Attention Is All You Need But You Don't Need All Of It For Inference of Large Language Models

Georgy Tyukin [*1] **Gbetondji J-S Dovonon** [1] **Jean Kaddour** [1] **Pasquale Minervini** [2]

Abstract

The inference demand for LLMs has skyrocketed in recent months, and serving models with low latencies remains challenging due to the quadratic input length complexity of the attention layers. In this work, we investigate the effect of dropping MLP and attention layers at inference time on the performance of Llama-v2 models. We find that dropping deeper attention layers only marginally decreases performance but leads to the best speedups alongside dropping entire layers. For example, removing 33% of attention layers in a 13B Llama2 model results in a 0.9% drop in average performance over the OpenLLM benchmark. We also observe that skipping layers except the latter layers reduces performances for more layers skipped, except for skipping the attention layers.

1. Introduction

The ubiquitous deployment of Large Language Models (LLMs) results in ever-growing amounts of compute spent on inference (Patterson et al., 2021; Chen et al., 2023; Kaddour et al., 2023a; Xia et al., 2024; Reid et al., 2024). Further, serving models with low latencies remains challenging because contemporary Transformer architectures employ the self-attention mechanism with quadratic input complexity (Touvron et al., 2023b; Jiang et al., 2023; Bi et al., 2024).

In this work, we delve deeper into the concept of layer skipping (Fan et al., 2019; Wang et al., 2022a) to reduce the computation on superfluous LLM components. Our findings demonstrate that pruning deeper attention layers does not significantly affect performance. When applied to Llama-v2 (Touvron et al., 2023b), we maintain good performance on the OpenLLM (ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022)) benchmarks (Beeching et al., 2023), recording only minimal performance deviations compared to the full model.

[*]Equal contribution [1]University College London, UK [2]University of Edinburgh, UK. Correspondence to: Georgy Tyukin <tyukinegor@gmail.com>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

2. Method

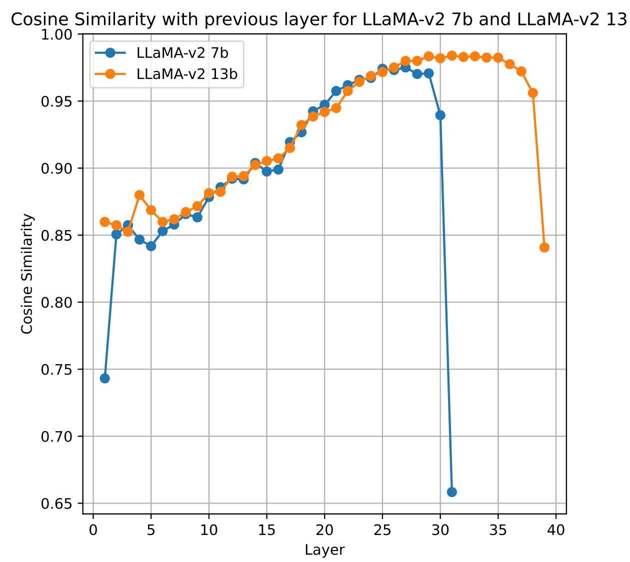


Figure 1. Cosine similarity of Llama-v2 layers with the previous layer: We observe that the deeper the layer, the more its features are similar to the previous layer except for the very last layer.

2.1. Layer skipping

Consider a Transformer model $\mathcal{M}$ with L layers, each consisting of an attention sub-layer followed by a multi-layer perceptron (MLP) sub-layer. We denote each layer as $\mathcal{M}_i = (\text{Attention}_i, \text{MLP}_i)$ for $i \in \{1, 2, \ldots, L\}$.

