Communication Compression for Tensor Parallel LLM Inference

Jan Hansen-Palmus[1,2*] **Michael Truong Le**[1] **Oliver Hausdörfer**[2] **Alok Verma**[1]
[1]Recogni [2]Technical University of Munich
{jan.hansenpalmus,michael,alokge}@recogni.com
oliver.hausdoerfer@tum.de

Abstract

Large Language Models (LLMs) have pushed the frontier of artificial intelligence but are comprised of hundreds of billions of parameters and operations. For faster inference latency, LLMs are deployed on multiple hardware accelerators through various Model Parallelism strategies. Our paper looks into the details on one such strategy - Tensor Parallel - and proposes to reduce latency by compressing inter-accelerator communication. We leverage fine grained quantization techniques to compress selected activations by 3.5 - 4.5x. Our proposed method leads up to 2x reduction of time-to-first-token (TTFT) with negligible model performance degradation.

1 Introduction

Large Language Models (LLMs) have become essential across various applications due to their exceptional performance. As model performance tends to improve with increased parameter counts, LLMs have been significantly scaled in recent years, with contemporary models now reaching 500B+ parameters [Chowdhery et al., 2023].

Deploying such large models for inference presents major challenges [Pope et al., 2022]. Tensor Parallel [Shoeybi et al., 2020] addresses this by splitting layers on multiple accelerators, enabling the execution of extremely large models and significantly reducing latency. However, Tensor Parallel demands accumulation of results from accelerators, as shown in figure 1, and can lead to data communication bottlenecks [Zhuang et al., 2024, Agrawal et al., 2024], especially during the first auto-regressive inference step (the prefill phase).

One approach to mitigate these bottlenecks, and thus reduce model latency even further, is to quantize activations before communication, which reduces the time needed to accumulate results from accelerators in a Tensor Parallel group. However, the presence of outliers [Dettmers et al., 2022, Lin et al., 2023] complicates this strategy, necessitating fine-grained quantization approaches. We leverage such approaches proposed by Rouhani et al. [2023] to compress activations and demonstrate the potency of communication compression by measuring time-to-first-token (TTFT) in realistic inference scenarios using different inference hardware setups. We find that for hardware setups which have slower inter-accelerator bandwidths, the TTFT can be improved by 3.5 - 4.5x with negligible degradation of model performance.

*Research conducted during internship at Recogni

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Background

2.1 Parallel Inference

Given the massive parameter and operations requirement of LLMs [et al., 2024, Minaee et al., 2024], distributing their parameters and operations across a large number of accelerators is essential, which is achieved through different parallelism techniques. Among common parallelism techniques used to fit LLMs on multiple accelerators for inference [Shoeybi et al., 2020, Korthikanti et al., 2022, Rajbhandari et al., 2020, Zhuang et al., 2024], Tensor Parallel (TP) is usually the most widely used one because it allows a very effective way of reducing latency and scaling down model size per accelerator. Other parallelism strategies, such as Pipeline Parallelism or Sequence Parallel, are combined with TP to improve further upon it. TP has two possible sub-variants: Column-wise and Row-wise parallelism [Shoeybi et al., 2020], enable the partitioning of linear layers along the column or row dimension of weights respectively.

While Tensor Parallel offers significant benefits, it also increases communication overhead, as results computed on individual accelerators must be frequently synchronized using collective operations [Clarke et al., 1994], [Nvidia, 2023]. This becomes particularly problematic during the prefill phase, where activation tensors for the entire token sequence need to be transmitted. This can lead to communication bottlenecks [Zhuang et al., 2024, Agrawal et al., 2024], especially if the inter-accelerator bandwidth is low. While Bian et al. [2024] explored various approaches to reduce communication overhead during training, to the best of our knowledge, this issue has not yet been addressed at inference-time by the research community.

