$$f(e, j, \vec{\theta}) = \begin{pmatrix} R(\theta_1, j) & 0 & \cdots & 0 \\ 0 & R(\theta_2, j) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & R(\theta_{d/2}, j) \end{pmatrix} \cdot W \cdot e,$$

  where $\vec{\theta} = (\theta_1, \ldots, \theta_{d/2})^\top$, $\theta_m = F^{-2m/d}$, $R(\theta_m, j)$ is 2x2 mx of rotn by $2\pi\theta_m j$, and F is a large number such as 10,000.
- Intuitive length interpolation for token $\tilde{j} \in [\tilde{T}]$ to new context length $\tilde{T} > T$:

$$\tilde{f}(\cdot, \tilde{j}, \vec{\theta}) = f(\cdot, \tilde{j}/s, \vec{\theta}) = f(\cdot, \tilde{j}, \vec{\theta}/s)$$

  where $s = \frac{\tilde{T}}{T}$ is a scale variable. Preserve relative token position.
  - Fine-tune this model on a relatively smaller dataset.

Motivation for YaRN (Peng et al., 2023)

- Let
$$\lambda_m = \frac{1}{\theta_m}$$
be the wavelength: number of tokens needed such that the RoPE embedding at dimension m performs a full rotation, i.e., $R(\theta_m, j + \lambda_m) = R(\theta_m, j)$
- "Given a context size T, there are some dimensions m where the wavelength is longer than the maximum context length: $\lambda_m > T$." Equivalent to $m > d/2 \cdot \log T / \log F$, so high-index/low-freq dims
- "In such cases, absolute positional information remains intact."; At this coord m, $R(\theta_m, j)$, $j \in [T]$, are all distinct

Motivation for YaRN (Peng et al., 2023)

- "Moreover, when we stretch the RoPE dimensions by a scale s, all tokens become closer to each other, as the dot product of two vectors rotated by a lesser amount is bigger."
- $R(\theta_m, \tilde{j}/s)$ is rotation by $2\pi\theta_m\tilde{j}/s$, which for any fixed $\tilde{j}$, it is smaller by a factor of $1/s$ than rotation of $R(\theta_m, \tilde{j})$; so inner product between specific components at coords $\tilde{j}_1$, $\tilde{j}_2$ at a specified distance $\tilde{j}_2 - \tilde{j}_1$ gets larger, despite the distance being fixed between context lengths
- "This scaling severely impairs a LLM's ability to understand small and local relationships between its internal embeddings."

Long context extension: YaRN (Peng et al., 2023)

- "Given these two observations, we choose not to interpolate the higher frequency dimensions, while interpolating the lower frequency dimensions."
- High-freq/low wavelength: better encodes rel pos (do not change rotn angle); Low-freq/high wavelength: better encodes abs pos (can change rotn angle);
- Define ratio $r_m = \frac{T}{\lambda_m} = T\theta_m$ of context length to the wavelength.
- Define linear interpolant:

$$\gamma(r) = \begin{cases} 0, & r < \alpha \\ 1, & r > \beta \\ \frac{r-\alpha}{\beta-\alpha}, & \text{else} \end{cases}$$

  for some hyperparameters α, β.
- YaRN interpolation: $f(\cdot, \tilde{j}, \vec{\theta}) = \tilde{f}(\cdot, \tilde{j}, h(\vec{\theta}))$ where for each m,

$$h(\theta_m) = (1 - \gamma(r_m))\frac{\theta_m}{s} + \gamma(r_m)\theta_m.$$

Additional Steps

- Scale everything dynamically by using current context length during inference
- Divide emb by $\ln(s)/10 + 1$

