Compressing Large Language Models to Any Size Without Re-Computation

Martin Genzel [*1] Patrick Putzky [*1] Pengfei Zhao [*2 3] Sebastian Schulze [1] Mattes Mollenhauer [1]
Robert Seidel [1] Stefan Dietzel [1] Thomas Wollmann [1]

Abstract

The adoption of Foundation Models in resource-constrained environments remains challenging due to their large size and inference costs. A promising way to overcome these limitations is post-training compression, which aims to balance reduced model size against performance degradation. This work presents *Any Compression via Iterative Pruning* (ACIP), a novel algorithmic approach to determine a compression-performance trade-off from a single stochastic gradient descent run. To ensure parameter efficiency, we use an SVD-reparametrization of linear layers and iteratively prune their singular values with a sparsity-inducing penalty. Importantly, the resulting pruning order gives rise to a global parameter ranking that allows compressing a model to any target size without requiring re-computation. We evaluate ACIP on a large selection of open-weight LLMs and downstream tasks, demonstrating state-of-the-art results compared to existing factorization-based compression methods.

1. Introduction

Post-training compression of Foundation Models, especially Large Language Models (LLMs), promises access to powerful tools where resources are limited, e.g., in automotive systems, mobile deployments, or on the shop floor. Typical reasons for scarcity include constrained access to hardware, monetary limitations, high inference speed requirements, and environmental concerns (Hohman et al., 2024).

As model compression is almost always lossy, a fundamental trade-off arises between the model size and desired downstream performance. While characterizing this trade-off supports practitioners in deployment decisions (Boggust et al., 2025), the scientific literature typically focuses on benchmarks with preset compression levels (Zhu et al., 2024). Consequently, in real-world applications, practitioners often find themselves adjusting their needs to a set of predefined compression parameters. We argue that the situation should be the other way round, and that a user should be enabled to adjust compression parameters to their individual needs.

We therefore advocate for approaches that permit **Any Compression** of pre-trained models. Here, '*Any*' signifies an algorithm's ability to scale a given base model to any desired target size, guided by the user's specific needs and limitations, rather than the algorithm dictating possible sizes. To facilitate decision-making, such an algorithm should allow accessing the compression-performance trade-off efficiently, without requiring extensive additional computations.

In this work, we develop *Any Compression via Iterative Pruning* (ACIP).[1] To the best of our knowledge, ACIP is the first algorithm that enables **model compression to any size in real-time** without requiring re-computation or re-calibration. As detailed in Section 2 below, the Any Compression feature is achieved by explicitly decoupling an (optimization-based) pruning stage from the actual compression stage. We empirically demonstrate the effectiveness of our approach in Section 3, accompanied by a series of additional experiments in the supplementary material (Appendix E). After a brief discussion of related literature in Section 4, we conclude in Section 5, also pointing out some limitations of our study.

2. Any Compression via Iterative Pruning

The approach of *Any Compression via Iterative Pruning* (ACIP) aims to compress linear layers of the form

$$\mathbf{y} = \mathbf{W}\mathbf{x} + \mathbf{b}, \tag{1}$$

where $\mathbf{W}$ is an $m \times n$ matrix, $\mathbf{b}$ is a bias term, and $\mathbf{x}$ and $\mathbf{y}$ are layer inputs and outputs, respectively. Linear layers are the molecular building blocks of modern ML models, typically accounting for more than 90% of model parameters in transformers (Vaswani et al., 2017), which makes them a natural target for compression.

[*]Equal contribution [1]Merantix Momentum GmbH, Berlin, Germany [2]Work done while at Merantix Momentum. [3]Understandable Machine Intelligence Lab, Leibniz Institute for Agriculture and Bioeconomy (ATB), Potsdam, Germany. Correspondence to: Martin Genzel <martin.genzel@merantix-momentum.com>.

Workshop on Efficient Systems for Foundation Models (ES-FoMo) at the 42nd International Conference on Machine Learning (2025), Vancouver, Canada. Copyright 2025 by the author(s).

[1]*Pronounced* like "a sip" of coffee.

Previous research has explored LLM compression through reparametrization using singular value decomposition (SVD) (Mirsky, 1960) followed by rank reduction (Idelbayev & Carreira-Perpiñán, 2020; Yuan et al., 2024; Wang et al., 2024). While ACIP builds on SVD-based low-rank approximations as well, unlike existing methods, it achieves Any Compression by decoupling the pruning and compression stages. More specifically, ACIP constructs a score map (LeCun et al., 1989; Hassibi et al., 1993) that establishes a global importance ranking of singular values across all linear layers within the network. Initially, the score map is derived by running a (stochastic) gradient descent on a sparsity-inducing objective (Tibshirani, 1996; Efron et al., 2004; Mairal & Yu, 2012; Hastie et al., 2015), using the pruning order of the singular values as a proxy for feature importance. In an independent step, the score map can then be used to compress the base model to any desired size without requiring re-computation.

Algorithmically, ACIP consists of the following three key steps, which are detailed in the sections below. For a visualization of ACIP, we refer to Figure A3.

Step 1. (Model Reparametrization) Apply SVD to the weights $\mathbf{W}_l$ of all linear layers based on (2).

Step 2. (Scoring via Iterative Pruning) Choose a surrogate loss $\mathcal{L}$ and a calibration data set $\mathbf{X}$. Perform iterative pruning of singular values and tuning of low-rank adapters $\boldsymbol{\Delta}$. Obtain a parameter score map $\boldsymbol{\rho}$.

Step 3. (Any Compression) Choose *any* desired compression rate. Use the score map $\boldsymbol{\rho}$ and adapters $\boldsymbol{\Delta}$ to materialize the compressed model in real-time.

2.1. Step 1. Model Reparametrization

We start by reparametrizing all linear layers of a network[2] using SVD and assign

$$\mathbf{W}_l \leftarrow \mathbf{U}_l \mathbf{M}_l \boldsymbol{\Sigma}_l \mathbf{V}_l^\top + \boldsymbol{\Delta}_l, \tag{2}$$

where l denotes the layer index, $\mathbf{M}_l$ is a diagonal matrix with entries $\mathbf{m}_l^{(i)} \in \{0,1\}$ masking the singular values $\mathbf{s}_l^{(i)}$ in $\boldsymbol{\Sigma}_l$, $\mathbf{U}_l$ and $\mathbf{V}_l$ are $m \times r$ and $n \times r$ matrices of (orthonormal) singular vectors, and $\boldsymbol{\Delta}_l$ is a low-rank adapter (LoRA) (Hu et al., 2022). We freeze $\boldsymbol{\Sigma}_l$, $\mathbf{U}_l$, and $\mathbf{V}_l$.

We find that adding a low-rank adapter helps to compensate for potential errors that are introduced through pruning in Step 2. We initialize $\mathbf{M}_l$ as the identity matrix and $\boldsymbol{\Delta}_l$ as zero weights. In this way, the reparametrized model remains identical to the original model up to numerical precision.

The above parametrization leads to a parameter-efficient compression scheme. Indeed, given an $m \times n$ matrix $\mathbf{W}$, the number of non-zero singular values is bounded by $r = \min(m,n)$, which means the number of tunable mask parameters scales linearly in the feature dimensions.

We parametrize the binary masks through a thresholding operation of the form

$$\mathbf{m}_l^{(i)} = \begin{cases} 0, & \text{for } \mathbf{p}_l^{(i)} \leq 0 \\ 1, & \text{for } \mathbf{p}_l^{(i)} > 0 \end{cases}, \tag{3}$$

where $\mathbf{p}_l^{(i)}$ are scalar learnable parameters. As this operation is not differentiable, we use the straight-through estimator for backpropagation (Bengio et al., 2013; Yin et al., 2018).

2.2. Step 2. Scoring via Iterative Pruning

We now aim to build a global score map over all singular values in the reparametrized layers, which can guide model compression subsequently in Step 3. Leveraging the sparsity-inducing property of ℓ_1-regularization, we progressively shrink the mask parameters $\mathbf{p}_l^{(i)}$ to zero and derive a score map based on the pruning order. The two key algorithmic components of this "iterative scoring" strategy are presented next.

Iterative Pruning The optimization problem solved by ACIP takes the form

$$\min_{\mathbf{p},\boldsymbol{\Delta}} \mathcal{L}\big(\mathbf{X};\boldsymbol{\theta},\mathbf{p},\boldsymbol{\Delta}\big) + \lambda \left\|\mathbf{p}\right\|_1, \tag{4}$$

where $\mathcal{L}$ denotes a suitable calibration loss for the model, $\mathbf{p} = \{\mathbf{p}_0, \ldots, \mathbf{p}_L\}$ the set of all mask parameters, $\boldsymbol{\Delta} = \{\boldsymbol{\Delta}_0, \ldots, \boldsymbol{\Delta}_L\}$ the set of all low-rank adapters, and $\boldsymbol{\theta}$ the set of all remaining model parameters that are frozen during optimization. We perform gradient-based optimization until a preset stopping compression rate r_{stop} is reached (see Appendix E.6 for further discussion). Optionally, we perform post-tuning for a fixed number of steps by freezing the masks $\mathbf{p}$ and continue optimizing the low-rank adapters $\boldsymbol{\Delta}$.

Remark 2.1 (Scaling of λ). If λ is chosen too small, the stopping rate r_{stop} might never be reached. If λ is too large, training might become unstable and the score map ambiguous. Therefore, we use a simple linear scheduler that scales λ by a fixed factor >1 every j optimization steps, so that pruning becomes increasingly aggressive over time.

From Iterative Pruning to a Score Map The optimization process of (4) is used to inform our score map. We hypothesize that there is a close relationship between the order in which the parameters $\mathbf{p}_l^{(i)}$ are pruned and their importance for the model — the least important parameters are pruned first and so on.

Based on the pruning order, the score map $\boldsymbol{\rho}$ is updated iteratively in order to represent these feature importances. A negative number in the map indicates how many steps ago a

[2]Following common practice, we ignore the embedding layer and classification head in (decoder-only) transformers.

parameter was pruned. For all parameters that have not been pruned, the score is set to the value of the corresponding parameter; see Appendix E.11 for visual examples of `ACIP` score maps. Our approach ensures that (i) the score map stores the pruning history, and (ii) it estimates future pruning based on the parameter magnitudes. Note that absolute values of the score are irrelevant for parameter ranking.

2.3. Step 3. Any Compression

From Step 2, we only retain the score map $\boldsymbol{\rho}$ and the low-rank adapters $\boldsymbol{\Delta}$. In particular, the pruned masks $\mathbf{m}_l^{(i)}$ are discarded, as they are irrelevant for compression at this stage (cf. Figure A3). As motivated above, the score map allows us to globally rank all singular values based on their score. This leads to a fully independent compression stage where we can flexibly create a model of any reduced size: we prune as many singular values $\mathbf{s}_l^{(i)}$ (and the corresponding singular vectors) according to their scores $\boldsymbol{\rho}_l^{(i)}$ so that a given compression rate r is achieved. Note that there is a monotonic but non-linear relationship between the total number of pruned singular values k and compression rate r. Given a target rate r, we find k via a binary search (in almost real-time). As this compression procedure operates directly on the reparametrized model from Step 1, it is reversible and therefore indeed allows for Any Compression.

3. Experiments

We evaluate `ACIP` on a representative selection of open-weight models: LLaMA-7B/13B (Touvron et al., 2023a), LLaMA-2-7B/13B (Touvron et al., 2023b), LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen2.5-7B/14B (Qwen et al., 2024), and Mistral-7B-v0.3 (Jiang et al., 2023). We use a subset of C4 training data (Raffel et al., 2019) for the pruning stage. Regarding evaluation tasks, we follow Wang et al. (2024) and report perplexity on validation held-outs of C4 (Raffel et al., 2019) and WikiText-2 (Merity et al., 2017), and we consider seven zero-shot tasks from EleutherAI LM Evaluation Harness (LM-Eval) (Gao et al., 2023). Implementation details of `ACIP` can be found in Appendix D.

3.1. Analyzing Compression-Performance Trade-Offs

We first study compression-performance trade-offs powered by `ACIP`. Figures 1 and 2 demonstrate smooth and consistent curve shapes for all considered models; analogous results for WikiText-2 and individual zero-shot LM-Eval tasks can be found in Figure A4. A remarkable observation is that the oldest models, LLaMA-7B/13B, perform best perplexity-wise, while newer, more capable models like Qwen2.5-7B/14B dominate on LM-Eval as expected, especially on the lower compression levels. This apparent contradiction is likely caused by a deviation of the pre-training data distributions from C4 in the case of more recent models.