2.2 Model Quantization

The quantization of weights and/or activations of LLMs [Zhao et al., 2024, Lin et al., 2023, Sheng et al., 2023, Kang et al., 2024, Xiao et al., 2024, Yuan et al., 2023, Frantar et al., 2023] has been a very active field of research. Quantization of weights to lower bit-widths reduces the memory needed to store parameters and also improves throughput due to better memory bandwidth utilization. Furthermore, quantization of activations and also computations allows reducing activation memory footprint and improves throughput as well by utilizing faster and cheaper computation engines on recent accelerators [NVIDIA, 2024].

Unlike weights, activations are hard to quantize due to their dynamic nature [Zhao et al., 2024, Sheng et al., 2023] and the presence of outliers, which has been extensively studied in literature [Dettmers et al., 2022, Lin et al., 2023, Xiao et al., 2024]. Dettmers et al. [2022] found that accurately representing these outliers is critical for maintaining model performance: Even though outliers represent only a small fraction of input features, removing them leads to a significant degradation in Perplexity.

A common approach to activation quantization is to normalize the values based on the tensor's absolute maximum. However, this can be problematic due to the presence of outlier values. Using the absolute maximum, which is dominated by these outliers, leads to poor representation of the remaining values in the tensor, as they become compressed into a narrow range and lose precision.

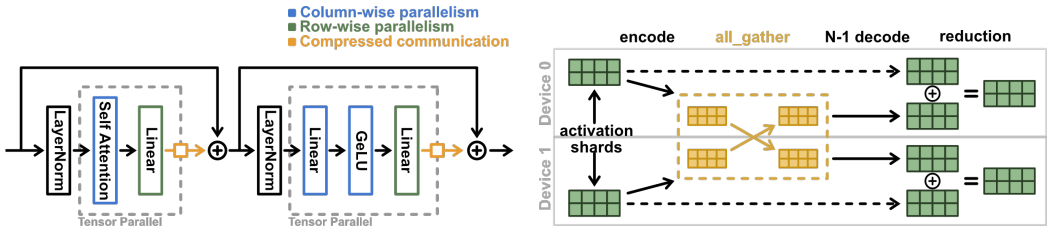
To address this issue, various solutions have been developed. One approach is to limit the impact of outliers by grouping tensor values and quantizing them together, which helps reducing the quantization error. Mixed precision methods [Zhao et al., 2024, Dettmers et al., 2022, Kang et al., 2024, Hooper et al., 2024], on the other hand, distinguish between outliers and non-outliers, using different data types for each group during quantization.

Since we compress activations of specific layers, in the following section, we briefly review methods related to low-bit activation quantization.

3 Related Work

3.1 KV-cache Quantization

LLM inference involves generating tokens in an auto-regressive manner, and the majority of the computation time is spent accessing the KV-cache from memory [Hooper et al., 2024, Zirui Liu et al.,



(a) (b)

Figure 1: An illustration of transformer-based LLM model parallelized using TP. In Figure 1a, column-wise and row-wise TP layers are marked blue and red, respectively. Before reduction, we propose to compress the all_gather collective op (orange in Figure 1a), as presented in Figure 1b.

2023]. KV-cache quantization addresses this issue by quantizing Key and Value activations, reducing the effective size of the KV-cache.

Zirui Liu et al. [2023] observed distinct outlier characteristics between Key and Value caches and apply per-channel and per-token quantization respectively. Per-channel quantization groups and quantizes values within the same channel while per-token quantization applies this to token dimension. Sheng et al. [2023] employs group-wise quantization, which takes per-channel quantization a step further by grouping small blocks of values within each channel and quantizing them independently, which reduces quantization errors. Hooper et al. [2024], Kang et al. [2024] propose extracting a sparse outlier matrix composed of the top 1-2% of the largest values, which are preserved in higher precision.

Unlike KV-cache activation compression, we target communication compression occurring after execution of row-wise TP linear layers, as shown in figure 1a and 1b. Compression and decompression of communication has to be done at much lower latency, otherwise improvements achieved by communication compression are offset by encoding and decoding steps.

3.2 Weight and Activation Quantization

The primary objective of weights and activation quantization is to reduce the overall model size while accelerating inference [Wu et al., 2023, Zhao et al., 2024] by the use of specialized hardware, such as Nvidia's INT4 Tensor Cores [NVIDIA, 2024].