To compare the performance of Transformer models when skipping specific sub-layers, we create two variants of the model:

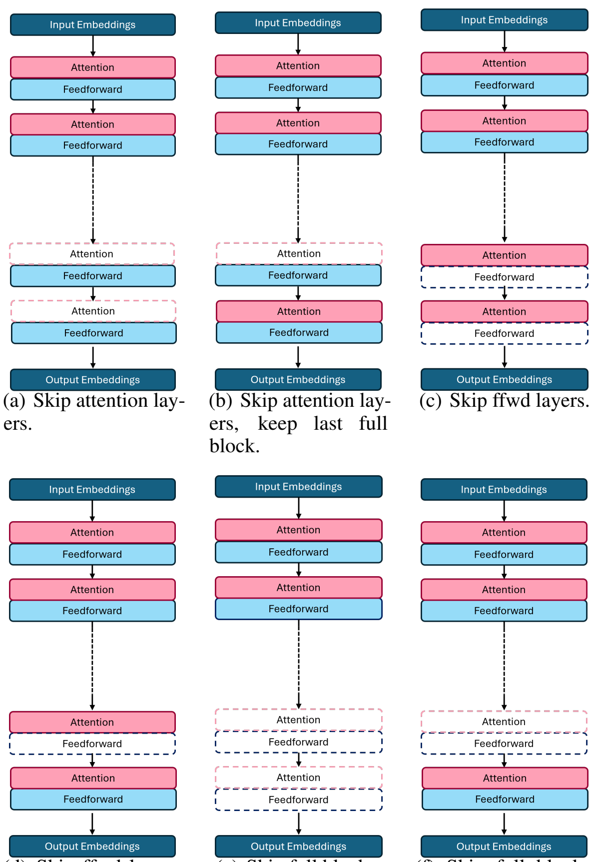
1. **Skipping MLP Layers:** We construct a model $\mathcal{M}_{\text{skip MLP}}$

by skipping the MLP sub-layer from the last k layers. The resulting model is $\mathcal{M}_{\text{skip MLP}} = \{(\text{Attention}_i, \text{MLP}_i) \mid i \in \{1, 2, \ldots, L-k\}\} \cup \{(\text{Attention}_i, \emptyset) \mid i \in \{L-k+1, \ldots, L\}\}$.

2. **Skipping Attention Layers:** We construct a model $\mathcal{M}_{\text{skip Attention}}$ by skipping the attention sub-layer from the last k layers. The resulting model is $\mathcal{M}_{\text{skip Attention}} = \{(\text{Attention}_i, \text{MLP}_i) \mid i \in \{1, 2, \ldots, L-k\}\} \cup \{(\emptyset, \text{MLP}_i) \mid i \in \{L-k+1, \ldots, L\}\}$.

3. **Skipping Transformer Blocks:** We construct a model $\mathcal{M}_{\text{skip Attention}}$ by skipping the entire last k layers. The resulting model is $\mathcal{M}_{\text{skip Block}} = \{(\text{Attention}_i, \text{MLP}_i) \mid i \in \{1, 2, \ldots, L-k\}\} \cup \{(\emptyset) \mid i \in \{L-k+1, \ldots, L\}\}$.

We then evaluate the performance of these modified models on the OpenLLM benchmark (Beeching et al., 2023), comparing metrics such as accuracy, computational efficiency, and memory usage. This comparison helps in understanding the individual contributions of the attention and MLP sub-layers to the overall performance of the Transformer model.



(a) Skip attention layers. (b) Skip attention layers, keep last full block. (c) Skip ffwd layers.

(d) Skip ffwd layers, keep last full block. (e) Skip full blocks. (f) Skip full blocks, keep last full block.

Figure 2. **Skip mechanisms** for skipping single layers and entire Transformer blocks (ffwd and attention layers) during inference.

2.2. Motivation: Are Deeper Layers More Redundant?

In Transformer models, the last layers have been shown to contribute less information than earlier layers, making it possible to drop those layers at a minimal performance cost (Fan et al., 2019; Zhang & He, 2020; Wang et al., 2022a; Schuster et al., 2022; Kaddour et al., 2023b; Belrose et al., 2023).