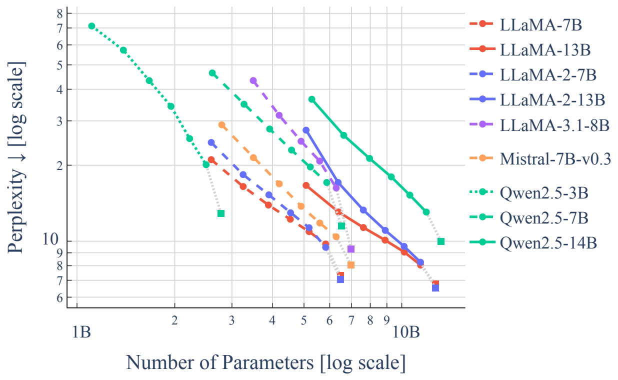


Figure 1: Compression-performance trade-offs generated by `ACIP` on **C4**. Each curve was obtained by the Any Compression stage (Step 3 in Section 2.3), i.e., no additional computation was required except for a perplexity evaluation. Square marks denote the base model performance.

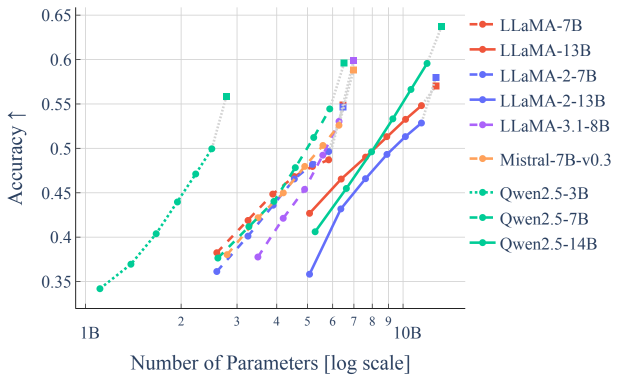


Figure 2: Compression-performance trade-off curves generated by `ACIP`, using average accuracy on all **LM-Eval** tasks as metric.

A second noteworthy outcome of Figures 1 and 2 are the gaps between LLMs of different base model sizes in the same family. Indeed, `ACIP` cannot match the performance of base models of smaller size, e.g., compare the compressed Qwen2.5-7B with the original Qwen2.5-3B. This is not surprising because the corresponding smaller-size base models were obtained by pre-training or knowledge distillation, which are orders of magnitudes more expensive than `ACIP`.

3.2. Comparison to Existing Works

In this section, we compare `ACIP` to two recent works focusing on SVD-based structured pruning, namely ASVD (Yuan et al., 2024) and SVD-LLM (Wang et al., 2024). Both approaches are backpropagation-free and perform (activation-aware) layer-wise updates instead. Table 1 shows that `ACIP` consistently outperforms both methods with a growing gap for higher compression levels. Note that SVD-LLM was calibrated on WikiText-2 instead of C4, which might ex-

Table 1: *Any Compression under SVD reparameterization.* Zero-shot evaluation of **LLaMA-7B**. Comparison with baselines ASVD (Yuan et al., 2024) and SVD-LLM (Wang et al., 2024). ↑: larger is better; ↓: smaller is better; best results for each task and size ratio are marked in **bold**. The scores for ASVD and SVD-LLM are taken from Wang et al. (2024).

Size	Method	C4 ↓	WikiText-2 ↓	Openb. ↑	ARC_e ↑	WinoG. ↑	HellaS. ↑	ARC_c ↑	PIQA ↑	MathQA ↑	LM Eval Avg. ↑
100%	Original	**7.34**	**5.68**	**0.28**	**0.67**	**0.67**	**0.56**	**0.38**	**0.78**	**0.27**	**0.52**
80%	ASVD	15.93	11.14	0.25	0.53	**0.64**	0.41	0.27	0.68	**0.24**	0.43
	SVD-LLM	15.84	**7.94**	0.22	0.58	0.63	0.43	0.29	0.69	**0.24**	0.44
	`ACIP` (ours)	**10.92**	8.83	**0.28**	**0.66**	0.63	**0.49**	**0.32**	**0.74**	0.23	**0.48**
70%	ASVD	41.00	51.00	0.18	0.43	0.53	0.37	0.25	0.65	0.21	0.38
	SVD-LLM	25.11	**9.56**	0.20	0.48	0.59	0.37	0.26	0.65	0.22	0.40
	`ACIP` (ours)	**12.22**	10.35	**0.28**	**0.64**	**0.62**	**0.47**	**0.31**	**0.73**	**0.23**	**0.47**
60%	ASVD	1109.00	1407.00	0.13	0.28	0.48	0.26	0.22	0.55	0.19	0.30
	SVD-LLM	49.83	13.11	0.19	0.42	0.58	0.33	0.25	0.60	0.21	0.37
	`ACIP` (ours)	**13.91**	**12.46**	**0.25**	**0.61**	**0.59**	**0.44**	**0.30**	**0.71**	**0.24**	**0.45**
50%	ASVD	27925.00	15358.00	0.12	0.26	0.51	0.26	0.22	0.52	0.19	0.30
	SVD-LLM	118.57	23.97	0.16	0.33	0.54	0.29	0.23	0.56	0.21	0.33
	`ACIP` (ours)	**16.47**	**16.16**	**0.21**	**0.57**	**0.57**	**0.40**	**0.27**	**0.68**	**0.22**	**0.42**
40%	ASVD	43036.00	57057.00	0.12	0.26	0.49	0.26	0.21	0.51	0.18	0.29
	SVD-LLM	246.89	42.30	0.14	0.28	0.50	0.27	0.22	0.55	0.21	0.31
	`ACIP` (ours)	**21.05**	**23.99**	**0.19**	**0.49**	**0.55**	**0.35**	**0.24**	**0.64**	**0.21**	**0.38**

plain slightly better results on the former dataset for 70% and 80% size. We think that these results underpin the benefits of an end-to-end scheme: (i) a simultaneous correction, e.g., by LoRA, can drastically improve performance, and (ii) robust pruning patterns can be found without leveraging any specific features of the SVD factorization. Moreover, note that re-computations are required to generate each row of Table 1 for ASVD and SVD-LLM, whereas `ACIP` only needs a single run. Analogous results for `ACIP` applied to all other models can be found in Table A2.

Further Experiments and Ablations Additional experiments and results can be found in Appendix E, demonstrating the effectiveness of fine-tuning (Appendix E.2) and quantization (Appendix E.3), analyzing the efficiency of `ACIP` (Appendix E.4), and studying the impact of several key components and design aspects (Appendix E.5–E.12).

4. The Current State of Any Compression

The two dominant approaches to efficient post-training model compression are quantization (reducing numerical precision) and (un-)structured weight pruning (removing redundant weights) (Zhu et al., 2024). However, these methods usually only permit a restricted set of compression rates — quantization due to fixed bit-length reductions and pruning due to $n{:}m$ structured sparsity required for efficient memory allocation. These structural constraints prevent Any Compression as described in Section 1.

In this work, we achieve Any Compression through the use of score maps to determine global parameter importance. Score maps have been used as a tool for model compression since the 1980s (LeCun et al., 1989; Hassibi et al., 1993). However, in the era of LLMs, deriving a score for each parameter poses significant challenges in terms of scalability. Existing approaches involve layer-wise scoring and the compression to preset factors (Sun et al., 2024; Wang et al., 2024; Yuan et al., 2024), while sacrificing the ability to choose different compression ratios without extra cost. In contrast, `ACIP` enables scalable Any Compression by (i) using weight factorization to significantly reduce the score map size (e.g., to 900k parameters for a 7B model), and (ii) decoupling the scoring and compression stages (Step 2 and 3 in Section 2, respectively). For a further discussion of related work, we refer to Appendix C.

5. Conclusion

We have introduced *Any Compression via Iterative Pruning* (`ACIP`), an end-to-end algorithm to determine the compression-performance trade-off of pre-trained models. The underlying score map ranking allows us to materialize models of any compression rate in real-time. We have demonstrated empirically that `ACIP` outperforms existing factorization approaches and that it can be combined with weight quantization (see Appendix E.3).

Discussion Our main results in Figures 1 and 2 resemble the well-known phenomenon of scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). Recently, it has been shown that any-size models can be achieved through pre-training (Devvrit et al., 2024; Gonzalez & Shivanna, 2025), exhibiting similar trade-offs as `ACIP`. Establishing a rigorous connection between these two fields of research would be an interesting avenue of future research.

A technical limitation of our work is that we have only focused on models that are tunable on a single (NVIDIA H100) GPU in `bf16`-precision. Hence, the scaling behavior of ACIP for larger LLMs (30B+) remains to be explored. Finally, a more detailed study of inference speed (beyond the results of Appendix E.4) could provide useful insights into the interplay of low-rank models and their efficiency.

Software and Data

The training and evaluation code can be found at https://github.com/merantix-momentum/acip.

Ready-to-use ACIP models can be found at https://huggingface.co/collections/MerantixMomentum/acip-67fe8f7b9f3132468a117ea6.

Acknowledgements

The authors thank Brennan Wilkerson, Ziyad Sheebaelhamd, and Timophey Molchanov for helpful discussions. We thank John Arnold and Jannis Klinkenberg for their technical cluster support.

We kindly acknowledge funding by the German Federal Ministry of Education and Research (BMBF) within the project “More-with-Less: Effiziente Sprachmodelle für KMUs” (grant no. 01IS23013A). Moreover, we thank WestAI for providing compute resources that were used to conduct the numerical experiments of this work.

References

Allen-Zhu, Z. and Li, Y. Physics of Language Models: Part 3.3, Knowledge Capacity Scaling Laws, 2024. arXiv:2404.05405 [cs].

Ashkboos, S., Croci, M. L., Nascimento, M. G. d., Hoefler, T., and Hensman, J. SliceGPT: Compress Large Language Models by Deleting Rows and Columns, 2024. arXiv:2401.15024 [cs].

Ben Noach, M. and Goldberg, Y. Compressing Pre-trained Language Models by Matrix Decomposition. In Wong, K.-F., Knight, K., and Wu, H. (eds.), *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 884–889. Association for Computational Linguistics, 2020.

Bengio, Y., Léonard, N., and Courville, A. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation, 2013. arXiv:1308.3432 [cs].

Boggust, A., Sivaraman, V., Assogba, Y., Ren, D., Moritz, D., and Hohman, F. Compress and Compare: Interactively Evaluating Efficiency and Behavior Across ML Model Compression Experiments. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):809–819, 2025.

Busbridge, D., Shidani, A., Weers, F., Ramapuram, J., Littwin, E., and Webb, R. Distillation scaling laws, 2025. arXiv:2502.08606 [cs].

Cover, T. M. and Thomas, J. A. *Elements of Information Theory*. Wiley-Interscience, Hoboken, N.J, 2nd edition, 2006.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10088–10115. Curran Associates, Inc., 2023.

Devvrit, Kudugunta, S., Kusupati, A., Dettmers, T., Chen, K., Dhillon, I., Tsvetkov, Y., Hajishirzi, H., Kakade, S., Farhadi, A., and Jain, P. Matformer: Nested transformer for elastic inference. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 140535–140564. Curran Associates, Inc., 2024.

Edalati, A., Tahaei, M., Rashid, A., Nia, V., Clark, J., and Rezagholizadeh, M. Kronecker Decomposition for GPT Compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 219–226. Association for Computational Linguistics, 2022.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. Least angle regression. *The Annals of Statistics*, 32(2), 2004.

Frantar, E. and Alistarh, D. SparseGPT: Massive Language Models Can Be Accurately Pruned in One-Shot, 2023. arXiv:2301.00774 [cs].

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac’h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 2023. tex.version: v0.4.0.

Gao, W., Liu, Y.-H., Wang, C., and Oh, S. Rate Distortion For Model Compression: From Theory To Practice. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 2102–2111. PMLR, 2019.

Gonzalez, L. and Shivanna, R. Announcing Gemma 3n preview: powerful, efficient, mobile-first AI, 2025. URL https://developers.googleblog.com/

en/introducing-gemma-3n/. Google Blog. Accessed 5/26/2025.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., De Paola, B., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The Llama 3 Herd of Models, 2024. arXiv:2407.21783 [cs].

Hassibi, B., Stork, D., and Wolff, G. Optimal Brain Surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pp. 293–299, 1993.

