Transformer tricks: Removing weights for skipless transformers

Nils Graef
OpenMachine, South San Francisco, CA 94080, info@openmachine.ai

Abstract

He and Hofmann [1] detailed a skipless transformer without the V and P (post-attention projection) linear layers, which reduces the total number of weights. However, this scheme is only applicable to MHA (multi-head attention) [2], but not for MQA (multi-query attention) [3] and GQA (grouped-query attention) [4]. The latter schemes are used by many popular LLMs such as Llama 2, Mistral, Mixtral, PaLM, and Gemma [5, 6, 7, 8, 9]. Therefore, this micro-paper [10] proposes mathematically equivalent versions that are suitable for MQA and GQA. For example, removing Q and P from a skipless version of Mistral-7B would remove 15% of its weights (and thus reduce its compute and memory complexity). See [11, 12] for code and more transformer tricks.

1 Vanilla transformer without skip connections

He et al. [13] have shown how transformers without skip connections and normalization (see Figure 1(a)) can be trained successfully.

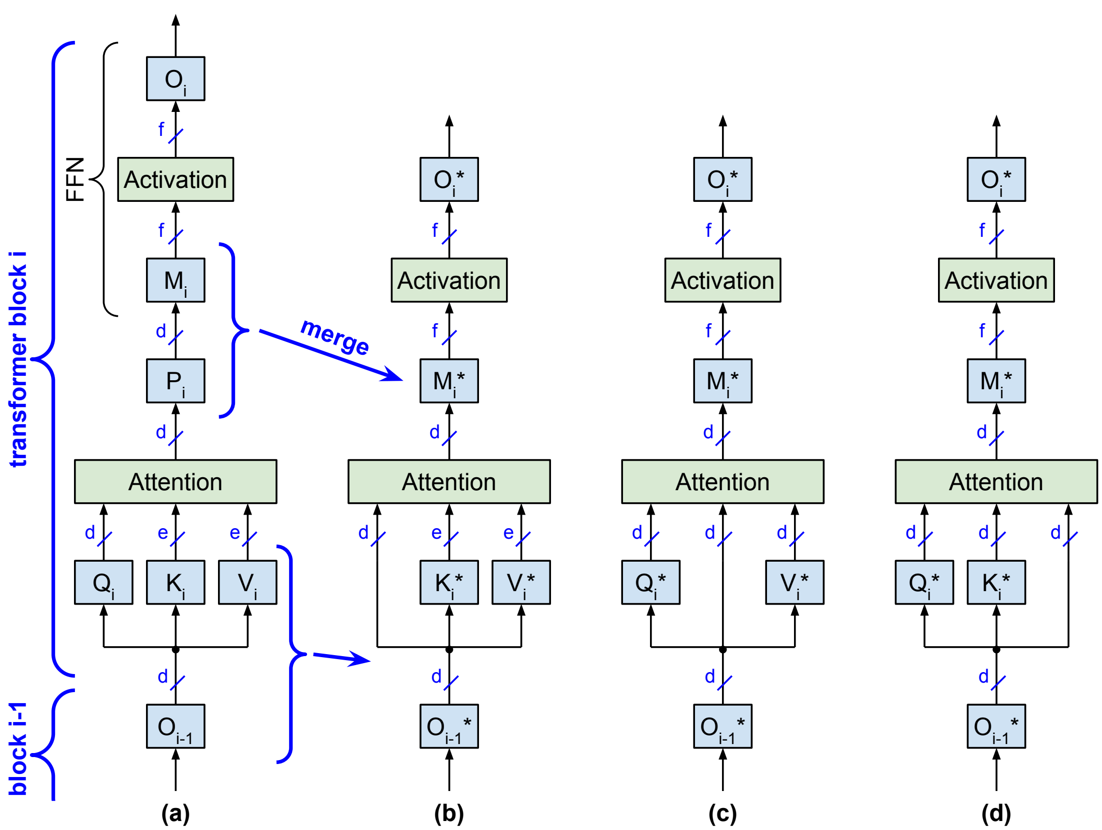

Figure 1: (a) Skipless vanilla transformer; equivalent versions with (b) Q and P merged into the FFN (feedforward network); (c) K and P merged into FFN; (d) V and P merged into FFN. $\mathbf{M}_i^*, \mathbf{Q}_i^*, \mathbf{K}_i^*, \mathbf{V}_i^*, \mathbf{O}_{i-1}^*$ are defined in table 1.

