

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

January 30, 2025



# Plan

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

# Table of Contents

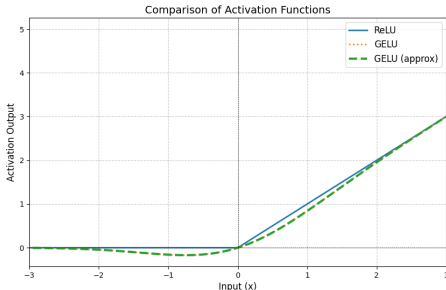
GPT

Llama

DeepSeek

# 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  $\Phi$  is normal cdf, or approximate  $x \mapsto 0.5x \left(1 + \tanh \left(\sqrt{\frac{2}{\pi}} (x + 0.044715x^3)\right)\right)$  (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:  $d/H = 128$

# GPT-3 "on a Napkin"



Figure: See [source](#) for a higher resolution.

# Table of Contents

GPT

Llama

DeepSeek

# LLama Series

- ▶ LLama 1 (Touvron et al., 2023a):
  - ▶ RMSNorm pre-normalization (Zhang and Sennrich, 2019).
  - ▶ FFN layer: SwiGLU (Shazeer, 2020):
$$x \mapsto \text{swish}(Wx + b) \odot (Vx + c), \text{ where } \text{swish}(z) = z / (1 + \exp(-z))$$
and  $W, V, b, c$  are learnable
  - ▶ 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



# Table of Contents

GPT

Llama

DeepSeek

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

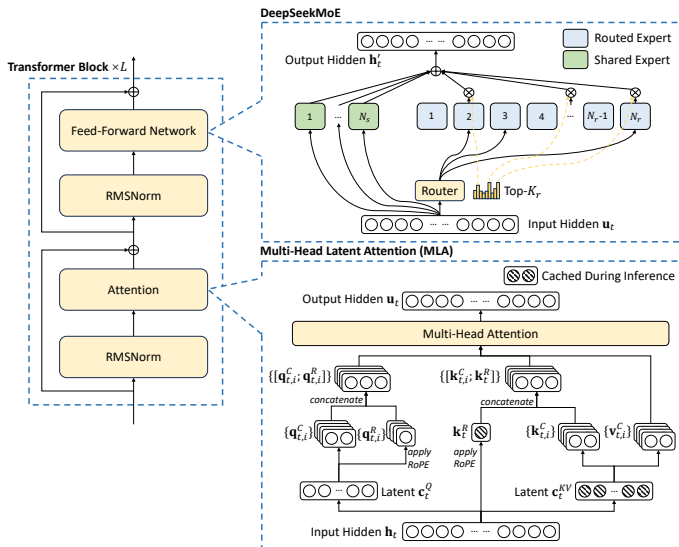


Figure: Our notation:  $h \rightarrow e$ ,  $c^Q \rightarrow e^Q$ ,  $c^{KV} \rightarrow e^{KV}$ ,  $o \rightarrow \hat{v}$ ,  $u \rightarrow e'$

# 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

$$\mathbf{c}_t^Q = W^{DQ} \mathbf{h}_t, \quad (37)$$

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

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

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

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

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

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

$$\mathbf{k}_{t,i} = [\mathbf{k}_{t,i}^C; \mathbf{k}_{t,i}^R], \quad (44)$$

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

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

$$\mathbf{u}_t = W^O [\mathbf{o}_{t,1}; \mathbf{o}_{t,2}; \dots; \mathbf{o}_{t,n_h}], \quad (47)$$

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:  $h \rightarrow e$ ,  $c^Q \rightarrow e^Q$ ,  $c^{KV} \rightarrow e^{KV}$ ,  $o \rightarrow \hat{v}$ ,  $u \rightarrow e'$