To verify this, we experiment with removing either the attention sublayers or the MLP sublayers. Figure 1 shows the cosine similarities between a layer's features and the previous layer showing that deeper layers have a lower impact on the features than earlier layers. One notable exception to this trend is that the last layer for both Llama-v2 7B and 13B has the lowest cosine similarity with the previous layer.

Previous analysis of the attention mechanism has shown that they can converge to the same value due to attention collapse (Zhai et al., 2023) and token features that also converge to the same value due to over-smoothing (Wang et al., 2022b; Dovonon et al., 2024) or rank collapse (Dong et al., 2023), with solutions to these issues typically improving performance (Ali et al., 2023; Choi et al., 2024).

3. Results

Experimental Setup For all experiments, we use either Llama-v2-7B or Llama-v2-13B (Touvron et al., 2023a;b), two LLMs trained on trillions of publically available tokens. We experiment with keeping 66%, 75%, 90% and 100% of the network and report the corresponding results in Table 1. We also experiment with removing attention sublayers in Table 2, MLP sublayers in Table 3, and a varying number of layers similar to Table 1 but keeping the last layer in Table 4.

3.1. Chopping Layers

Table 1. Llama-v2 skipping full layer

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	35.2	46.8	46.2	40.3	**42.1**
7B-75%	38.3	53.0	45.1	45.9	**45.6**
7B-90%	47.7	69.3	39.6	46.4	**50.8**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	37.8	46.8	45.3	51.8	**45.4**
13B-75%	40.9	53.6	42.5	53.2	**47.6**
13B-90%	51.3	71.3	37.1	54.8	**53.6**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

On all datasets except TruthfulQA, performance drops which is expected. It had already been observed that larger language models are less truthful (Lin et al., 2022), but we now also observe that reducing the size of already trained models can also make them more truthful. The observation still holds when the last layer is preserved. Skipping

Table 2. Llama-v2 skipping attention sublayers

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	51.2	77.0	42.2	39.4	**52.5**
7B-75%	52.5	78.3	42.3	41.4	**53.6**
7B-90%	52.8	78.9	40.0	44.0	**53.9**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	55.6	80.1	40.1	51.3	**56.8**
13B-75%	55.9	79.7	39.9	52.1	**56.9**
13B-90%	57.0	81.3	38.2	54.8	**57.8**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

Table 3. Llama-v2 skipping ffwd sublayers

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	35.1	52.5	42.2	43.9	**43.4**
7B-75%	40.4	60.3	39.2	46.3	**46.6**
7B-90%	48.5	71.4	38.0	46.1	**51.0**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	41.6	56.9	40.7	53.4	**48.2**
13B-75%	47.3	65.2	40.0	53.2	**51.4**
13B-90%	54.2	75.8	38.3	54.7	**55.8**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

attention layers only leads to better results with only a 0.9% decrease in performance when keeping 66% of the network compared to a 13.1% decrease in performance when dropping dropping the MLP layers only. This seems to indicate that MLP layers are more important than attention layers, at least in deeper parts of the network.

3.2. Last Layer Inclusion

Table 4. Llama-v2 skip full layers with last layer

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	32.0	45.8	46.9	40.7	**41.3**
7B-75%	34.5	49.4	45.9	38.3	**42.0**
7B-90%	46.5	73.1	41.8	41.4	**50.7**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	35.1	50.0	46.9	19.1	**37.8**
13B-75%	38.7	56.6	43.7	25.2	**41.1**
13B-90%	51.2	78.1	38.0	27.1	**47.9**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

Surprisingly, we notice that skipping layers except the latter layers reduces performances for more layers skipped, except for skipping the attention layers. This is even more exaggerated compared to just dropping layers, including the last one. The reason for this could be attributed to the (lack of) robustness of feedforward sublayers, as the last layer now has to process perturbed information from earlier layers. For future work, it would be interesting to see if these performance drops can be compensated by a small amount