Hastie, T., Tibshirani, R., and Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall, 2015.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., Driessche, G. v. d., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training Compute-Optimal Large Language Models, 2022. arXiv:2203.15556 [cs].

Hohman, F., Kery, M. B., Ren, D., and Moritz, D. Model Compression in Practice: Lessons Learned from Practitioners Creating On-device Machine Learning Experiences. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–18. Association for Computing Machinery, 2024.

Hsu, Y.-C., Hua, T., Chang, S., Lou, Q., Shen, Y., and Jin, H. Language model compression with weighted low-rank factorization, 2022. arxiv:2207.00112 [cs].

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. In *ICLR*, 2022.

Idelbayev, Y. and Carreira-Perpiñán, M. A. Low-Rank Compression of Neural Nets: Learning the Rank of Each Layer. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8046–8056, 2020.

Isik, B., Weissman, T., and No, A. An Information-Theoretic Justification for Model Pruning. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pp. 3821–3846. PMLR, 2022.

Jaiswal, A., Yin, L., Zhang, Z., Liu, S., Zhao, J., Tian, Y., and Wang, Z. From GaLore to WeLore: How Low-Rank Weights Non-uniformly Emerge from Low-Rank Gradients, 2024. arXiv:2407.11239 [cs].

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7B, 2023. arXiv:2310.06825 [cs].

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling Laws for Neural Language Models, 2020. arXiv:2001.08361 [cs].

Kaushal, A., Vaidhya, T., and Rish, I. LORD: Low Rank Decomposition Of Monolingual Code LLMs For One-Shot Compression, 2023. arXiv:2309.14021 [cs].

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *ICLR*, 2015.

LeCun, Y., Denker, J., and Solla, S. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989.

Mairal, J. and Yu, B. Complexity analysis of the lasso regularization path. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1835–1842. Omnipress, 2012.

Merity, S., Xiong, C., Bradbury, J., and Socher, R. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Mirsky, L. Symmetric Gauge Functions and Unitarily Invariant Norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: an imperative style, high-performance deep learning library. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, number 721, pp. 8026–8037. Curran Associates Inc., 2019.

Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 Technical Report, 2024. arxiv:2412.15115 [cs].

Raffel, C., Shazeer, N. M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 2019.

Raschka, S. New LLM pre-training and post-training paradigms, 2024. Blog post. Accessed 1/31/2025.

Sharma, P., Ash, J. T., and Misra, D. The Truth is in There: Improving Reasoning in Language Models with Layer-Selective Rank Reduction. In *ICLR*, 2023.

Sun, M., Liu, Z., Bair, A., and Kolter, J. Z. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Tahaei, M., Charlaix, E., Nia, V., Ghodsi, A., and Rezagholizadeh, M. KroneckerBERT: Significant Compression of Pre-trained Language Models Through Kronecker Decomposition and Knowledge Distillation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2116–2127, Seattle, United States, 2022. Association for Computational Linguistics.

Tibshirani, R. Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., 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. arXiv:2302.13971 [cs].

Touvron, H., Martin, L., Stone, K. R., 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. M., Korenev, A., Koura, P. S., Lachaux, M., 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., 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. arXiv:2307.09288 [cs].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Wang, X., Zheng, Y., Wan, Z., and Zhang, M. SVD-LLM: Truncation-aware Singular Value Decomposition for Large Language Model Compression, 2024. arXiv:2403.07378 [cs].

Xia, M., Gao, T., Zeng, Z., and Chen, D. Sheared LLaMA: Accelerating Language Model Pre-training via Structured Pruning, 2024. arXiv:2310.06694 [cs].

Yin, P., Lyu, J., Zhang, S., Osher, S., Qi, Y., and Xin, J. Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets. In *International Conference on Learning Representations*, 2018.

Yu, H. and Wu, J. Compressing Transformers: Features Are Low-Rank, but Weights Are Not! *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11007–11015, 2023. Number: 9.

Yu, M., Wang, D., Shan, Q., Reed, C., and Wan, A. The Super Weight in Large Language Models, 2024. arXiv:2411.07191 [cs].

Yuan, Z., Shang, Y., Song, Y., Wu, Q., Yan, Y., and Sun, G. ASVD: Activation-aware Singular Value Decomposition for Compressing Large Language Models, 2024. arXiv:2312.05821 [cs].

Zhu, X., Li, J., Liu, Y., Ma, C., and Wang, W. A Survey on Model Compression for Large Language Models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

A. Question & Answers

In this section, we discuss a few common questions about our method and experimental design that were not (fully) addressed in the main part due to length restrictions.

Q1. *Why did you not directly compare your results to quantization and full-weight (unstructured) pruning?*

A1. We argue that these are fundamentally different compression approaches. Full weight manipulations, in principle, have the potential to lead to more powerful compressions because they have more degrees of freedom (analogously to full-weight fine-tuning vs. PEFT). Therefore, they should not be seen as competing methods but complementary ones. We admit that practitioners probably would not favor `ACIP` over well-established and widely supported quantization techniques. However, the adapter-style nature of `ACIP` makes it suitable for a combination which can lead to extra gains, as demonstrated in Appendix E.3.

Q2. *Why did you not compare with model distillation or combined `ACIP` with it?*

A2. While model distillation can lead to outstanding compression results, e.g., see (Busbridge et al., 2025; Raschka, 2024), this approach requires significantly more resources than `ACIP`, typically orders of magnitudes more. A direct comparison is therefore not meaningful from our point of view, as it should at least be based on approximately the same computational budget.

Q3. *Why do you propose a backpropagation-based algorithm instead of layer-wise weight updates?*

A3. Let us first summarize several benefits of our end-to-end optimization approach from the main paper: (i) it is conceptually simple and requires no feature engineering, (ii) an error correction can be injected with almost no extra costs, (iii) it allows us to perform efficient and accurate Any Compression.

Apart from that, and to the best of our knowledge, existing compression algorithms that use layer-wise updates like ASVD (Yuan et al., 2024), SVD-LLM (Wang et al., 2024), or WeLore (Jaiswal et al., 2024) require a separate fine-tuning step to achieve competitive downstream performance at stronger compression ratios. Therefore, the lower costs of layer-wise compression are actually dominated by a more expensive backpropagation-based step. It remains open if similar results can be obtained by a fully tuning-free algorithm.

Q4. *Why do you use matrix factorization, and SVD in particular?*

A4. Committing to a backpropagation-based algorithm (see Q3) means that we have to deal with increased memory requirements. As such, matrix factorization is not helpful in that respect because the number of parameters might even increase initially (for instance, an SVD-parametrization basically doubles the size of a quadratic weight matrix). On the other hand, tuning and pruning only the bottleneck layer (i.e., the singular value mask in case of `ACIP`) has the potential for drastic size reductions and is highly parameter-efficient. For example, the number of tunable mask parameters for LLaMA-7B with `ACIP` is $<$1M.

With this in mind, SVD as a specific matrix factorization is an obvious candidate due to its beneficial mathematical and numerical properties, in particular, optimal low-rank matrix approximation and stable matrix operations due to orthogonality.

B. Schematic Overview of ACIP

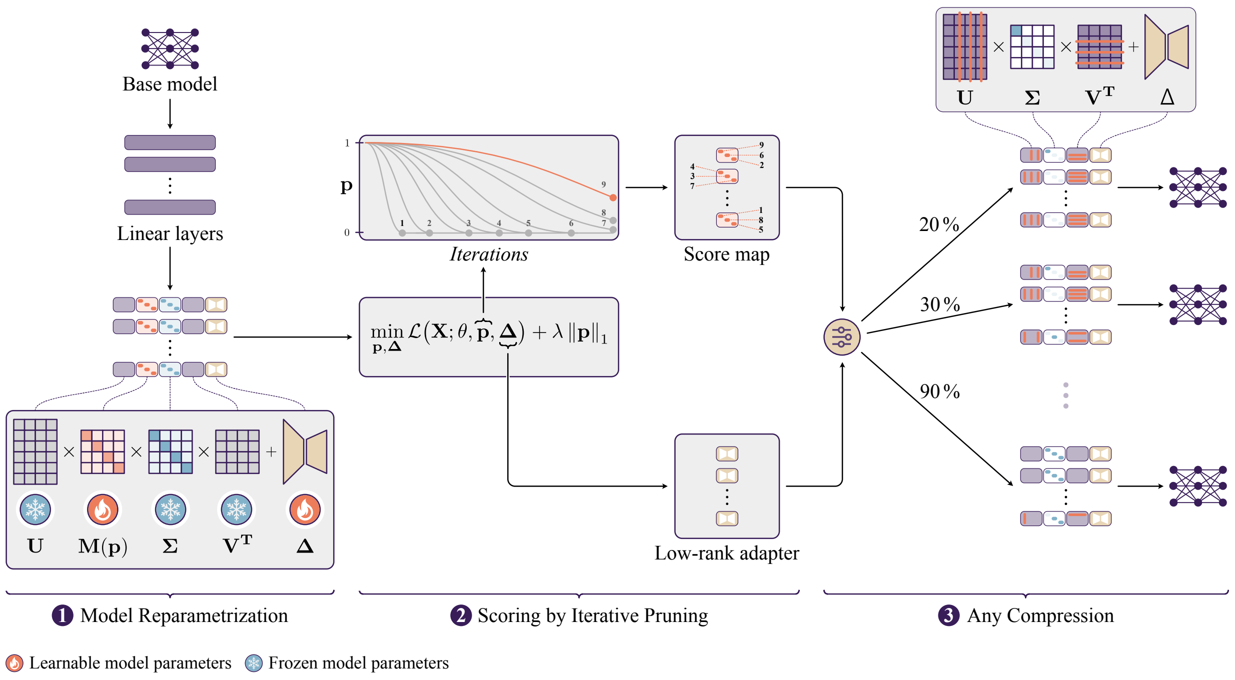

Base model
Linear layers
U
M(p)
Σ
V^T
Δ
p
Iterations
$\min_{\mathbf{p},\mathbf{\Delta}} \mathcal{L}(\mathbf{X};\theta,\mathbf{p},\mathbf{\Delta}) + \lambda\|\mathbf{p}\|_1$
Score map
Low-rank adapter
20 %
30 %
90 %
1 Model Reparametrization
2 Scoring by Iterative Pruning
3 Any Compression
Learnable model parameters
Frozen model parameters

Figure A3: A visual overview of ACIP. ❶ The linear layers of the base model are reparametrized in terms of their singular value decomposition $\mathbf{U}\mathbf{M}\mathbf{\Sigma}\mathbf{V}^\top$, with a (binary) singular value mask $\mathbf{M} = \mathbf{M}(\mathbf{p})$ and a low-rank adapter $\mathbf{\Delta}$. ❷ The objective function is optimized over the mask parameters $\mathbf{p}$ and adapters $\mathbf{\Delta}$, where sparsity is induced on $\mathbf{p}$ by an ℓ_1-penality leading to pruned entries in the mask $\mathbf{M}(\mathbf{p})$. The optimization path of $\mathbf{p}$ gives rise to a score map that determines the global importance of the singular values across the full model. Potential compression errors are compensated by $\mathbf{\Delta}$. ❸ Based on the parameter scores, the base model can be flexibly compressed to any target size by masking the entries of $\mathbf{\Sigma}$. The learned adapters $\mathbf{\Delta}$ are used as correction for any compression level.

C. Related work

Post-training compression methods for Foundation Models largely fall into three categories: knowledge distillation, quantization and (un-)structured pruning. For a comprehensive survey covering all of these, we refer to (Zhu et al., 2024) and the references therein.

`ACIP` represents a specific form of structured pruning: factorization-based compression. Conventional structured pruning (Frantar & Alistarh, 2023; Xia et al., 2024; Ashkboos et al., 2024) jointly removes groups of parameters (e.g., network layers or matrix columns/rows). At high compression rates, however, such coarse approaches often remove important weights, causing a significant performance drop that is only recoverable through additional fine-tuning.

Low-Rank Decomposition Approximation of weight matrices through appropriate factorization aims to preserve critical information while being simple to implement. After initial exploration for smaller language models (Edalati et al., 2022; Tahaei et al., 2022), methods for LLMs primarily built on (weighted) SVD of linear layers (Ben Noach & Goldberg, 2020; Hsu et al., 2022). Large approximation errors, however, made additional fine-tuning on down-stream tasks necessary.