# Mixtures of Experts in a LLM

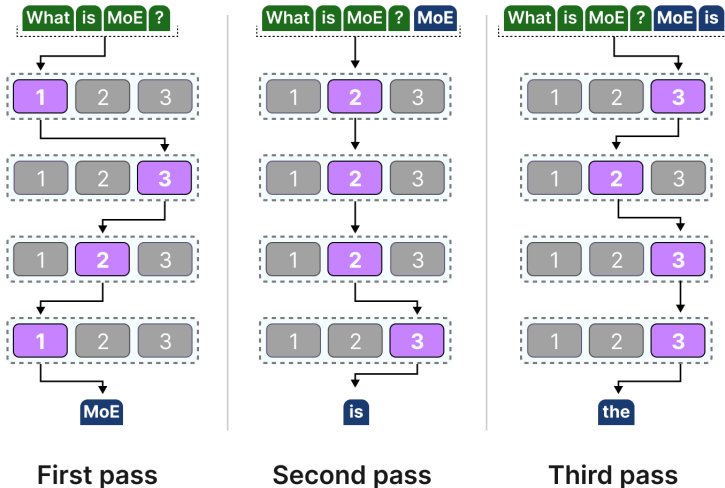


Figure: Source: [A Visual Guide to Mixture of Experts \(MoE\)](#). The blocks represent FFNs.

# "Specialization": Tokens Routed in LLMs

Expert specialization	Expert position	Routed tokens
<b>Punctuation</b>	Layer 2	, , , , , , , , - , , , , , . )
	Layer 6	, , , , , : : : , & , & & ? & - , ? , , , .
<b>Conjunctions and articles</b>	Layer 3	The the the the the the the the The...
	Layer 6	a and and and and and and and or and ...
<b>Verbs</b>	Layer 1	died falling identified fell closed left posted lost felt left said read miss place struggling falling signed died...
<b>Visual descriptions</b> <i>color, spatial position</i>	Layer 0	her over her know dark upper dark outer center upper blue inner yellow raw mama bright bright over open your dark blue
<b>Counting and numbers</b> <i>written and numerical forms</i>	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

$$h' = e'_t + \sum_{i=1}^{N_s} \phi_i^{(s)}(e'_t) + \sum_{i=1}^{N_r} g_i \phi_i^{(r)}(e'_t),$$
$$g_i = \begin{cases} s_i, & s_i \in \text{Topk}(\{s_j \mid 1 \leq j \leq N_r\}, K_r), \\ 0, & \text{otherwise,} \end{cases}$$
$$s'_i = \text{Sigmoid}\left(e'^{\top}_t f_i\right), \quad s_i = s'_i / \left(\sum_{j=1}^{N_r} s'_j\right)$$

where

- $\phi_i^{(s)}(\cdot)$ ,  $i \in [N_s]$  and  $\phi_i^{(r)}(\cdot)$ ,  $i \in [N_r]$  denote the  $i$ -th shared and routed experts, resp;
- $K_r$  denotes the number of activated routed experts;
- $g_i$  is the gate value for the  $i$ -th expert;  $s_i$  is token-to-expert affinity;
- $f_i$  is the mean over tokens of the activations of the  $i$ -th routed expert.

# DeepSeek-V3

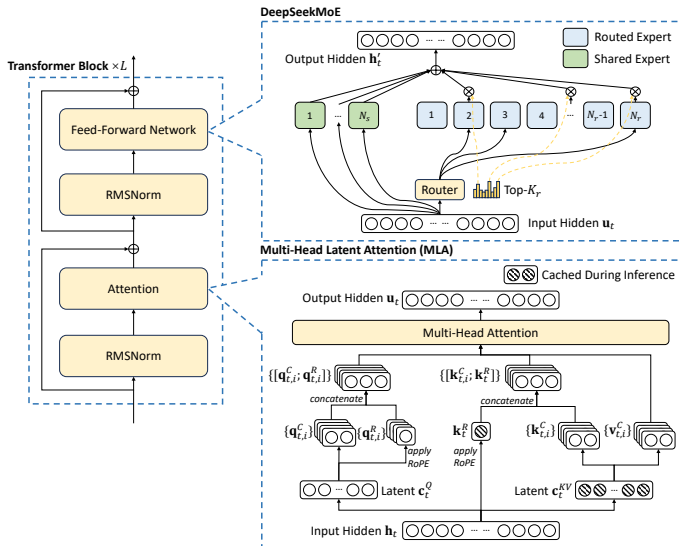


Figure: Our notation:  $h \rightarrow e$ ,  $c^Q \rightarrow e^Q$ ,  $c^{KV} \rightarrow e^{KV}$ ,  $o \rightarrow \hat{v}$ ,  $u \rightarrow e'$