Table 5. Llama-v2 skip attention sublayers with last layer

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	49.3	77.1	40.5	42.5	**52.4**
7B-75%	51.8	78.3	41.1	44.1	**53.8**
7B-90%	51.9	78.7	39.4	45.7	**53.9**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	56.8	82.1	38.0	50.3	**56.8**
13B-75%	57.5	82.1	37.0	51.4	**57.0**
13B-90%	58.9	82.4	36.6	54.5	**58.1**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

Table 6. Llama-v2 skip ffwd sublayers with last layer

Model	Performances				
	ARC	HellaSwag	TruthfulQA	MMLU	**Average**
7B-66%	32.0	45.8	46.9	39.4	**41.0**
7B-75%	34.5	49.4	45.9	40.2	**42.5**
7B-90%	46.5	73.1	41.8	40.2	**50.4**
7B-100%	53.1	78.6	38.8	46.6	**54.3**
13B-66%	35.1	50.0	46.9	20.4	**38.1**
13B-75%	38.7	56.6	43.7	33.6	**43.2**
13B-90%	51.2	78.1	38.0	34.4	**50.4**
13B-100%	59.6	82.1	36.9	55.4	**58.5**

of continued training; since model growing techniques for training seem to not suffer from instabilities (Kaddour et al., 2023b).

3.3. Compute-matched Comparison

To measure the efficiency of the networks we conducted a separate experiment, where we record the time it takes for the model to output a sequence of length 1, averaging over 1000 sequences. We conducted this experiment for both 50 and 100 length input sequences. We notice that full layer droppings do improve time costs the best, followed by attention sublayers, and then feedforward sublayers which do not impact the speed of processing a lot.

We report the time$\times 10^2$ (for clarity) it takes to predict 1 token for 1000 sequences as well as the percentage improvement. We show the results of this experiment for Llama 2 7B with 0%, 10%, 25%, 33% of layers skipped and we label these as 7B-100%, 7B-90%, 7B-75%, 7B-66% respectively.

Table 7. Llama-v2 time results, 50 length sequence, no last layer

Model	Full		Attention		ffwd	
	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)
7B-66%	31.35	32.96	36.72	21.47	43.51	6.95
7B-75%	35.48	24.12	39.46	15.61	42.88	8.30
7B-90%	43.31	7.38	42.93	8.19	44.17	5.53
7B-100%	46.76	0	-	-	-	-

Table 8. Llama-v2 time results, 50 length sequence, last layer included

Model	Full		Attention		ffwd	
	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)
7B-66%	31.78	32.04	36.92	21.04	41.31	11.66
7B-75%	34.98	25.19	40.24	13.94	42.62	8.85
7B-90%	40.92	12.49	42.43	9.26	43.51	6.95
7B-100%	46.76	0	-	-	-	-

Table 9. Llama-v2 time results, 100 length sequence, no last layer

Model	Full		Attention		ffwd	
	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)
7B-66%	32.36	32.58	38.97	18.18	43.08	10.25
7B-75%	36.58	23.79	41.27	14.02	44.13	8.06
7B-90%	43.65	9.06	44.62	7.04	46.30	3.54
7B-100%	48.00	0	-	-	-	-

Table 10. Llama-v2 time results, 100 length sequence, last layer included

Model	Full		Attention		ffwd	
	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)	Time(s) $\times 10^2$	(%)
7B-66%	32.05	33.23	38.52	19.75	42.66	11.13
7B-75%	36.41	24.15	41.00	14.58	43.92	8.50
7B-90%	43.28	9.83	44.27	7.77	45.20	5.83
7B-100%	48.00	0	-	-	-	-

4. Related Work

Early Exit during inference Early exit methods have also been proposed in other domains (Graves, 2017; Teerapittayanon et al., 2017) before getting adapted to autoregressive models (Elbayad et al., 2020; Schuster et al., 2022; Din et al., 2023; Elhoushi et al., 2024; Fan et al., 2024; Chen et al., 2024). The idea works by dynamically allocating compute based on the difficulty of the input sequence. Our method prunes the deepest layers and does not involve any level of adaptability. This is beneficial because it does not require the entire model to be loaded in memory. Dropping layers during inference has been done on BERT-like models in (Wang et al., 2022a; Sajjad et al., 2023). We apply a similar analysis to more recent LLMs and study the impact of skipping attention and/or MLP layers in more detail. Concurrent work to ours by Gromov et al. (2024) yields similar results by pruning deeper layers and applying fine-tuning on the pruned model.