Removing skip connections and normalization allows us to merge linear layers in a mathematically identical way as shown in Figures 1(b) to (d). This reduces the number of weights without changing the functionality as follows:

- Figure 1(b) is mathematically identical to Figure 1(a) and eliminates $2d^2$ weights per transformer block by merging $\mathbf{P}_i$ into $\mathbf{M}_i^*$ and $\mathbf{Q}_i$ into $\mathbf{O}_{i-1}^*$.
- For MHA where $e = d$, Figures 1(c) and (d) are mathematically identical to Figure 1(a) and eliminate $2d^2$ weights per transformer block by merging $\mathbf{P}_i$ into $\mathbf{M}_i^*$ and $\mathbf{K}_i$ or $\mathbf{V}_i$ into $\mathbf{O}_{i-1}^*$.
- This requires that $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i$ are invertible (i.e. nonsingular). It is extremely rare that a square matrix with random values is not invertible [14] (which requires its determinant to be exactly 0).

Figure 1 uses the following dimensions and weight matrices, based on the type of attention:

- d: embedding dimension
- e: $e = d$ for MHA. For MQA, $e = d/n_{heads}$. And for GQA, $e = d \cdot n_{kv\_heads}/n_{heads}$.
- f: hidden dimension of the FFN. $f = 4d$ in the vanilla transformer; Shazeer [3] uses $f > 4d$. For models that use a GLU variant [15] (such as Llama and Mistral), the effective f' for the first linear layer M is $f' = 2f$, because the GLU variant uses two linear layers that are combined (via pointwise multiplication) with a non-linear activation function.
- $\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i, \mathbf{P}_i$: The weight matrices of the linear layers for query, keys, values, and the post-attention projection of transformer block i.
- $\mathbf{M}_i, \mathbf{O}_i$: The weight matrices of the FFN input and output linear layers.

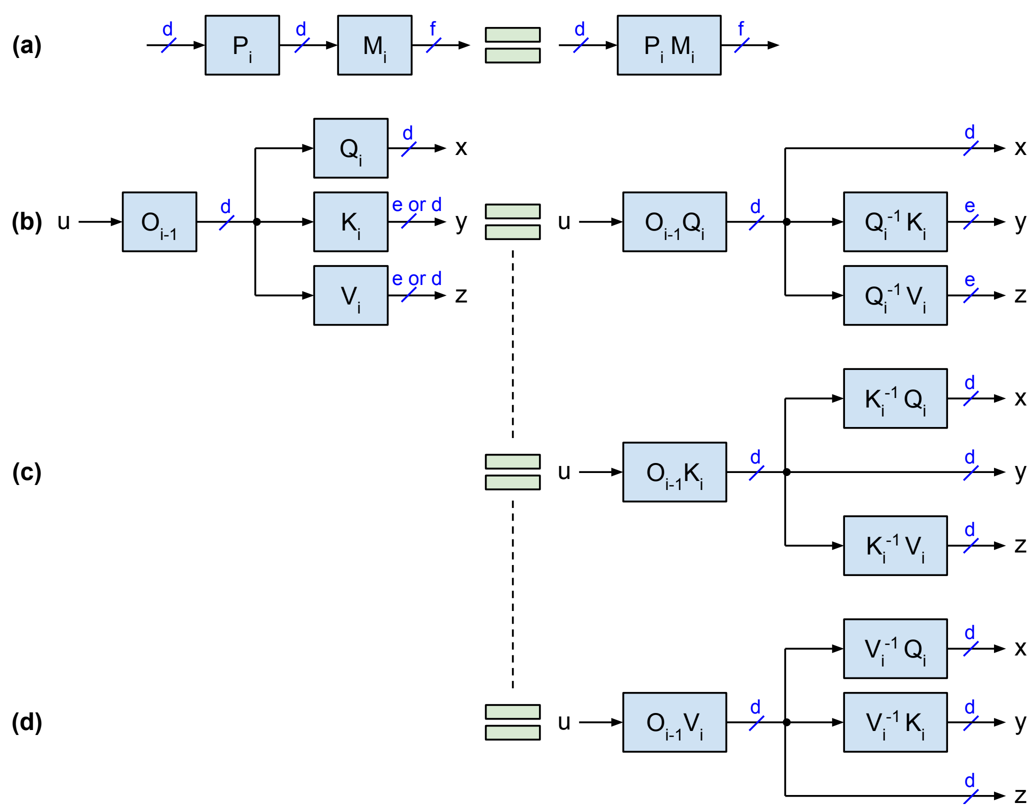

Figure 2: (a) Merging P and M; (b) eliminating Q; (c) eliminating K; (d) eliminating V.