Zhao et al. [2024] proposed a mixed-precision technique that addresses the challenge of activation quantization by representing the 128 largest outlier channels with 8 bits per value. OCP Specification [2023] introduces low-bit data formats applicable to matrix multiplication operations. In this method, a block of values is encoded in a low-bit floating-point format along with a shared exponent. Given the strong performance of OCP Specification [2023] [Rouhani et al., 2023] data types, we based our work on their provided code [Mircosoft, 2024] and apply low-bit data types for activation compression experiments as well.

4 Method

4.1 Communication Compression

Figure 1a highlights our approach to communication compression in a TP setting. We aim to compress the partial results of each worker after row-wise TP linear layers, and decompress them before reduction in each worker. Since our method introduces extra computations which offsets slower inter-accelerator communication, we must strike a balance between computation and compression. We found that block-wise quantization as proposed in OCP Specification [2023] for low-bit compression is a strong contender to balance quantization error and compression latency. So, we evaluate a variety of low-bit data types using this fine-grained block-wise quantization scheme.

We also extended the data types available in [Mircosoft, 2024] to experiment with more variety of bit-widths to test higher compression. The following data types and parameters were evaluated:

1. **Value data types**: FP5 (E3M1, E2M2, E1M3), FP4 (E2M1, E1M2), FP2 (E1M1), INT3, INT4, INT5
2. **Block size**: 8, 16, 32
3. **Scale data type**: E8M0, E7M0, E6M0, E5M0, E4M0

4.2 Model Evaluation

We measure the performance of compression approaches based on the compression rate, which is measured by the number of *effective bits* [Frantar et al., 2023, Lin et al., 2023], and the increase in Perplexity metric [Hooper et al., 2024, Zhao et al., 2024, Dettmers et al., 2022, Wei et al., 2023], relative to the compression-free model with 16 bit (FP16) activations. We show that each possible hyper-parameter of quantization data type proposed in OCP Specification [2023] can have varied effect on latency and model's Perplexity in Section 5.1.

4.3 Profiling

To show we can reduce communication bottlenecks by utilizing our most performant quantization approaches, we measure the TTFT of models of different sizes in a deployment scenario. We base our profiling on the code provided by IBM [2023] which uses `torch.compile` [Wen, 2023] to speed up inference of Llama 2 models along with TP. The architecture of the Llama 2 model family is very similar to other state-of-the-art LLMs and therefore provides valuable compression insights. We extended the code to add communication compression. In our profiling setup, each worker in a TP group of N workers compresses output activations of each row-wise linear layer before communication, and decompresses N-1 activations gathered from all other workers. We finally reduce the decompressed activations using `torch.sum` as shown in Figure 1b.

5 Experiments

5.1 Optimal Compression Scheme Search

To pick an optimal compression scheme for each model we first evaluate the Perplexity of various state-of-the-art LLMs on 10% of the Wikitext train dataset, which we deemed sufficient for this purpose, Merity et al. [2016] by testing a subset of combinations of data types and block sizes. Please refer to A.1 for further details how we restricted the search space of quantization hyper-parameters prior to these experiments.

Next, we pick the final compression schemes for each model using the results from Table 1: To ensure minimal model performance degradation, we consider only combinations which lead to an increase in Perplexity of less than 3%. From the remaining candidates, we then choose the one with the largest compression ratio (lowest effective bits). This procedure ensures a good balance between compression ratio and quantization accuracy. Finally, to validate the performance of our chosen compression schemes, we evaluate them on the entire Wikitext test set (Table 2). We observe minimal performance degradation with respect to 16 bit uncompressed communication, while being able to compress tensors by a factor of roughly 3.3X.