Layer dropping/growing during training There are various works studying the dropping/growing layers dynamically during training (Fan et al., 2019; Gong et al., 2019; Kaddour et al., 2023b; Jiang et al., 2020; Liu et al., 2023). In contrast, this work focuses on dropping layers of an already pre-trained model in a way similar to Men et al. (2024).

Other Inference Speedup Methods Other works to speed up inference include compressing KV caches (Nawrot et al., 2024; Wu & Tu, 2024; Bi et al., 2024), speculative decoding (Chen et al., 2023), efficient memory management (Kwon et al., 2023), or subqudratic attention architectures (Fu et al., 2022; Peng et al., 2023; Gu & Dao, 2023), an overview has been provided by Kaddour et al. (2023a).

5. Conclusion

We investigated the effect of dropping the last layers from the 7B and 13B Llama2 models. We observe that dropping attention sublayers lead to much lower drops in performance than dropping the MLP sublayers, whether the last layer is included or not, while also leading to better inference speedups. For example, removing 33% of attention layers leads to an 18% speedup in a 13B Llama2 model at the cost of a 0.9% drop in average performance. This shows that massive improvements can be made over dropping entire layers from just dropping the attention sublayer.

References

Ali, A., Galanti, T., and Wolf, L. Centered self-attention layers, 2023.

Beeching, E., Fourrier, C., Habib, N., Han, S., Lambert, N., Rajani, N., Sanseviero, O., Tunstall, L., and Wolf, T. Open llm leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

Belrose, N., Furman, Z., Smith, L., Halawi, D., Ostrovsky, I., McKinney, L., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

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

Chen, C., Borgeaud, S., Irving, G., Lespiau, J., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318, 2023. doi: 10.48550/ARXIV.2302.01318. URL https://doi.org/10.48550/arXiv.2302.01318.

Chen, Y., Pan, X., Li, Y., Ding, B., and Zhou, J. Ee-llm: Large-scale training and inference of early-exit large language models with 3d parallelism, 2024.

Choi, J., Wi, H., Kim, J., Shin, Y., Lee, K., Trask, N., and Park, N. Graph convolutions enrich the self-attention in transformers!, 2024.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

Din, A. Y., Karidi, T., Choshen, L., and Geva, M. Jump to conclusions: Short-cutting transformers with linear transformations. *arXiv preprint arXiv:2303.09435*, 2023.

Dong, Y., Cordonnier, J.-B., and Loukas, A. Attention is not all you need: Pure attention loses rank doubly exponentially with depth, 2023.

Dovonon, G. J.-S., Bronstein, M. M., and Kusner, M. J. Setting the record straight on transformer oversmoothing, 2024.

Elbayad, M., Gu, J., Grave, E., and Auli, M. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJg7KhVKPH.

Elhoushi, M., Shrivastava, A., Liskovich, D., Hosmer, B., Wasti, B., Lai, L., Mahmoud, A., Acun, B., Agarwal, S., Roman, A., et al. Layer skip: Enabling early exit inference and self-speculative decoding. *arXiv preprint arXiv:2404.16710*, 2024.

Fan, A., Grave, E., and Joulin, A. Reducing transformer depth on demand with structured dropout, 2019.

Fan, S., Jiang, X., Li, X., Meng, X., Han, P., Shang, S., Sun, A., Wang, Y., and Wang, Z. Not all layers of llms are necessary during inference, 2024.