Follow-up work recognized that the poor approximations are caused by LLM weights being high-rank and instead turned to decompose network features which are sparse (Kaushal et al., 2023; Yu & Wu, 2023). Notably, ASVD (Yuan et al., 2024) explicitly accounts for the data distribution eliciting these activations and SVD-LLM (Wang et al., 2024) derives an analytical layer-wise correction.

Recent studies (Sharma et al., 2023; Yuan et al., 2024; Jaiswal et al., 2024) have shown rank reduction to differently affect layers in a network and proposed different heuristics for non-uniform pruning of singular values.

A common feature of these works is that they first truncate singular values to a desired size before computing an error correction. A change in compression ratio therefore prompts another round of computation. At the same time, the correction quality varies across different ratios, so that milder compression may not automatically preserve more predictive quality. In contrast, our approach creates consistent compression trade-offs from a single round of computation (optimization) and performs corrections on-the-fly.

Rate-distortion theory Rate-distortion theory investigates the analytical trade-off between achievable data compression rates and the error (distortion) introduced by general lossy compression algorithms (Cover & Thomas, 2006). While some recent work (Gao et al., 2019; Isik et al., 2022) investigates rate-distortion theory of machine learning models for simple architectures under rather specific assumptions, the information-theoretic limits of neural network compression are generally unknown in practically relevant settings. In this context, the family of compressed models generated by `ACIP` conveniently provides an empirical (upper) bound on the distortion-rate function of a large-scale model from a single optimization run.

D. Implementation Details

In this section, we report more technical details and hyperparameters used for our experiments.

Dataset and Models Following previous work on LLM compression, we use C4 (Raffel et al., 2019) for training as it is a good proxy of a general-purpose dataset. In the context of ACIP, it should be primarily seen as a calibration dataset that allows us to propagate meaningful activations through a pre-trained model while performing structured pruning. Overfitting to the distribution of C4 is implicitly mitigated, since we only tune very few parameters (masks and LoRA-adapters) compared to the total model size. As loss function $\mathcal{L}$ in (4), we use the standard negative log-likelihood loss for next-token prediction.

All considered (evaluation) datasets and pre-trained models are imported with the HuggingFace `transformers`-library in `bfloat16`-precision. Our experiments were implemented with `PyTorch` (Paszke et al., 2019) and the `Lightning` package.

ACIP-Specifics As mentioned in Remark 2.1, we apply a linear scheduler that increases the regularization parameter λ dynamically over the pruning process. This ensures that the pruning becomes more and more aggressive over time and the stopping criterion will be reached at some point. Across all experiments, we use $\lambda = 1e{-3}$ as initial value and increase it by a factor of 1.01 every 4 steps (this amounts to a doubling of λ at about every 280 steps).

As pointed out in Section 2.2, we choose a (maximum) target compression rate as a stopping criterion for ACIP. In most experiments, a rate of $r_{\text{stop}} = 0.4$ is reasonable (i.e., only 40% or the original parameters remain), and we refer to Appendix E.6 for further discussion and analysis. After the stopping criterion is reached, we tune the low-rank adapter for 1k more steps while the masks are frozen (see Section 2.2).

The mask parameters in (3) are rescaled by a fixed factor of 0.02 to ensure a better alignment with the numerical range of the remaining network weights. The low-rank adapters are created with $r = 32$, $\alpha = 16$, and dropout 0.05. For LLaMA-7B, the number of tunable parameters amounts to <1M mask parameters and approximately 80M low-rank adapter parameters.

For sample data from C4, we use 1024 tokens per sample and a batch size of 4. We use Adam (Kingma & Ba, 2015) as optimizer without weight decay and a learning rate of $5e{-5}$.

Runtime Analysis ACIP requires significantly fewer steps than fine-tuning. Depending on when the stopping criterion is reached, it typically takes 1.5k - 2.5k steps, including 1k post-tuning steps of the low-rank adapters. For LLaMA-7B, for example, this amounts to a wall clock runtime of < 30 minutes, including the initial SVD computations for the base model parametrization. All runs were performed on single NVIDIA H100 GPUs. See also Appendix E.4 for a more detailed efficiency analysis.

Fine-Tuning In all post-compression fine-tunings (see Appendix E.2), we simply continue training ACIP's low-rank adapters (the optimizer states are reset). We train for 25k steps on C4 with a batch size of 4 and a learning rate of $2e{-4}$.

E. Further Experiments and Ablations

This section presents several additional experiments and results to (i) demonstrate the effectiveness of fine-tuning (Appendix E.2) and quantization (Appendix E.3), (ii) analyze the efficiency of ACIP (Appendix E.4), and (iii) study the impact of several key components and design aspects (Appendix E.5–E.12). Note that the most detailed analyses and ablations are carried out with LLaMA-7B as it was most extensively studied in previous research on structured weight pruning.

E.1. Supplementary Results for Section 3.1 (Analyzing Compression-Performance Trade-Offs)

Figure A4 complements the trade-off curves in Figures 1 and 2 by all other considered evaluation metrics (see Section 3.1).

E.2. Improving Performance Through Fine-Tuning

While the main goal of this work is to produce a full family of accurate, compressed models from a few optimization steps, their performance can be certainly improved through continued fine-tuning. Figure A5 highlights the gains of fine-tuning LLaMA-7B; see Table A2 for more detailed numerical results on all other models. We observe that fine-tuning leads to a performance offset that is almost constant across all compression levels, which underlines the predictive capacity of ACIP. Note that we even observe a jump at zero compression because inserting the low-rank adapters learned by ACIP leads to a slight initial performance drop.

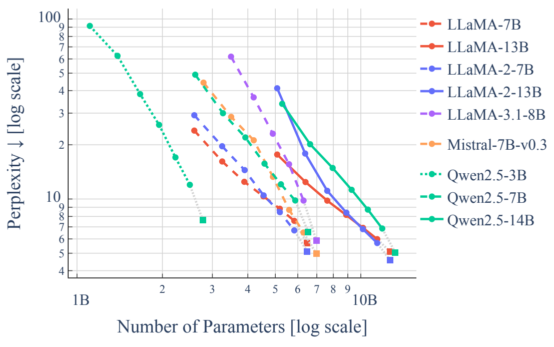
An optional fine-tuning step is not exclusive to ACIP but can be applied to many other compression approaches as well. Table A3 provides a comparison with ASVD (Yuan et al., 2024) and SVD-LLM (Wang et al., 2024) (cf. Section 3.2) when fine-tuned with LoRA. While ACIP still performs best in this respect, we argue that post-compression fine-tuning should be still seen as an independent (and much more costly) algorithmic step for two reasons. (i) Its outcome strongly depends on the specific training protocol and data, making a fair and direct comparison challenging; (ii) it requires us to fix a compression level, which breaks the crucial Any Compression feature of ACIP. Therefore, promoting a costly fine-tuning step after compression is not the primary concern of our work.

E.3. Combining ACIP with Quantization

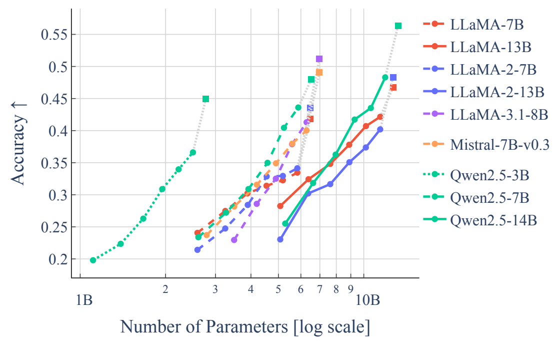
In the field of low-cost compression for LLMs, quantization is still considered as the gold standard (Hohman et al., 2024; Zhu et al., 2024), so that a practitioner might not be willing to exchange its gains for the benefits of ACIP. Fortunately, ACIP only tunes a tiny fraction of weights with high precision, so that all remaining modules are suitable for quantization. In our experiments, we quantize all parameterized and unparametrized linear layers to 4-bit in fp4-format (Dettmers et al., 2023) using the bitsandbytes-Package (W4A16), except for the embedding layer and final classification head. We study the gains of quantization for ACIP in the following two ways.

Compress first, then quantize We first apply ACIP as usual, compress the model to a given target size, and then quantize all remaining linear layers. Figure A5 confirms that this approach works fairly well, only producing a slight performance drop compared to non-quantized versions; see Table A4 for a full evaluation on all other metrics. We also observe that an optional fine-tuning step as in Appendix E.2 can almost fully compensate for the errors introduced by quantization after compression. This finding is well in line with the effectiveness of the popular QLoRA approach (Dettmers et al., 2023). Moreover, Figure A5 reveals a drastic improvement through quantization in terms of required memory. Here, the ACIP-trade-off allows practitioners to study and apply a more fine-grained compression on top of quantization.

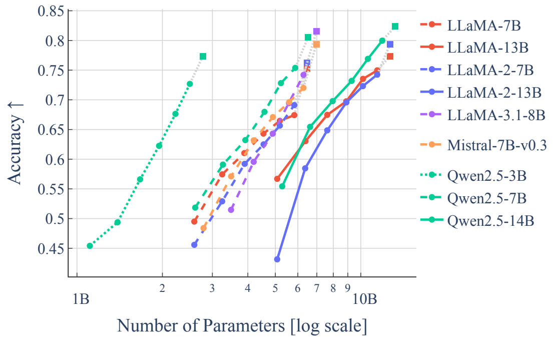
Quantize first, then compress and transfer Compared to layer-wise methods like ASVD and SVD-LLM, ACIP has a higher demand in GPU memory due to backpropagation. A quantization of all frozen weight matrices can be an effective remedy in this respect. For the experiment shown in Figure A6, we have applied quantization before ACIP, which leads to very similar compression-performance trade-offs as in the non-quantized case. Going one step further, we transfer the score maps and low-rank adapters from this quantized version of ACIP back to full precision: We load the base model in bf16, apply layer-wise SVD-parametrization, insert the low-rank adapters learned by quantized ACIP, and use the corresponding score map to obtain a compressed model (W16A16). The resulting trade-off curve in Figure A6 confirms that this simple strategy works fairly well, especially for lower compression levels.



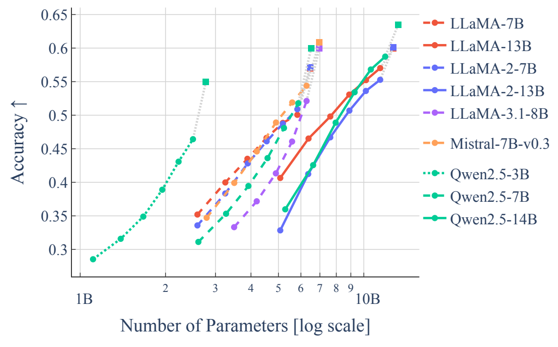
(a) WikiText-2



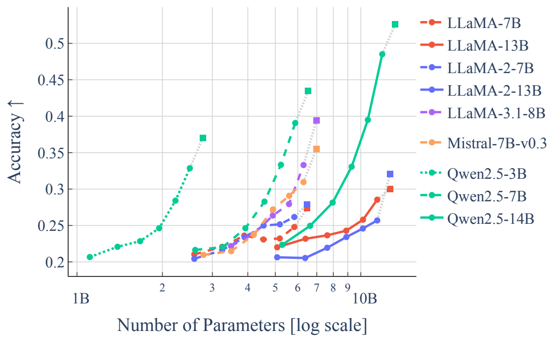
(b) ARC Challenge



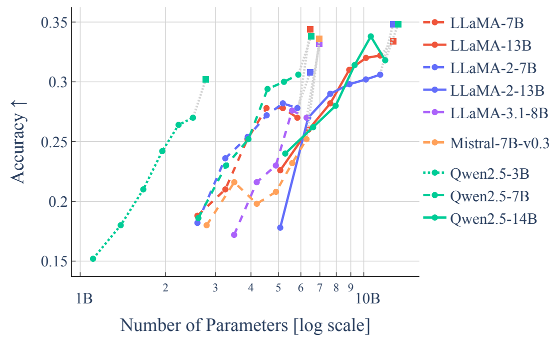
(c) ARC Easy



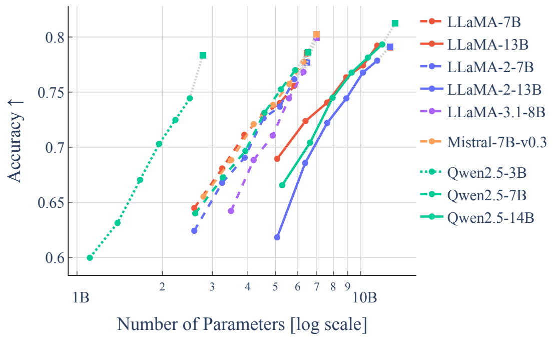
(d) HellaSwag



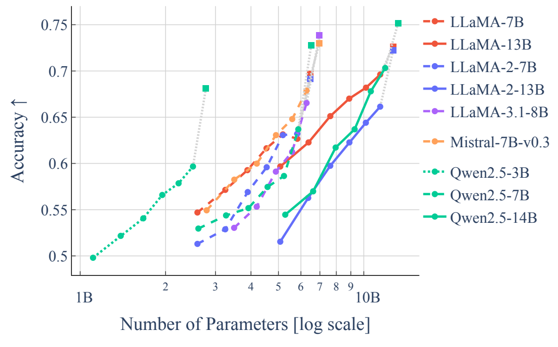
(e) MathQA



(f) OpenBookQA



(g) PIQA



(h) Winogrande

Figure A4: Compression-performance trade-off curves generated by `ACIP` on **WikiText-2** and **individual LM-Eval tasks**, complementing the results of Figures 1 and 2.