5.2 Measuring TTFT Speedups

We demonstrate the potential speedups achievable by compressing communication in 3. As IBM [2023] only supports the Llama 2 model family, we decided to conduct measurements using the FP4 quantization scheme deemed best by Rouhani et al. [2023], which has a similar compression ratio to the schemes we picked in 5.1. We achieve a speed-up between 2x - 1.2x based on the TP degree and hardware setup. We profile our method on cutting-edge NVIDIA accelerators: L4, and A100 which are accessed through Google Cloud Platform [Google, 2024]. L4 GPUs in a node are connected with PCIe Gen4 x16 and have 64GB/s bandwidth, while A100 have 600 GB/s bidirectional any-to-any bandwidth. These both types of GPU provide good coverage of computing throughput(FLOPs/sec) and GPU-GPU bandwidth to showcase communication compression benefits. Generally, setups containing more than two L4 GPUs greatly benefit from our approach of communication compression. Due to the rather slow interconnect, these setups are heavily bottle-necked by communication

Value Dtype	Block Size	Eff. Bits	Llama 3.1		Gemma2		Mistral		
			8B	70B	2B	9B	7B	22B	123B
FP16	-	16	7.12	4.24	14.50	10.17	5.55	4.16	2.66
FP3	8	3.6	9.39%	11.42%	12.17%	5.49%	3.40%	9.24%	9.29%
	16	3.3	13.87%	14.78%	19.59%	7.10%	4.46%	11.79%	11.79%
	32	3.2	19.67%	18.50%	31.07%	13.51%	5.50%	13.23%	14.58%
FP4	8	4.6	2.92%	3.91%	3.91%	1.47%	1.27%	2.36%	3.39%
	16	4.3	3.01%	3.85%	4.64%	1.94%	1.31%	6.58%	3.41%
	32	4.2	3.37%	4.14%	6.30%	2.20%	1.22%	6.69%	3.53%
FP5	8	5.6	0.58%	1.09%	0.85%	-0.19%	0.49%	0.30%	0.69%
	16	5.3	0.57%	1.14%	1.17%	0.56%	0.52%	0.35%	0.66%
	32	5.2	0.70%	1.22%	1.54%	1.02%	0.49%	0.44%	0.59%

Table 1: Perplexity evaluation is done on 10% of the Wikitext2 training set and the degradation is reported relative (in %) to the absolute FP16 performance in the first row. Grid search over block size and effective bits are done which show a significant effect on model performance. For FP3, FP4 and FP5, we chose E1M1, E2M1 and E2M2 sub-variants respectively.

Model	Sub-variant	Value Dtype	Block Size	Bits	Perplexity FP16	Perplexity Increase
Llama 3.1	8B	FP4	8	4.6	7.22	3.22%
	70B	FP5	32	5.2	3.86	1.68%
Gemma 2	2B	FP5	32	5.2	14.27	1.39%
	9B	FP4	32	4.2	10.40	1.83%
Mistral	7B	FP4	32	4.2	5.23	1.18%
	22B	FP4	8	4.6	4.02	1.62%
	123B	FP5	32	5.2	2.65	0.48%

Table 2: Evaluation of the best-performing quantization schemes on Wikitext2 test set. The Perplexity column shows the relative degradation of the compressed model performance compared to uncompressed FP16 model. As in Table 1, for FP4 and FP5 the sub-variants E2M1 and E2M2 are chosen, respectively.

overhead. In contrast, our A100 setup is slowed down by our quantization scheme: Due to the fast interconnect, the quantization overhead is so large that the we don't benefit from data compression.

5.3 SoTA Comparison

We compare our compression method with the relevant state-of-the-art method described in [Bian et al., 2024] which proposes multiple learned and non-learned methods for communication compression. Since our method targets inference-only non-learned optimization, we compare to the two fastest non-learned approaches: channel-wise INT quantization, and TopK compression which keeps the K largest magnitudes and zeroes all other values.

In Table 4, we show that both of their compression techniques are lead to much higher degradation of Perplexity metric. INT4 compression offers substantial speedups due to the minimal computational overhead, but shows higher Perplexity degradation when compared to fine-grained quantization approach. Improving the accuracy of this approach is left for future work.