Fu, D. Y., Dao, T., Saab, K. K., Thomas, A. W., Rudra, A., and Ré, C. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Gong, L., He, D., Li, Z., Qin, T., Wang, L., and Liu, T. Efficient training of bert by progressively stacking. In *International conference on machine learning*, pp. 2337–2346. PMLR, 2019.

Graves, A. Adaptive computation time for recurrent neural networks, 2017.

Gromov, A., Tirumala, K., Shapourian, H., Glorioso, P., and Roberts, D. A. The unreasonable ineffectiveness of the deeper layers, 2024.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces, 2023.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Jiang, Y.-G., Cheng, C., Lin, H., and Fu, Y. Learning layer-skippable inference network. *IEEE Transactions on Image Processing*, 29:8747–8759, 2020. doi: 10.1109/TIP.2020.3018269.

Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., and McHardy, R. Challenges and applications of large language models. *CoRR*, abs/2307.10169, 2023a. doi: 10.48550/ARXIV.2307.10169. URL https://doi.org/10.48550/arXiv.2307.10169.

Kaddour, J., Key, O., Nawrot, P., Minervini, P., and Kusner, M. J. No train no gain: Revisiting efficient training algorithms for transformer-based language models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023b.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods, 2022.

Liu, Z., Wang, J., Dao, T., Zhou, T., Yuan, B., Song, Z., Shrivastava, A., Zhang, C., Tian, Y., Re, C., and Chen, B. Deja vu: Contextual sparsity for efficient LLMs at inference time. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22137–22176. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23am.html.

Men, X., Xu, M., Zhang, Q., Wang, B., Lin, H., Lu, Y., Han, X., and Chen, W. Shortgpt: Layers in large language models are more redundant than you expect, 2024. URL https://arxiv.org/abs/2403.03853.

Nawrot, P., Łańcucki, A., Chochowski, M., Tarjan, D., and Ponti, E. M. Dynamic memory compression: Retrofitting llms for accelerated inference, 2024.

Patterson, D. A., Gonzalez, J., Le, Q. V., Liang, C., Munguia, L., Rothchild, D., So, D. R., Texier, M., and Dean, J. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021. URL https://arxiv.org/abs/2104.10350.

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.

Reid, M., Savinov, N., Teplyashin, D., Lepikhin, D., Lillicrap, T., Alayrac, J.-b., Soricut, R., Lazaridou, A., Firat, O., Schrittwieser, J., et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Sajjad, H., Dalvi, F., Durrani, N., and Nakov, P. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429, jan 2023. doi: 10.1016/j.csl.2022.101429. URL https://doi.org/10.1016%2Fj.csl.2022.101429.

Schuster, T., Fisch, A., Gupta, J., Dehghani, M., Bahri, D., Tran, V., Tay, Y., and Metzler, D. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472, 2022.

Teerapittayanon, S., McDanel, B., and Kung, H. T. Branchynet: Fast inference via early exiting from deep neural networks, 2017.

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

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023b.

Wang, J., Chen, K., Chen, G., Shou, L., and McAuley, J. Skipbert: Efficient inference with shallow layer skipping. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7287–7301, 2022a.

Wang, P., Zheng, W., Chen, T., and Wang, Z. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022b. URL https://openreview.net/forum?id=O476oWmiNNp.

Wu, H. and Tu, K. Layer-condensed kv cache for efficient inference of large language models, 2024.

Xia, H., Yang, Z., Dong, Q., Wang, P., Li, Y., Ge, T., Liu, T., Li, W., and Sui, Z. Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding. *arXiv preprint arXiv:2401.07851*, 2024.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence?, 2019.

Zhai, S., Likhomanenko, T., Littwin, E., Busbridge, D., Ramapuram, J., Zhang, Y., Gu, J., and Susskind, J. Stabilizing transformer training by preventing attention entropy collapse, 2023.

Zhang, M. and He, Y. Accelerating training of transformer-based language models with progressive layer dropping. *Advances in neural information processing systems*, 33:14011–14023, 2020.