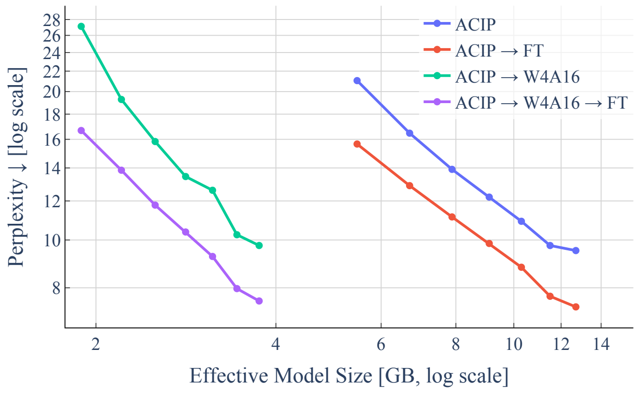


Figure A5: Compression-performance trade-off curves for **LLaMA-7B** on **C4** showing the impact of **fine-tuning** and **quantization** *after* compression with `ACIP`. The horizontal axis measures size in terms of required (weight) memory to visualize the gains of quantization more clearly.

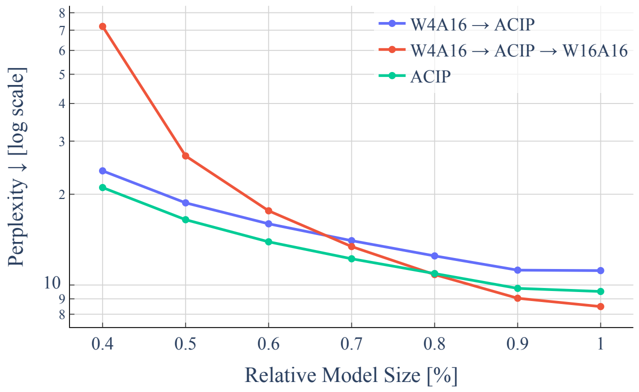


Figure A6: Compression-performance trade-off curves for **LLaMA-7B** on **C4**, showing that **quantization** *before* `ACIP` leads to similar results as without.

Table A2: Evaluation results for `ACIP` on all considered LLMs. Scores on C4 and WikiText-2 are measured in perplexity (smaller is better), and the LM-Eval zero-shot tasks are measured in accuracy (higher is better). *The results of LLaMA2-13B were achieved by ignoring all up-projection layers in `ACIP` (see Appendix E.10 for more details).

		C4 ↓		WikiText-2 ↓		ARC_c ↑		ARC_e ↑		HellaS. ↑		MathQA ↑		Openb. ↑		PIQA ↑		WinoG. ↑		LM Eval Avg. ↑	
Model	Type Size	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT	ACIP	FT
LLaMA-7B	40%	21.05	15.66	23.99	17.33	0.24	0.24	0.49	0.53	0.35	0.38	0.21	0.21	0.19	0.20	0.64	0.67	0.55	0.57	0.38	0.40
	50%	16.47	12.89	16.16	11.88	0.27	0.27	0.57	0.59	0.40	0.43	0.22	0.23	0.21	0.23	0.68	0.71	0.57	0.60	0.42	0.44
	60%	13.91	11.14	12.46	9.63	0.30	0.32	0.61	0.64	0.44	0.47	0.24	0.23	0.25	0.26	0.71	0.73	0.59	0.63	0.45	0.47
	70%	12.22	9.84	10.35	8.27	0.31	0.34	0.64	0.69	0.47	0.50	0.23	0.24	0.28	0.28	0.73	0.75	0.62	0.66	0.47	0.49
	80%	10.92	8.81	8.83	7.19	0.32	0.38	0.66	0.71	0.49	0.53	0.23	0.24	0.28	0.31	0.74	0.77	0.63	0.68	0.48	0.52
	90%	9.75	7.69	7.56	6.12	0.33	0.39	0.67	0.73	0.50	0.56	0.25	0.25	0.27	0.33	0.76	0.78	0.63	0.69	0.49	0.53
	100%	9.52	7.32	7.20	5.75	0.33	0.40	0.69	0.75	0.51	0.57	0.25	0.26	0.26	0.34	0.76	0.79	0.63	0.70	0.49	0.54
	Original	7.31		5.68		0.42		0.75		0.57		0.27		0.34		0.79		0.70		0.55	
LLaMA-13B	40%	16.64	13.38	17.66	13.42	0.28	0.28	0.57	0.59	0.41	0.43	0.22	0.23	0.23	0.24	0.69	0.70	0.60	0.62	0.43	0.44
	50%	13.06	11.12	12.42	10.49	0.32	0.34	0.63	0.63	0.47	0.48	0.23	0.23	0.26	0.28	0.72	0.73	0.62	0.64	0.47	0.47
	60%	11.33	9.76	9.79	8.30	0.35	0.33	0.67	0.68	0.50	0.51	0.24	0.24	0.28	0.30	0.74	0.76	0.65	0.67	0.49	0.50
	70%	10.10	8.72	8.17	7.06	0.38	0.38	0.70	0.69	0.53	0.54	0.24	0.26	0.31	0.31	0.76	0.77	0.67	0.70	0.51	0.52
	80%	9.06	7.91	6.91	6.21	0.41	0.41	0.74	0.74	0.55	0.57	0.26	0.27	0.32	0.34	0.77	0.78	0.68	0.69	0.53	0.54
	90%	8.04	7.06	5.98	5.40	0.42	0.44	0.75	0.76	0.57	0.59	0.29	0.29	0.32	0.33	0.79	0.79	0.70	0.72	0.55	0.56
	100%	7.86	6.79	5.83	5.15	0.42	0.46	0.75	0.77	0.57	0.60	0.29	0.30	0.31	0.32	0.79	0.79	0.70	0.73	0.55	0.57
	Original	6.77		5.09		0.47		0.77		0.60		0.30		0.33		0.79		0.73		0.57	
LLaMA-2-7B	40%	24.62	16.74	29.20	18.00	0.21	0.22	0.46	0.50	0.34	0.37	0.20	0.21	0.18	0.19	0.62	0.66	0.51	0.55	0.36	0.39
	50%	18.36	13.32	19.64	12.25	0.25	0.27	0.53	0.57	0.38	0.42	0.22	0.22	0.24	0.23	0.67	0.69	0.53	0.56	0.40	0.42
	60%	15.27	11.15	14.47	9.54	0.28	0.31	0.59	0.63	0.43	0.46	0.23	0.24	0.25	0.23	0.69	0.72	0.57	0.62	0.44	0.46
	70%	12.96	9.73	10.47	7.74	0.33	0.35	0.62	0.68	0.46	0.50	0.25	0.24	0.27	0.26	0.73	0.74	0.60	0.64	0.47	0.49
	80%	11.31	8.63	8.46	6.54	0.33	0.37	0.66	0.70	0.49	0.53	0.25	0.26	0.28	0.31	0.74	0.77	0.63	0.66	0.48	0.51
	90%	9.46	7.43	6.69	5.45	0.34	0.43	0.69	0.75	0.51	0.56	0.26	0.28	0.28	0.33	0.76	0.78	0.63	0.69	0.50	0.54
	100%	9.34	7.06	6.54	5.13	0.34	0.43	0.70	0.76	0.51	0.57	0.26	0.28	0.27	0.32	0.76	0.78	0.64	0.69	0.50	0.55
	Original	7.04		5.11		0.44		0.76		0.57		0.28		0.31		0.78		0.69		0.55	
LLaMA-2-13B	40%	27.55	84.28	41.22	145.79	0.23	0.23	0.43	0.46	0.33	0.30	0.21	0.21	0.18	0.18	0.62	0.63	0.52	0.52	0.36	0.36
	50%	17.10	12.76	17.89	13.12	0.30	0.32	0.58	0.61	0.41	0.44	0.21	0.23	0.27	0.27	0.69	0.70	0.56	0.58	0.43	0.45
	60%	13.29	10.05	11.11	8.43	0.32	0.36	0.65	0.68	0.47	0.50	0.22	0.23	0.29	0.31	0.72	0.74	0.60	0.63	0.47	0.49
	70%	11.04	8.64	8.40	6.71	0.35	0.41	0.70	0.74	0.51	0.54	0.23	0.25	0.30	0.33	0.74	0.77	0.62	0.66	0.49	0.53
	80%	9.54	7.68	6.80	5.66	0.37	0.44	0.72	0.76	0.54	0.57	0.25	0.28	0.30	0.34	0.77	0.78	0.64	0.71	0.51	0.55
	90%	8.26	6.86	5.70	4.87	0.40	0.47	0.74	0.78	0.55	0.60	0.26	0.31	0.31	0.34	0.78	0.79	0.66	0.72	0.53	0.57
	100%	7.87	6.56	5.42	4.61	0.41	0.47	0.75	0.79	0.56	0.60	0.28	0.32	0.31	0.35	0.78	0.79	0.69	0.71	0.54	0.58
	Original	6.52		4.57		0.48		0.79		0.60		0.32		0.35		0.79		0.72		0.58	
LLaMA-3.1-8B	50%	43.32	26.52	61.77	29.52	0.23	0.27	0.51	0.58	0.33	0.37	0.22	0.23	0.17	0.19	0.64	0.68	0.53	0.54	0.38	0.41
	60%	31.55	21.00	36.69	19.26	0.29	0.29	0.60	0.61	0.37	0.42	0.24	0.24	0.22	0.23	0.69	0.72	0.55	0.56	0.42	0.44
	70%	24.90	17.08	23.06	13.55	0.33	0.32	0.64	0.66	0.41	0.47	0.26	0.27	0.23	0.27	0.71	0.73	0.59	0.62	0.45	0.48
	80%	20.78	14.21	15.60	10.12	0.38	0.40	0.69	0.72	0.46	0.51	0.28	0.30	0.28	0.29	0.74	0.77	0.61	0.66	0.49	0.52
	90%	16.25	11.28	9.80	7.24	0.41	0.48	0.74	0.80	0.52	0.57	0.33	0.35	0.27	0.31	0.77	0.79	0.67	0.71	0.53	0.57
	100%	14.57	9.42	8.04	5.95	0.40	0.51	0.75	0.82	0.53	0.60	0.36	0.40	0.27	0.34	0.78	0.80	0.66	0.74	0.54	0.60
	Original	9.31		5.86		0.51		0.82		0.60		0.39		0.33		0.80		0.74		0.60	
Mistral-7B-v0.3	40%	28.92	19.21	44.29	23.24	0.24	0.26	0.48	0.53	0.35	0.38	0.21	0.21	0.18	0.19	0.66	0.67	0.55	0.57	0.38	0.40
	50%	21.44	14.86	28.60	16.53	0.28	0.28	0.57	0.59	0.40	0.43	0.21	0.23	0.22	0.21	0.69	0.70	0.58	0.60	0.42	0.43
	60%	16.89	12.49	21.19	12.29	0.32	0.32	0.63	0.66	0.45	0.48	0.24	0.24	0.20	0.23	0.72	0.73	0.60	0.62	0.45	0.47
	70%	13.75	10.95	13.28	9.69	0.35	0.34	0.67	0.68	0.49	0.52	0.27	0.28	0.21	0.26	0.74	0.76	0.63	0.63	0.48	0.49
	80%	11.80	9.84	8.70	7.49	0.38	0.39	0.70	0.73	0.52	0.55	0.29	0.29	0.23	0.26	0.76	0.77	0.65	0.69	0.50	0.53
	90%	10.42	8.84	6.51	5.85	0.40	0.43	0.72	0.75	0.54	0.59	0.31	0.33	0.25	0.31	0.78	0.79	0.68	0.70	0.53	0.56
	100%	9.85	8.31	6.04	5.31	0.40	0.45	0.73	0.77	0.55	0.60	0.33	0.34	0.26	0.33	0.78	0.79	0.68	0.72	0.53	0.57
	Original	8.05		4.96		0.49		0.79		0.61		0.36		0.34		0.80		0.73		0.59	
Qwen2.5-3B	40%	71.23	36.85	91.51	39.44	0.20	0.22	0.45	0.51	0.29	0.31	0.21	0.22	0.15	0.16	0.60	0.64	0.50	0.52	0.34	0.37
	50%	57.17	29.43	62.42	26.92	0.22	0.25	0.49	0.57	0.32	0.34	0.22	0.21	0.18	0.19	0.63	0.67	0.52	0.53	0.37	0.39
	60%	43.30	23.30	38.26	18.38	0.26	0.29	0.57	0.63	0.35	0.39	0.23	0.23	0.21	0.24	0.67	0.69	0.54	0.56	0.40	0.43
	70%	34.24	19.04	25.81	13.50	0.31	0.34	0.62	0.68	0.39	0.43	0.25	0.26	0.24	0.26	0.70	0.72	0.57	0.58	0.44	0.47
	80%	25.50	16.16	17.02	10.68	0.34	0.36	0.68	0.73	0.43	0.47	0.28	0.31	0.26	0.24	0.72	0.74	0.58	0.57	0.47	0.49
	90%	20.10	13.82	11.99	8.57	0.37	0.42	0.73	0.77	0.46	0.52	0.33	0.36	0.27	0.33	0.74	0.78	0.60	0.67	0.50	0.55
	100%	18.73	12.81	10.63	7.70	0.36	0.47	0.72	0.78	0.46	0.54	0.33	0.41	0.24	0.31	0.74	0.78	0.60	0.69	0.49	0.57
	Original	12.90		7.64		0.45		0.77		0.55		0.37		0.30		0.78		0.68		0.56	
Qwen2.5-7B	40%	46.43	29.26	49.04	27.24	0.23	0.24	0.52	0.58	0.31	0.34	0.22	0.21	0.19	0.20	0.64	0.67	0.53	0.53	0.38	0.40
	50%	34.90	23.26	29.96	19.72	0.27	0.30	0.59	0.64	0.35	0.39	0.22	0.23	0.23	0.25	0.67	0.70	0.54	0.57	0.41	0.44
	60%	27.84	18.73	21.98	13.73	0.31	0.33	0.63	0.68	0.39	0.45	0.25	0.26	0.25	0.28	0.70	0.73	0.55	0.60	0.44	0.48
	70%	22.97	15.96	15.72	10.71	0.35	0.42	0.68	0.74	0.44	0.49	0.28	0.30	0.29	0.30	0.73	0.76	0.57	0.63	0.48	0.52
	80%	19.68	14.02	12.07	8.89	0.40	0.46	0.73	0.78	0.48	0.53	0.33	0.36	0.30	0.33	0.75	0.78	0.59	0.67	0.51	0.56
	90%	17.09	12.59	9.82	7.63	0.44	0.48	0.75	0.80	0.52	0.56	0.39	0.41	0.31	0.32	0.77	0.79	0.64	0.70	0.54	0.58
	100%	15.34	11.43	8.38	6.60	0.43	0.50	0.75	0.82	0.53	0.59	0.43	0.46	0.28	0.34	0.77	0.79	0.66	0.72	0.55	0.60
	Original	11.47		6.55		0.48		0.81		0.60		0.43		0.34		0.79		0.73		0.60	
Qwen2.5-14B	40%	36.51	25.58	33.78	22.22	0.26	0.29	0.55	0.61	0.36	0.38	0.22	0.23	0.24	0.26	0.67	0.68	0.54	0.57	0.41	0.43
	50%	26.27	19.53	20.15	14.57	0.32	0.33	0.65	0.68	0.43	0.44	0.25	0.25	0.26	0.27	0.70	0.71	0.57	0.59	0.45	0.47
	60%	21.29	15.63	14.87	10.48	0.36	0.40	0.70	0.73	0.49	0.51	0.28	0.32	0.28	0.31	0.74	0.76	0.62	0.66	0.50	0.53
	70%	17.99	13.99	11.25	8.89	0.42	0.43	0.73	0.76	0.53	0.54	0.33	0.36	0.31	0.32	0.77	0.78	0.64	0.68	0.53	0.55
	80%	15.23	11.97	8.73	7.11	0.44	0.48	0.77	0.78	0.57	0.58	0.39	0.44	0.34	0.36	0.78	0.79	0.68	0.73	0.57	0.60
	90%	13.05	10.77	6.86	6.04	0.48	0.52	0.80	0.82	0.59	0.60	0.49	0.49	0.32	0.35	0.79	0.81	0.70	0.74	0.60	0.62
	100%	12.37	9.98	6.23	5.11	0.47	0.53	0.79	0.82	0.58	0.62	0.51	0.53	0.32	0.35	0.79	0.81	0.73	0.77	0.60	0.63
	Original	9.99		5.05		0.56		0.82		0.63		0.53		0.35		0.81		0.75		0.64	