Figure 2 details how the linear layers are merged:

- Figure 2(a) shows how the two linear layers with weight matrices $\mathbf{P}_i$ and $\mathbf{M}_i$ are collapsed and replaced by a single linear layer with weight matrix $\mathbf{M}_i^* = \mathbf{P}_i\mathbf{M}_i$, which eliminates d^2 weights.
- Figure 2(b) illustrates how to merge $\mathbf{Q}_i$ into the preceding $\mathbf{O}_{i-1}$-matrix, which eliminates d^2 weights and requires $\mathbf{Q}_i$ to be invertible. Note that $\vec{y} = \vec{u}\mathbf{O}_{i-1}(\mathbf{Q}_i\mathbf{Q}_i^{-1})\mathbf{K}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{K}_i$ and $\vec{z} = \vec{u}\mathbf{O}_{i-1}(\mathbf{Q}_i\mathbf{Q}_i^{-1})\mathbf{V}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{V}_i$.

- For MHA where $e = d$, $\mathbf{K}_i$ can be removed as shown in Figure 2(c), which eliminates d^2 weights. Note that $\vec{x} = \vec{u}\mathbf{O}_{i-1}(\mathbf{K}_i\mathbf{K}_i^{-1})\mathbf{Q}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{Q}_i$ and $\vec{z} = \vec{u}\mathbf{O}_{i-1}(\mathbf{K}_i\mathbf{K}_i^{-1})\mathbf{V}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{V}_i$. This requires that $\mathbf{K}_i$ is invertible.
- For MHA where $e = d$, $\mathbf{V}_i$ can be removed as shown in Figure 2(d), which eliminates d^2 weights. Note that $\vec{x} = \vec{u}\mathbf{O}_{i-1}(\mathbf{V}_i\mathbf{V}_i^{-1})\mathbf{Q}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{Q}_i$ and $\vec{y} = \vec{u}\mathbf{O}_{i-1}(\mathbf{V}_i\mathbf{V}_i^{-1})\mathbf{K}_i = \vec{u}\mathbf{O}_{i-1}\mathbf{K}_i$. This requires that $\mathbf{V}_i$ is invertible.

Table 1 specifies how the new weight matrices ($\mathbf{M}_i^*, \mathbf{Q}_i^*, \mathbf{K}_i^*, \mathbf{V}_i^*, \mathbf{O}_{i-1}^*$) of Figure 1 are calculated from the original ones. For the first transformer block ($i = 1$), we use the input embedding instead of $\mathbf{O}_{i-1}$ (because there is no $\mathbf{O}_{i-1}$ for $i = 1$).

	Figure 1(b)	Figure 1(c)	Figure 1(d)
$\mathbf{O}_{i-1}^*$	$\mathbf{O}_{i-1}\mathbf{Q}_i$	$\mathbf{O}_{i-1}\mathbf{K}_i$	$\mathbf{O}_{i-1}\mathbf{V}_i$
$\mathbf{Q}_i^*$	1 (eliminated)	$\mathbf{K}_i^{-1}\mathbf{Q}_i$	$\mathbf{V}_i^{-1}\mathbf{Q}_i$
$\mathbf{K}_i^*$	$\mathbf{Q}_i^{-1}\mathbf{K}_i$	1 (eliminated)	$\mathbf{V}_i^{-1}\mathbf{K}_i$
$\mathbf{V}_i^*$	$\mathbf{Q}_i^{-1}\mathbf{V}_i$	$\mathbf{K}_i^{-1}\mathbf{V}_i$	1 (eliminated)
$\mathbf{M}_i^*$	$\mathbf{P}_i\mathbf{M}_i$		

Table 1: How to calculate the new weight matrices from the original ones for Figure 1.

2 Parallel transformer without skip connections

Similar to the parallel transformer [16], Figure 3 shows parallel versions of Figures 1(b) to (d). Here, "parallel" refers to having the attention (including its linear layers) in parallel to the FFN.