Model	Accelerators	Input	TTFT [s] Uncompressed	Compressed	Speedup
LLama 2 70b	8xL4	2x64	0.58 (0.04)	0.32 (0.68)	1.83
		2x128	1.07 (0.04)	0.52 (0.63)	2.08
	4xA100	2x128	0.09 (0.00)	0.15 (0.00)	0.56
		2x256	0.13 (0.00)	0.19 (0.00)	0.70
Llama 2 13b	4xL4	8x128	0.67 (0.00)	0.33 (0.00)	2.05
		8x256	1.37 (0.00)	0.70 (0.00)	1.96
Llama 2 7b	2xL4	16x128	0.39 (0.00)	0.45 (0.00)	0.88
		16x256	0.79 (0.00)	0.77 (0.00)	1.03

Table 3: Inference profiling results are shown for different models and TP configurations. For compression, quantization scheme with value data type FP4 E2M1, batch size 32 and scale data type E8M0 was used, which has 4.25 effective bits. The TTFT values correspond to the median of 32 model forward passes, additionally we report the standard deviation in brackets.

	Perplexity			**TTFT** (Llama 2 70B)	
	Llama 3.1 8B	Gemma2 2B	Mistral 7B	8xL4	4xA100
FP16	7.22	14.27	5.23	1.06s	0.13s
MX4 E2M1	3.22%	6.06%	1.17%	2.07x	0.70x
Int4	6.19%	8.83%	15.05%	2.60x	0.95x
TopK 3x	115.54%	80.17%	21.38%	1.80x	0.55x

Table 4: Comparison with methods from Bian et al. [2024]. MX4 E2M1 (batch size 32 and E8M0 as scaling dtype) is compared to channel-wise INT4 and TopK compression using the Wikitext2 test set. The speedup of the compressed model is compared to the uncompressed FP16 model, we present absolute numbers in the first row. Input shapes 2x128 and 2x256 were used for the setups 8xL4 and 4xA100 respectively.

6 Conclusion and Limitations

Our work builds on the foundation set by Rouhani et al. [2023] and OCP Specification [2023] by incorporating additional data types and tuning parameters to optimize compression. To achieve this, we developed a model-dependent hyper-parameter selection procedure that effectively balances the compression rate with model accuracy. The proposed compression method reduces activation sizes by a factor of 3.5 to 4.5x, with minimal degradation in performance which results in an improvement of LLMs' TTFT by a factor of 1.2 to 2x, depending on the hardware setup.

While our compression methods shows substantial TTFT improvements for some hardware configurations, it does not lead to improvement when inter-accelerator bandwidth is substantially high and inference is not bandwidth-bound anymore. Furthermore, the latency introduced by compression algorithms can offset any communication compression speedup, so compressing more but with slower algorithms is not helpful. It could be possible to accelerate compression by utilizing specialized hardware, but such strategies are beyond the current scope and are left for future investigation. Furthermore, by improving model quantization techniques and adopting 8 bit matrix multiplications, the communication size of TP linear layers could be reduced easily without extra compression costs.

Our current profiling setup could also be enhanced further to take into account in-flight batching and other parallelism strategies and provide more extensive details of throughput improvements.

Acknowledgments and Disclosure of Funding

We acknowledge the insights offered by Frederik Gerzer and Thomas Pfeil, which helped shape the experiments presented in this paper. We are also grateful to the engineering and research teams at Recogni who provided necessary infrastructure to run various experiments on cloud machines easily.

References

A. Agrawal, J. Chen, Í. Goiri, R. Ramjee, C. Zhang, A. Tumanov, and E. Choukse. Mnemosyne: Parallelization strategies for efficiently serving multi-million context length llm inference requests without approximations. *arXiv preprint arXiv:2409.17264*, 2024.

S. Bian, D. Li, H. Wang, E. Xing, and S. Venkataraman. Does compressing activations help model parallel training? *Proceedings of Machine Learning and Systems*, 6:239–252, 2024.

A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

L. Clarke, I. Glendinning, and R. Hempel. The mpi message passing interface standard. In K. M. Decker and R. M. Rehmann, editors, *Programming Environments for Massively Parallel Distributed Systems*, pages 213–218, Basel, 1994. Birkhäuser Basel. ISBN 978-3-0348-8534-8.