# 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 as in  $s_i + c_i \in \text{Topk}(\{s_j + c_j \mid 1 \leq j \leq N_r\}, K_r)$ . These values are constant across tokens.
  - ▶ Update them heuristically during training to balance loads.

# Loss-based balancing in MoE

- Loss-based balancing (Lepikhin et al., 2020; Fedus et al., 2022), with a small weight: For a sequence of length  $T$ , define auxiliary loss  $\mathcal{L}_{\text{Balance}} = \alpha \sum_{i=1}^N f_i P_i$ , where

$$f_i = \frac{N}{KT} \sum_{t=1}^T I(\text{Token } t \text{ selects Expert } i), \quad P_i = \frac{1}{T} \sum_{t=1}^T s_{i,t}.$$

Here:

- $N$  is the total number of experts.
- $K$  is the number of experts selected for each token.
- $s_{i,t}$  is the routing score of Expert  $i$  for Token  $t$ .
- $f_i$  represents the fraction of tokens routed to Expert  $i$ .
- $P_i$  denotes the average gating scores of Expert  $i$ .
- $\alpha$  is a hyper-parameter controlling the strength of the auxiliary loss.

This loss promotes balance, if  $f_i$  is correlated with  $P_i$  across tokens;

- larger average scores (across tokens) for an expert correspond to larger selection frequencies (across tokens) of that specific expert.

# DeepSeek-V3

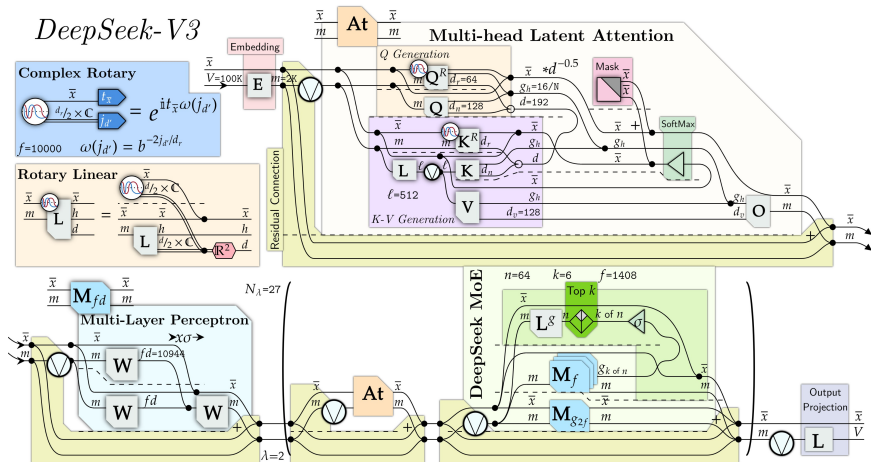


Figure: Via X (suspended account)

# DeepSeek-V3

- ▶ Long context extension: YaRN ([Peng et al., 2023](#)).

# 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.
- S. Kobayashi, Y. Akram, and J. Von Oswald. Weight decay induces low-rank attention layers. *arXiv preprint arXiv:2410.23819*, 2024.
- D. Lepikhin, H. Lee, Y. Xu, D. Chen, O. Firat, Y. Huang, M. Krikun, N. Shazeer, and Z. Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.
- 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.
- OpenAI. Gpt-4 technical report, 2023.
- 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.

# References

- 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.
- 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.