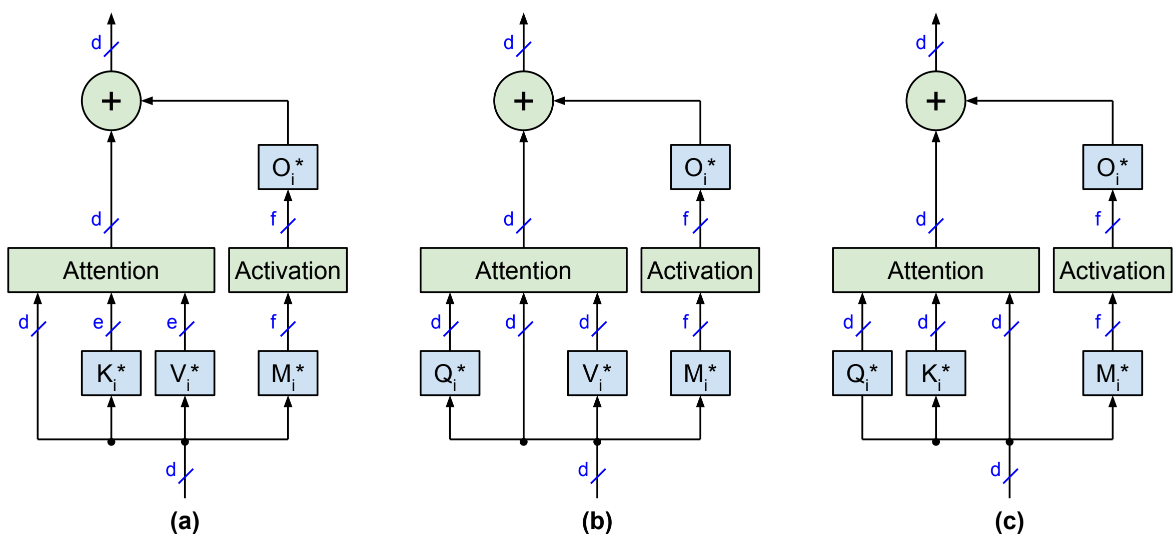

Figure 3: Parallel skipless transformers (a) without Q and P; (b) without K and P; (c) without V and P.

Figures 3(b) and (c) require that $e = d$, so they are only suitable for MHA, but not for MQA and GQA. Figure 3(a) is suitable for MHA, MQA, and GQA. Figure 3(c) is identical to the simplified transformer proposed in [1].

3 Examples

The table below lists the configurations and weight counts for Pythia-6.9B and Mistral-7B. For a skipless version of Mistral-7B we would save 15% of weights after merging the Q and P linear layers into the FFN layers. For a batch 1 system that is limited by memory bandwidth, these 15% weight savings can speed up inference by 1.17x during the autoregressive next-token-generation phase, see the table below.

Parameter	Pythia-6.9B	Mistral-7B	Notes
Parallel attention/FFN?	parallel	serial	[16]
MHA, MQA, or GQA?	MHA	GQA	[2, 3, 4]
`dim` (aka d)	4,096		embedding dimension
`n_layers`	32		number of layers
`n_heads`	32		number of heads
`n_kv_heads`	32	8	number of KV-heads
`e` (output dim. of K, V)	4,096	1,024	`e = d * n_kv_heads / n_heads`
FFN type	MLP	MLP with SwiGLU	[15]
FFN `hidden_dim`	16,384	14,336	FFN hidden dimension
`vocab_size`	50,400	32,000	vocabulary size
Number of weights (calculated from above parameters):			
Q+P weights per layer	33,554,432		`2 * dim * dim`
K+V weights per layer	33,554,432	8,388,608	`2 * dim * dim / n_heads * n_kv_heads`
FFN weights per layer	134,217,728	176,160,768	`(2 or 3) * dim * hidden_dim`
Input+output embed.	412,876,800	262,144,000	`2 * dim * vocab_size`
Total weights:	6.9B	7.2B	
Weight savings and speedup after removing Q and P:			
Total w/o Q+P weights:	5.8B	6.2B	total after removing Q and P
Weight savings:	**16%**	**15%**	
Possible speedup:	**1.19x**	**1.17x**	assumes batch size 1

4 Experiments

Refer to [11] for Python code that demonstrates the numerical equivalency of the weight reduction illustrated in Figures 1(b) and 2(b). The code also confirms that all square matrices of Mistral-7B are invertible.