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. URL https://arxiv.org/abs/2208.07339.

D. et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Google. Google Cloud Platform. https://cloud.google.com/?hl=en, 2024. Accessed: 2024-09-28.

C. Hooper, S. Kim, H. Mohammadzadeh, M. W. Mahoney, Y. S. Shao, K. Keutzer, and A. Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization, 2024. URL https://arxiv.org/abs/2401.18079.

IBM. Foundation model stack. https://github.com/foundation-model-stack/foundation-model-stack, 2023.

H. Kang, Q. Zhang, S. Kundu, G. Jeong, Z. Liu, T. Krishna, and T. Zhao. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm, 2024. URL https://arxiv.org/abs/2403.05527.

V. Korthikanti, J. Casper, S. Lym, L. McAfee, M. Andersch, M. Shoeybi, and B. Catanzaro. Reducing activation recomputation in large transformer models, 2022. URL https://arxiv.org/abs/2205.05198.

J. Lin, J. Tang, H. Tang, S. Yang, X. Dang, and S. Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *ArXiv*, abs/2306.00978, 2023. URL https://api.semanticscholar.org/CorpusID:271271084.

S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models, 2016.

S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196.

Mircosoft. microxcaling. https://github.com/microsoft/microxcaling/tree/main, 2024.

Nvidia. Nccl library. https://developer.nvidia.com/nccl, 2023. Accessed: 2024-09-28.

NVIDIA. Blackwell Architecture Overview. https://resources.nvidia.com/en-us-blackwell-architecture, 2024. Accessed: 2024-09-28.

OCP Specification. Ocp microscaling formats (mx) specification. https://www.opencompute.org/documents/ocp-microscaling-formats-mx-v1-0-spec-final-pdf, 2023. Accessed: 2024-09-28.

R. Pope, S. Douglas, A. Chowdhery, J. Devlin, J. Bradbury, A. Levskaya, J. Heek, K. Xiao, S. Agrawal, and J. Dean. Efficiently scaling transformer inference, 2022. URL https://arxiv.org/abs/2211.05102.

S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He. Zero: Memory optimizations toward training trillion parameter models, 2020. URL https://arxiv.org/abs/1910.02054.

B. D. Rouhani, R. Zhao, A. More, M. Hall, A. Khodamoradi, S. Deng, D. Choudhary, M. Cornea, E. Dellinger, K. Denolf, S. Dusan, V. Elango, M. Golub, A. Heinecke, P. James-Roxby, D. Jani, G. Kolhe, M. Langhammer, A. Li, L. Melnick, M. Mesmakhosroshahi, A. Rodriguez, M. Schulte, R. Shafipour, L. Shao, M. Siu, P. Dubey, P. Micikevicius, M. Naumov, C. Verrilli, R. Wittig, D. Burger, and E. Chung. Microscaling data formats for deep learning, 2023. URL https://arxiv.org/abs/2310.10537.

Y. Sheng, L. Zheng, B. Yuan, Z. Li, M. Ryabinin, D. Y. Fu, Z. Xie, B. Chen, C. Barrett, J. E. Gonzalez, P. Liang, C. Ré, I. Stoica, and C. Zhang. Flexgen: High-throughput generative inference of large language models with a single gpu, 2023. URL https://arxiv.org/abs/2303.06865.

M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism, 2020. URL https://arxiv.org/abs/1909.08053.

X. Wei, Y. Zhang, Y. Li, X. Zhang, R. Gong, J. Guo, and X. Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling, 2023. URL https://arxiv.org/abs/2304.09145.

W. Wen. torch.compile tutorial. https://pytorch.org/tutorials/intermediate/torch_compile_tutorial.html, 2023. Accessed: 2024-09-28.

X. Wu, C. Li, R. Y. Aminabadi, Z. Yao, and Y. He. Understanding int4 quantization for transformer models: Latency speedup, composability, and failure cases. *arXiv preprint arXiv:2301.12017*, 2023.