Table A3: Evaluation of **LLaMA-7B** on **WikiText-2** (perplexity, smaller is better) under different compression ratios, with and without post-training fine-tuning. We compare ACIP with the existing SVD-based compression methods ASVD (Yuan et al., 2024) and SVD-LLM (Wang et al., 2024), see also Section 3.2. The scores for ASVD and SVD-LLM are taken from Wang et al. (2024, Table 4). Note that ACIP was fine-tuned on C4, while ASVD and SVD-LLM fine-tuned on WikiText-2 directly.

Compression Ratio Method	40%	50%	60%	70%	80%
ASVD	57057.00	15358.00	1407.00	51.00	11.14
ASVD + LoRA FT	44.81	21.83	14.86	10.16	8.37
SVD-LLM	42.30	23.97	13.11	9.56	7.94
SVD-LLM + LoRA FT	17.93	13.26	10.65	9.14	7.78
ACIP	24.00	16.17	12.46	10.34	8.83
ACIP + FT	17.33	11.88	9.63	8.27	7.19

Table A4: More detailed evaluation results for our quantization experiments in Appendix E.3, reported in terms of numbers.

Size	Ablation	Eff. model size [GB]	C4 ↓	WikiText-2 ↓	ARC_c ↑	ARC_e ↑	HellaS. ↑	MathQA ↑	Openb. ↑	PIQA ↑	WinoG. ↑	LM Eval Avg. ↑
40%	ACIP	5.47	21.05	24.00	0.24	0.49	0.35	0.21	0.19	0.65	0.55	0.38
	ACIP → FT	5.47	15.66	17.33	0.24	0.53	0.38	0.21	0.20	0.67	0.57	0.40
	ACIP → W4A16	1.89	27.12	35.40	0.22	0.46	0.33	0.20	0.18	0.62	0.53	0.36
	ACIP → W4A16 → FT	1.89	16.67	18.90	0.23	0.52	0.37	0.22	0.20	0.67	0.57	0.40
50%	ACIP	6.70	16.47	16.17	0.28	0.58	0.40	0.22	0.21	0.68	0.57	0.42
	ACIP → FT	6.70	12.89	11.88	0.27	0.59	0.43	0.23	0.23	0.71	0.60	0.44
	ACIP → W4A16	2.21	19.28	19.96	0.26	0.54	0.37	0.21	0.20	0.67	0.55	0.40
	ACIP → W4A16 → FT	2.21	13.85	13.33	0.25	0.58	0.41	0.23	0.23	0.70	0.59	0.43
60%	ACIP	7.88	13.91	12.46	0.30	0.61	0.43	0.23	0.25	0.71	0.60	0.45
	ACIP → FT	7.88	11.14	9.63	0.32	0.64	0.47	0.23	0.26	0.73	0.63	0.47
	ACIP → W4A16	2.51	15.84	14.64	0.29	0.58	0.42	0.22	0.22	0.69	0.57	0.43
	ACIP → W4A16 → FT	2.51	11.77	10.31	0.29	0.64	0.45	0.22	0.26	0.72	0.61	0.46
70%	ACIP	9.10	12.22	10.34	0.31	0.64	0.47	0.23	0.27	0.73	0.62	0.47
	ACIP → FT	9.10	9.84	8.27	0.34	0.69	0.50	0.24	0.28	0.75	0.66	0.49
	ACIP → W4A16	2.83	13.45	11.80	0.29	0.63	0.45	0.23	0.24	0.72	0.60	0.45
	ACIP → W4A16 → FT	2.83	10.38	8.74	0.32	0.67	0.48	0.23	0.28	0.75	0.64	0.48
80%	ACIP	10.30	10.91	8.83	0.33	0.67	0.49	0.23	0.28	0.74	0.63	0.48
	ACIP → FT	10.30	8.81	7.19	0.38	0.71	0.53	0.24	0.31	0.77	0.68	0.52
	ACIP → W4A16	3.13	12.61	9.87	0.32	0.65	0.47	0.23	0.28	0.74	0.61	0.47
	ACIP → W4A16 → FT	3.13	9.26	7.60	0.36	0.69	0.52	0.24	0.30	0.76	0.66	0.50
90%	ACIP	11.50	9.75	7.56	0.34	0.68	0.50	0.25	0.27	0.75	0.63	0.49
	ACIP → FT	11.50	7.69	6.12	0.39	0.73	0.56	0.25	0.33	0.78	0.69	0.53
	ACIP → W4A16	3.44	10.25	7.90	0.32	0.66	0.50	0.24	0.27	0.75	0.63	0.48
	ACIP → W4A16 → FT	3.44	7.97	6.39	0.38	0.73	0.56	0.25	0.35	0.78	0.69	0.53
100%	ACIP	12.70	9.52	7.20	0.33	0.69	0.51	0.25	0.26	0.76	0.63	0.49
	ACIP → FT	12.70	7.32	5.75	0.40	0.75	0.57	0.26	0.34	0.79	0.70	0.54
	ACIP → W4A16	3.75	9.75	7.37	0.34	0.69	0.51	0.25	0.27	0.76	0.63	0.49
	ACIP → W4A16 → FT	3.75	7.52	5.94	0.40	0.75	0.56	0.27	0.34	0.78	0.69	0.54

Table A5: Efficiency analysis of `ACIP` for **LLaMA-7B**. The first three rows report the runtime and memory statistics of `ACIP`'s key steps (see Section 2 and Figure A3) both in terms of numbers and their qualitative asymptotics. Here, the model sizes are measured as (uncompressed) checkpoint sizes. "Runtime pruning" refers to the process of pruning the mask parameters to a desired compression ratio (revertible), whereas "Runtime compress" refers to the process of discarding pruned singular vectors and possibly unparametrizing linear layers, so that the model gets actually compressed. The statistics of inference speed were obtained by generating new text of sequence length 64 and batch size 64. To measure FLOPs, we used the `fvcore` package and an input sequence of length 512.

Stage	Metric	LLaMA-7B
ACIP Step 1 (Model Reparametrization) $\mathcal{O}$(#Layers $\times$ SVD of Layer)	Runtime [min]	4.95
	Size parametrized model [GB]	19.71
	Size base model [GB]	12.70
ACIP Step 2 (Scoring by Iterative Pruning) $\mathcal{O}$(#Steps of Masks & LoRA Updates)	Runtime [min]	23.12
	Reserved GPU memory peak [GB]	62.45
	Steps / s	1.68
ACIP Step 3 (Any Compression) $\mathcal{O}$(#Layers $\times$ Layer Input Dimension)	Runtime pruning [s]	0.49
	Runtime compress [s]	0.18
Inference at 40% Size	Size model [GB]	5.47
	Reserved GPU memory peak [GB]	25.68
	Latency [s]	2.57
	Tokens / s	1594.99
	GigaFLOPs	1335.85
Inference at 70% Size	Size model [GB]	9.10
	Reserved GPU memory peak [GB]	29.43
	Latency [s]	2.47
	Tokens / s	1658.63
	GigaFLOPs	2265.03
Inference at 100% Size (Original)	Size model [GB]	12.70
	Reserved GPU memory peak [GB]	32.79
	Latency [s]	1.67
	Tokens / s	2447.75
	GigaFLOPs	3188.63

E.4. Efficiency Analysis

Table A5 reports several statistics on the efficiency of the `ACIP` algorithm and inference speed of compressed models. While these preliminary results do not immediately indicate gains in inference speed, we expect that further optimization like merging the low-rank adapters can compensate for the matrix-factorization overhead and outperform the base model. Moreover, we note that compared to performance-size trade-offs, which are our main concern, analyzing inference speed-ups requires a very careful consideration about the hardware in use (accelerator model, parallel processing units, etc.) and measurement setup (sequence length, batch size, etc.).