5 Future work

Because skipless transformers are not very popular right now, future work should investigate whether removing P and Q (or K or V) is also beneficial for transformers with normalization and skip connections as illustrated in Figure 4. Adding normalization and skip connections again could simplify and speed up training relative to skipless transformers.

Acknowledgements

We would like to thank Bobby He (ETH Zürich) and James Martens (DeepMind) for helpful discussions on this work.

References

[1] Bobby He and Thomas Hofmann. Simplifying Transformer Blocks. November 2023. *arXiv:2311.01906*.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. June 2017. *arXiv:1706.03762*.

[3] Noam Shazeer. Fast Transformer Decoding: One Write-Head is All You Need. November 2019. *arXiv:1911.02150*.

[4] Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: Training generalized multi-query transformer models from multi-head checkpoints. May 2023. *arXiv:2305.13245*.

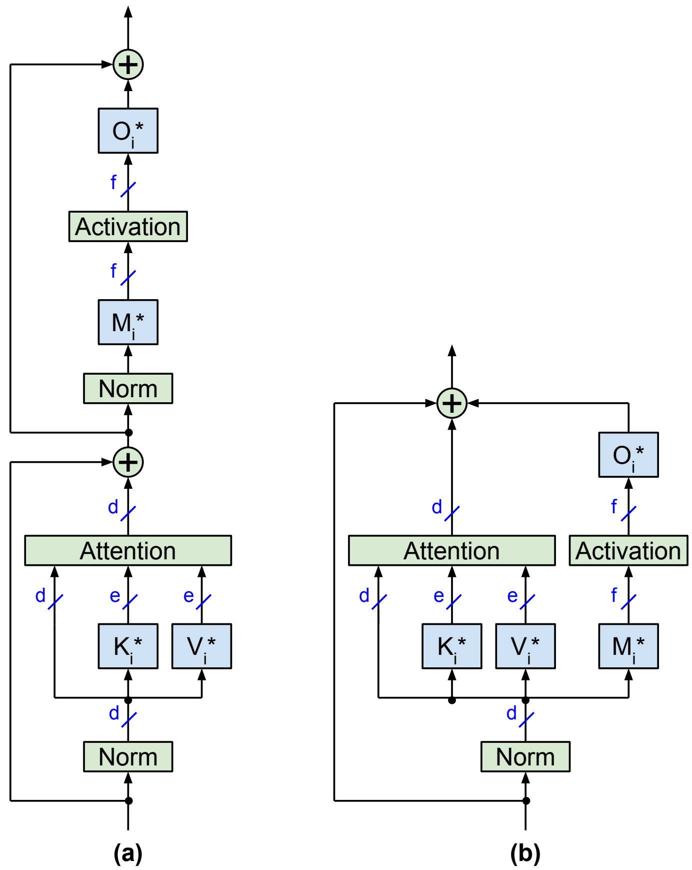

Figure 4: (a) Transformer block without Q and P; (b) version with parallel attention / FFN.

[5] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. July 2023. *arXiv:2307.09288*.

[6] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. October 2023. *arXiv:2310.06825*.

[7] Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of Experts. January 2024. *arXiv:2401.04088*.

[8] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari,

Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with Pathways. April 2022. *arXiv:2204.02311*.

[9] Gemma Team, Google DeepMind. Gemma: Open Models Based on Gemini Research and Technology. 2024.

[10] Frank Elavsky. The Micro-Paper: Towards cheaper, citable research ideas and conversations. February 2023. *arXiv:2302.12854*.

[11] OpenMachine. Transformer tricks. 2024. *Github repository*.

[12] Nils Graef. Transformer tricks: Precomputing the first layer. February 2024. *arXiv:2402.13388*.

[13] Bobby He, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L Smith, and Yee Whye Teh. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. February 2023. *arXiv:2302.10322*. And ICLR 2023.

[14] Wikipedia. Invertible matrix, 2024. Accessed Mar-2024.

[15] Noam Shazeer. GLU Variants Improve Transformer. February 2020. *arXiv:2002.05202*.

[16] Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. 2021. *Github repo*.