Illustrative Results

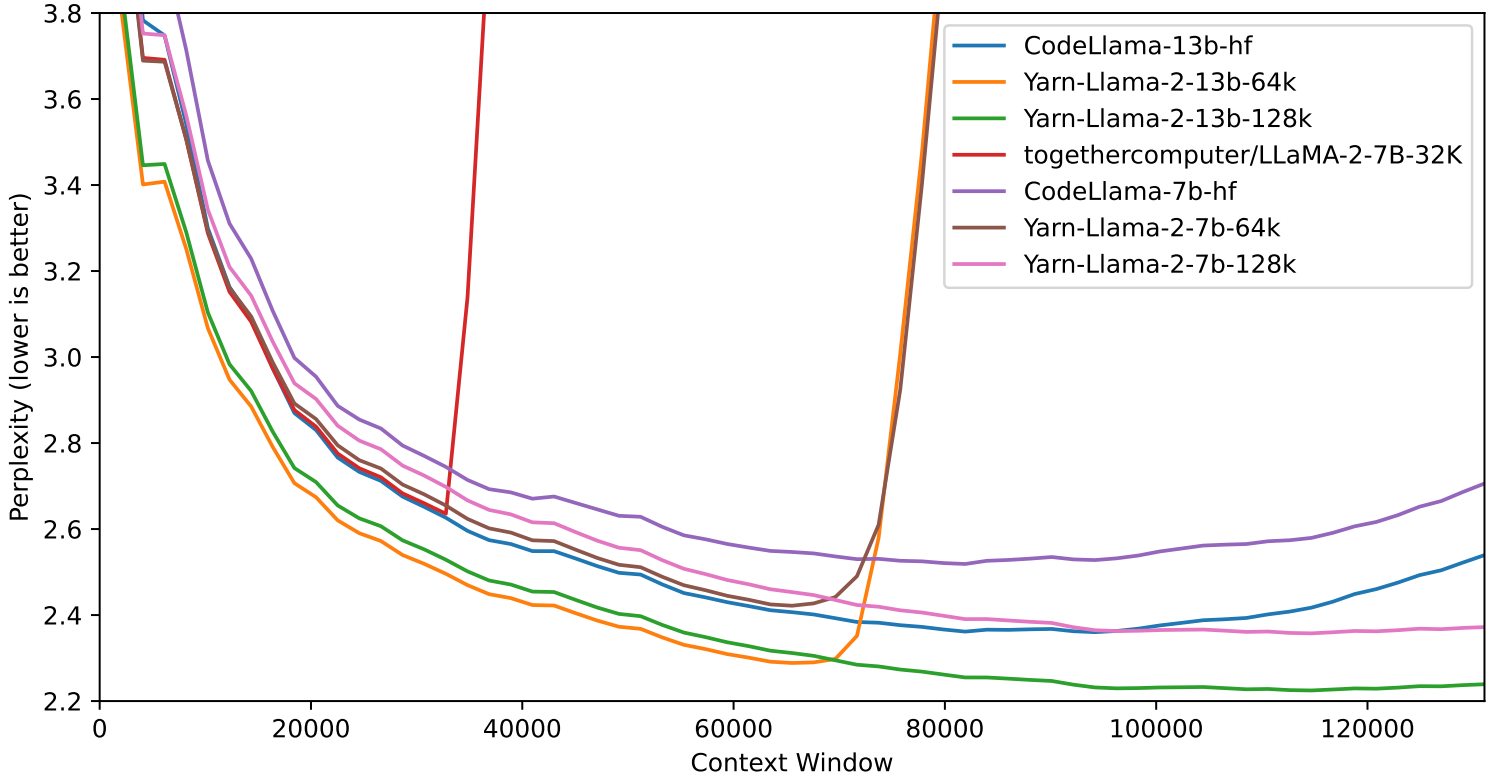


Figure: Sliding window perplexity (S $= 256$) of ten 128k Proof-pile documents truncated to evaluation context window size (Peng et al., 2023). Original models have $T = 4096$ and are extended to target values of $\tilde{T}$ (e.g., 64k). For Yarn, this involves taking $s = \tilde{T}/T$, e.g., $s = 16$. CodeLlama uses something like the intuitive interpolation.

Discussion of long-context extension

- Do we find YaRN compelling?
- Any ideas on how to improve it?

Table of Contents

LLM360

- LLM360 (Liu et al., 2023, 2025) is a fully open LLM: open weights, data, code, checkpoints, ...
- LLM360 K2 Diamond 65B: comparable to LLaMA2-70B, while requiring fewer FLOPs and tokens
- Similar arch to Llama.

References

J. Ainslie, J. Lee-Thorp, M. de Jong, Y. Zemlyanskiy, F. Lebron, and S. Sanghai. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901, 2023.

X. Bi, D. Chen, G. Chen, S. Chen, D. Dai, C. Deng, H. Ding, K. Dong, Q. Du, Z. Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

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.

R. Child, S. Gray, A. Radford, and I. Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

A. Choudhury. A simple approximation to the area under standard normal curve. *Mathematics and Statistics*, 2(3):147–149, 2014.

D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, et al. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *arXiv preprint arXiv:2401.06066*, 2024.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

References

W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

M. I. Jordan and R. A. Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

S. Kobayashi, Y. Akram, and J. Von Oswald. Weight decay induces low-rank attention layers. *arXiv preprint arXiv:2410.23819*, 2024.

A. Liu, B. Feng, B. Wang, B. Wang, B. Liu, C. Zhao, C. Dengr, C. Ruan, D. Dai, D. Guo, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024a.

A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024b.

Z. Liu, A. Qiao, W. Neiswanger, H. Wang, B. Tan, T. Tao, J. Li, Y. Wang, S. Sun, O. Pangarkar, et al. Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*, 2023.

Z. Liu, B. Tan, H. Wang, W. Neiswanger, T. Tao, H. Li, F. Koto, Y. Wang, S. Sun, O. Pangarkar, et al. Llm360 k2: Building a 65b 360-open-source large language model from scratch. *arXiv preprint arXiv:2501.07124*, 2025.

OpenAI. Gpt-4 technical report, 2023.

References

B. Peng, J. Quesnelle, H. Fan, and E. Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training, 2018.

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

N. Shazeer. Gelu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, and Y. Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample. Llama: Open and efficient foundation language models, 2023a.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

L. Wang, H. Gao, C. Zhao, X. Sun, and D. Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*, 2024.

B. Zhang and R. Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32, 2019.