E.5. Impact of Low-Rank Adapters

The primary purpose of the low-rank adapters used in `ACIP` is to correct compression errors on-the-fly during the optimization. A surprising finding of our work is that the final adapters are "universal" in the sense that they can be used across all seen compression levels. While we expect that other PEFT-style approaches would lead to similar findings, it is natural to ask how `ACIP` would perform without any correction, i.e., just the mask parameters are tuned according to (4). This ablation study is shown in Figure A7. While performing significantly worse than with LoRA, we observe that the perplexity does not blow up and the results are even slightly better than SVD-LLM (see Table 1). This stable behavior of `ACIP` is closely related to our parameterization of the mask in (3) which ensures that the forward pass corresponds to the actual outputs of the pruned model with binary masks. On the other hand, the straight-through estimator still enables backpropagation.

E.6. Impact of the Stopping Criterion

In most experiments, we have used a compression ratio r_{stop} of 0.4 as stopping criterion, i.e., the pruning of masks is stopped if the size of the model is only 40% of the original one (measured in number of parameters of all target weight matrices). We have observed that at this point, the model performance has typically dropped so much that even a fine-tuned model would be of limited practical use.

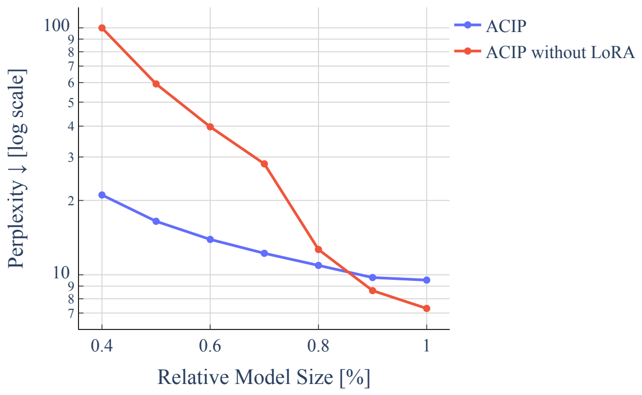


Figure A7: Compression-performance trade-off curves for **LLaMA-7B** on **C4** with and without using a LoRA-adapter for correction in `ACIP`.

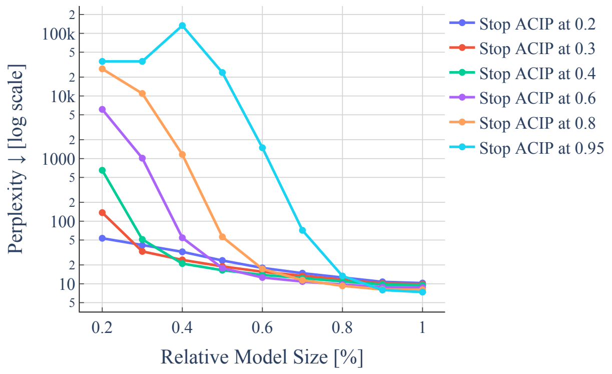


Figure A8: Compression-performance trade-off curves for **LLaMA-7B** on **C4**, using different stopping compression ratios r_{stop} for `ACIP`.

Nevertheless, it is interesting to explore the sensitivity of compression-performance curves against different stopping ratios. The comparison shown in Figure A8 provides several insights in this respect: (i) "Forecasting" compressed models beyond the stopping ratio does not work very well, especially when stopping very early (> 0.8). (ii) The predictive capacity of `ACIP` remains valid for even stronger stopping compression ratios than 0.4. However, finding the largest reasonable stopping ratio is highly model-dependent. For less compressible models like LLaMA-3.1-8B, it could make sense to stop even earlier than 0.4 (cf. Figure 1). In general, we hypothesize that older models are more compressible than new ones, as the latter "carry" more information per weight due to significantly more training data (Allen-Zhu & Li, 2024).

E.7. Impact of the Score Map – Forecasting Pruning Patterns

Here, we pick up the observation from Appendix E.6 that forecasting the performance of compressed models beyond the stop ratio leads to inaccurate predictions, i.e., the model is compressed more strongly than it has been done by `ACIP` itself. However, it turns out that the score map itself exhibits a certain forecasting capability. To this end, we run `ACIP` as usual until a stop ratio is reached, say $r_{\text{stop}} = 0.4$, but we stop updating the score map earlier in the optimization process. A few compression-performance curves with this modification are reported in Figure A9. We observe very similar curve shapes even if the score map is frozen after only a tiny fraction of mask parameters was pruned. This underpins our intuition that the pruning path of each parameter is fully determined at very early stage of `ACIP`.

E.8. Impact of the Score Map – A Trivial One Does Not Work

There are certainly alternative ways to design useful score maps. For example, simply accumulating the gradients of all mask parameters entrywise over an `ACIP`-run works equally well as the strategy proposed in Section 2. It is therefore valid to ask whether one could even design score maps without any optimization. We demonstrate that perhaps the most

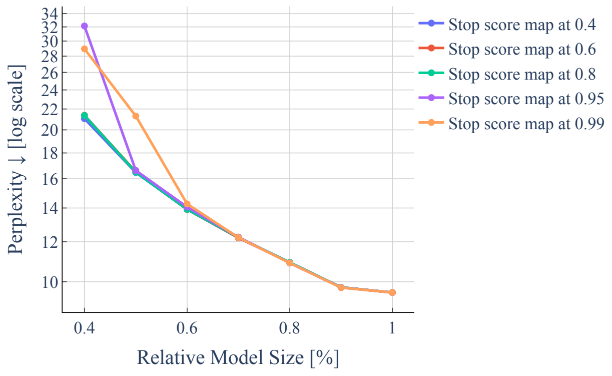


Figure A9: Compression-performance trade-off curves for **LLaMA-7B** on **C4**, stopping updates of the score map before the actual stopping criterion of `ACIP`.

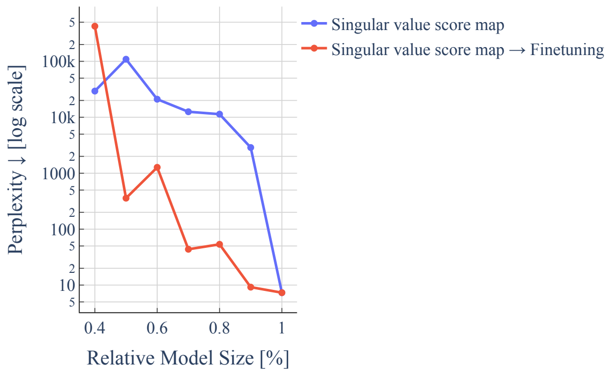


Figure A10: Compression-performance trade-off curves for **LLaMA-7B** on **C4**, using a trivial score map based on the initial singular values of the base model.

obvious approach, namely setting the score map equal to the singular values of the weight matrices, does not work very well. Figure A10 shows that this training-free approach does not produce any reasonable compressed models and decent performance cannot be easily recovered with LoRA-finetuning. This simple experiment confirms that designing useful score maps is not a trivial endeavour and requires a carefully crafted algorithmic approach.

E.9. Impact of Post-Tuning

Our main experiments are performed with 1k post-tuning steps in `ACIP` (see the description in Section 2.2). Figure A11 shows analogous compression-performance trade-off curves for fewer or no post-tuning steps. We observe that post-tuning can indeed notably increase performance for higher compression ratios.

E.10. Impact of Individual Layers – Example of LLaMA2-13B

As pointed out in the caption of Table A2, the linear layers targeted by `ACIP` were slightly modified for LLaMA2-13, namely all up projection layers were ignored. Figure A12 shows what would happen if they are compressed as well. While the performance predictions for ≥ 0.6 look decent, the perplexity explodes for stronger compression; note that even additional fine-tuning does not recover a reasonable performance in this situation. We hypothesize that `ACIP` has pruned one or more singular values of the up projection layers that are crucial for model's integrity. This finding might be related to the recent work by Yu et al. (2024) on pruning so-called super weights. In any case, `ACIP` is capable of revealing this undesirable behavior as demonstrated in Figure A12.

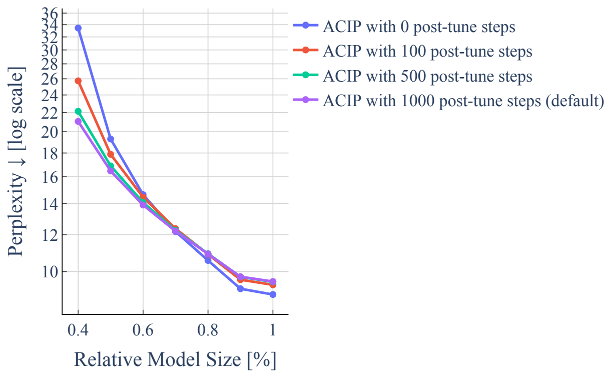


Figure A11: Compression-performance trade-off curves for **LLaMA-7B** on **C4** with different numbers of post-tuning steps.

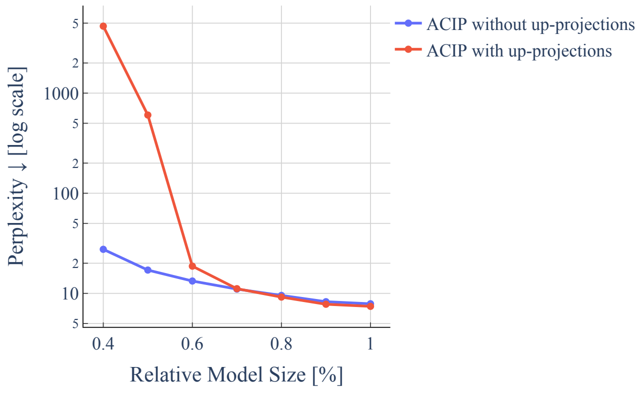


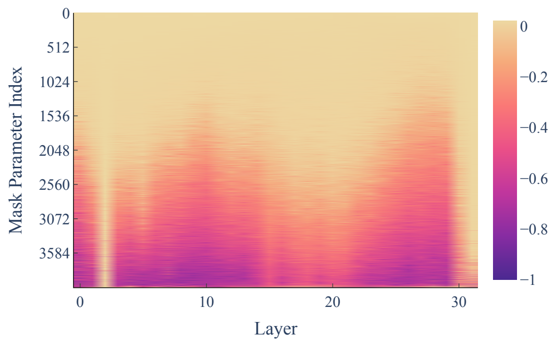
Figure A12: Compression-performance trade-off curves for **LLaMA2-13B** on **C4**, (not) ignoring the up projection layers in `ACIP`.

E.11. Examples of Score Maps Generated by `ACIP`

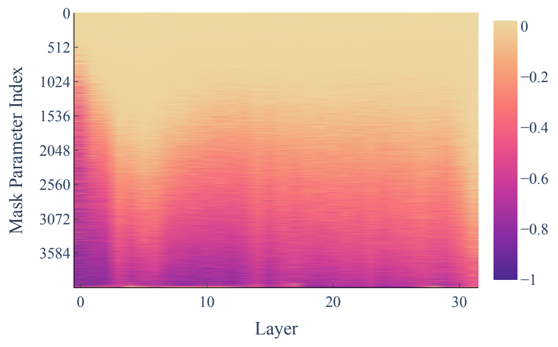
Figure A13 and Figure A14 show two typical score maps generated by `ACIP` for LLaMA-7B and Qwen2.5-7B, respectively. A characteristic feature is that attention layers can be pruned more aggressively than the MLP layers. Similarly, we observe non-uniform pruning patterns for layers of the same type across all transformer layers. This confirms the findings of (Yuan et al., 2024; Jaiswal et al., 2024) and demonstrates that non-uniform structured compression can be achieved without any feature engineering.