G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024. URL https://arxiv.org/abs/2211.10438.

Z. Yuan, L. Niu, J. Liu, W. Liu, X. Wang, Y. Shang, G. Sun, Q. Wu, J. Wu, and B. Wu. Rptq: Reorder-based post-training quantization for large language models, 2023. URL https://arxiv.org/abs/2304.01089.

Y. Zhao, C.-Y. Lin, K. Zhu, Z. Ye, L. Chen, S. Zheng, L. Ceze, A. Krishnamurthy, T. Chen, and B. Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving, 2024. URL https://arxiv.org/abs/2310.19102.

Y. Zhuang, H. Zhao, L. Zheng, Z. Li, E. P. Xing, Q. Ho, J. E. Gonzalez, I. Stoica, and H. Zhang. On optimizing the communication of model parallelism, 2024. URL https://arxiv.org/abs/2211.05322.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, V. Braverman, Beidi Chen, and X. Hu. Kivi : Plug-and-play 2bit kv cache quantization with streaming asymmetric quantization. 2023. doi: 10.13140/RG.2.2.28167.37282. URL https://rgdoi.net/10.13140/RG.2.2.28167.37282.

A Appendix / supplemental material

A.1 Ablation over quantization hyper-parameters

We ablate over the extended space of possible data types and parameters (4.1). We aim at restricting our parameter space by picking parameters (Table 5) which promise good quantization accuracy for different compression ratio requirements. As scale data type, E5M0 was generally the best option, as lower bits for scale introduce unacceptable Perplexity degradation while higher do not lead to any improvement. For value bits we found the data types FP4 E2M1 or FP5 E2M2 were sufficient to cover different required compression ratios, while FP3 E1M1 showed large Perplexity increase. Since the choice of the block size has a large effect on Gemma's performance, we decided to keep all possible parameters. We deem it necessary to evaluate all of these compression configuration because naively just choosing the highest compression configuration leads to high degradation of model Perplexity.

	Parameter	Perplexity increase [%]		
		Llama 3.1B	Mistral 7B	Gemma 2B
Scale Bits	4	3.67	6.11	7.46
	5	3.37	1.22	6.30
	6	3.37	1.29	6.30
	7	3.37	1.29	6.30
Value Data Type	FP3 E1M1	19.67	5.52	31.07
	FP4 E1M2	4.45	1.27	9.95
	FP4 E2M1	3.37	1.29	6.30
	FP5 E1M3	0.90	0.38	3.35
	FP5 E2M2	0.70	0.45	1.54
	FP5 E3M1	2.63	1.18	3.02
	INT 3	19.67	5.52	31.07
	INT 4	4.45	1.27	9.95
	INT 5	0.90	0.38	3.35
Block Size	8	2.92	1.27	3.91
	16	3.01	1.37	4.64
	32	3.37	1.29	6.30
Parallelism	2	3.37	1.29	6.30
	4	3.35	1.13	5.22
	8	2.83	1.03	4.83
	16	2.88	0.97	4.59
	32	2.88	0.95	3.80

Table 5: Ablation over scale bits, value bits and type, block size and degree of parallelism. Evaluation on 10% of the Wikitext2 training set.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We provide experimental results in the paper to validate our claims stated in the abstract.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations we are aware of, in the last section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.

- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Most of the code libraries needed to reproduce the results are mentioned in the paper. The authors intend to release scripts to reproduce results more easily soon.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example

(a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Company-internal code was used to produce experimental results. But authors intend to soon release trimmed down scripts for easy repeatability.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We either directly put these details in table captions, or described them when explaning the experimental setup.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Most of our experiments are deterministic. For the profiling experiments, we reported the standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We mentioned we used GCP to run our experiments. The exact compute usage cannot be disclosed, since this paper was written in the context of a Master's thesis, which includes other experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: As we only aim at reducing latencies, we do not believe there are any societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: We don't believe our research can be misused, as we only aimed at improving model latency and didn't release any potentially harmful data.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cited all owners of assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

Justification: We did not release any assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.