E.12. Examples of Generated Text by Compressed Models

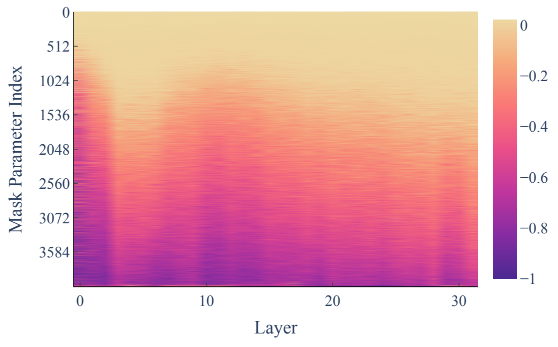
Table A6 shows examples of generated text by compressed versions of LLaMA-7B.



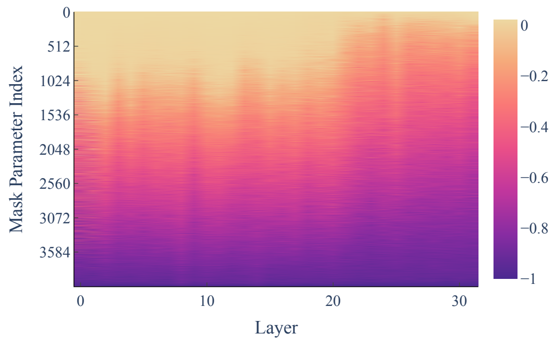
(a) Down Projections



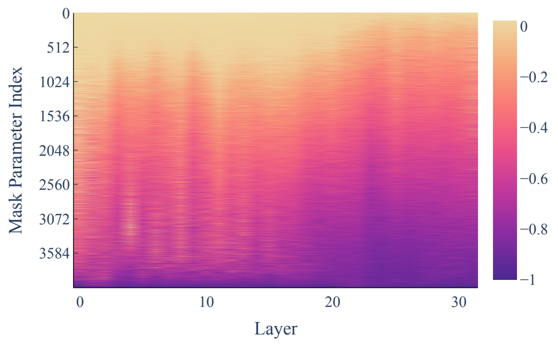
(b) Up Projections



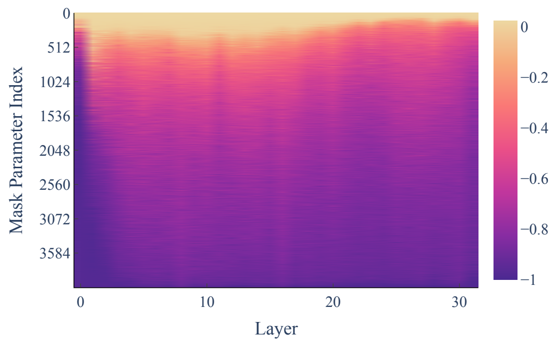
(c) Gate Projections



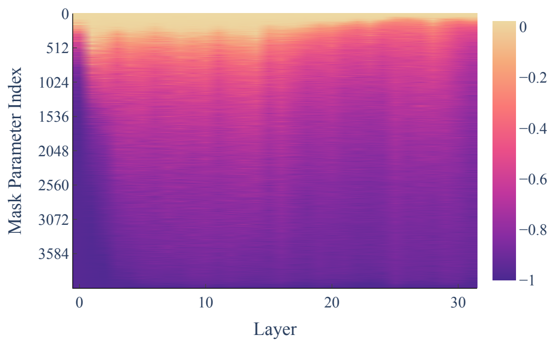
(d) Attention-O



(e) Attention-V

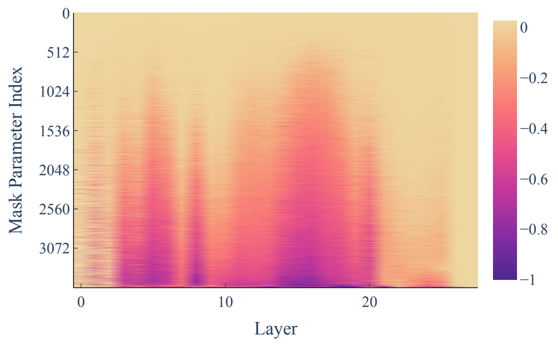


(f) Attention-Q

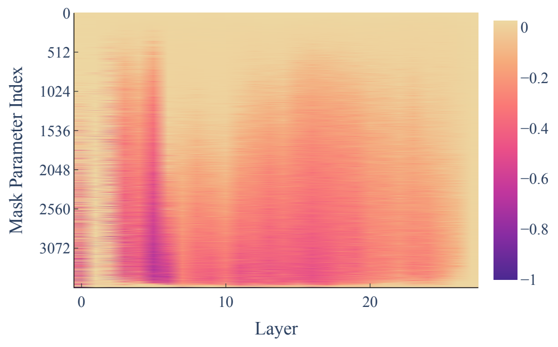


(g) Attention-K

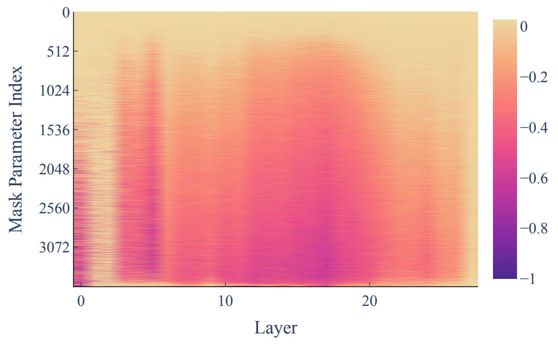
Figure A13: Example score maps generated by `ACIP` for **LLaMA-7B**. The negative values (cf. Step 2 of `ACIP` in Section 2.2) are normalized to -1 for the purpose of visualization.



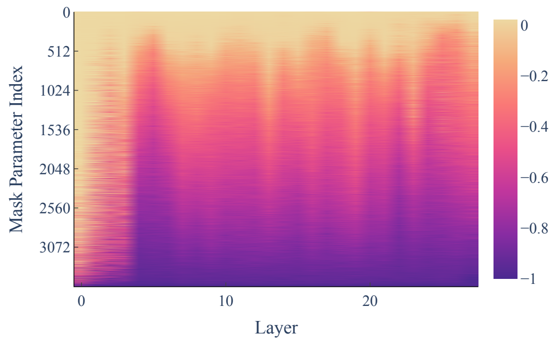
(a) Down Projections



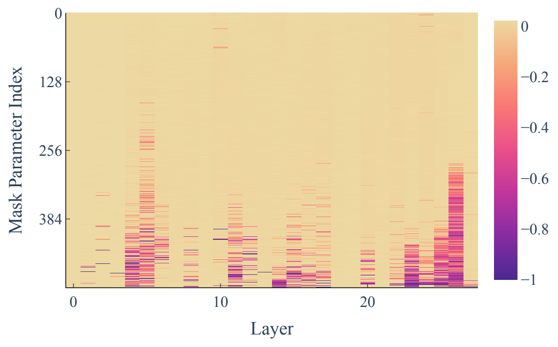
(b) Up Projections



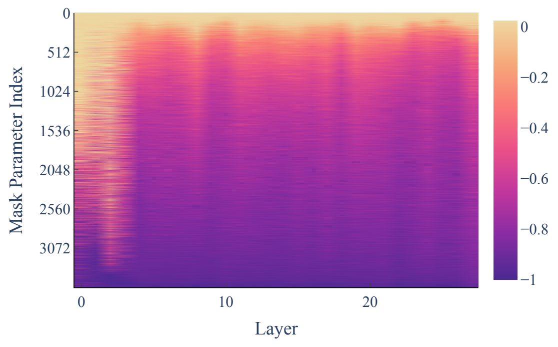
(c) Gate Projections



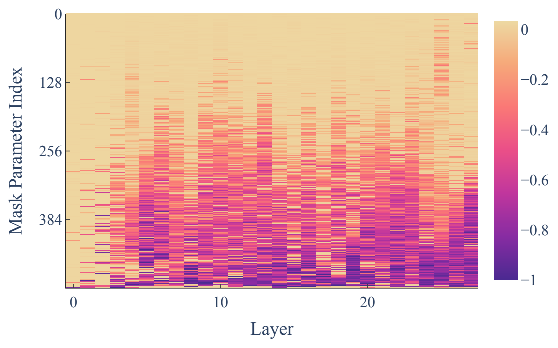
(d) Attention-O



(e) Attention-V



(f) Attention-Q



(g) Attention-K

Figure A14: Example score maps generated by `ACIP` for **Qwen2.5-7B**. The negative values (cf. Step 2 of `ACIP` in Section 2.2) are normalized to -1 for the purpose of visualization.

Table A6: Example texts for two prompts generated for **LLaMA-7B** under different compressions produced by `ACIP`.

Size	**"How to compress Large Language Models?"**	**"What is the concept of time?"**
Original	I am trying to run a large Language model generation, but I have a few files on disk of around 3gb each. I was wondering if there is a way to compress these files? I have searched the internet but have found nothing. Re: How to compress Large Language Models? I suggest compressing it to HDF5, you can find instructions on how to do this using TensorPacker and PNet on the TensorFlow Github repository …	Have you ever tried to ask a baby what it sees? Time, is it real? What is the concept of time? Have you ever tried to ask a baby what it sees? Let's see what they can understand. The concept of time. What it is about, who discovered the idea of time, and how was it discovered? How is time measured? We can't define it. We can't point at what we are defining, we can't even show a picture of it …
90%	Jeffrey Hunt, a linguist and applied deep learning researcher at the University of Technology, Sydney, talks with Caleigh Blackstock"s to discuss how to compress a large language model to speed up training and model execution. What does the language model compression actually do to the overall model? …	Please let me know what is the concept of time. The concept of time is all about the measurement of time. This measurement was first done using the hourglass. When this hour Glass was invented, it was the only thing that could represent the passage of time. …
80%	Jeffrey Shafman, Zsolt Nemeth, Jeffrey Shafman, Zsolt Nemeth, Jeffrey Shafman, Zsolt Nemeth. (2013). How to compress Large Language Models? In: ICML proceedings, Vol. 22, pp. 462-463. Magnus Skallman. (2013). How to compress Large Language Models? In: ICML13 Proceedings, pp. 462-463. …	You can't even imagine that in that time, The life of a tree in nature is as brief As a flash of a lightning bolt, But in this human world, it may endure Through many millennia, if the will of man. So tell me, what is the concept of time? Hey, I've got one, and it's this: …
70%	Apr 04, 2018 at 12:05 pm In the last 5-6 years, Natural Language Systems have been an important direction for my work and the research I do in the course of my PhD. I have been using them to teach myself Swedish and improve my Swedish by machine time in this post. Let's see if I've got it! …	Please tell me where is the concept of time in this poem. What is the significance of the poem "I Am Wine" by H.W.C. Carson"? What does he mean by the poem title? What does the stairs, the wine, & the man have in common? What does he mean by "I am wine"? What do you mean by "I am wine"? …
60%	This is a 14 day free trial. Please use the following link to sign in. Select the country you need here to view pricing and purchase. Whenever you use our web site, we will track your website traffic, the page where you first connected to our site, and the search keywords you use to find us. We'll use this information to help us continually improve our website. …	You may be wondering what is the concept of time. The concept of time is all about the phenomenon of time. It says that there is never no time. It states that there is always a moment in time. At the moments of time is when we refer the time of life. So, as a matter of the moment, time is just a point in space. …
50%	Jeffrey Huan, Shu-Hua Ji, Xiao-Hong Li, Xiao-Xiao Li, Layin Ma"s. 2019:5, Springer.0000.0001/13/8/19/12/720-22-F43. Miller A. "The Structure of the Language in Unsupervised Deep-Processing" in Journal of the Acoustical Actors (v. 11(6). 2006:, pp. 278-295). Dong Ming, Sun Jin, He Mei-Chen,' …	Getting to this answer isn't that hard. The concept of time is all about the study of time. The study of time isn't so hard to get a hold on. When you're younger, you can begin thinking of how much you could have had if you had known how much time you had...if only we had it back then …
40%	Apr 14, 2018 Resumes at The University of Florida will be released on Friday, May 29, 2018 2.05:53pm. Cover your next job announcement with the help of our new Resume Template Builder. To create your own Resume Template in minutes. Improving the quality of your Resume. To improve your Resume …	You may think that it is just a fancy word, or just the idea it had in the earlier world. But there exists a way to understand it. To understand the idea of time by using the example of a clock, you can learn the very importance of time with a simple strategy. The clock ticks with a watch. The clock has it time to operate. …