TASKD-LLM: TASK-AWARE SELECTIVE KNOWLEDGE DISTILLATION FOR LLMS

Khouloud Saadi & Di Wang
Provable Responsible AI and Data Analytics (PRADA) Lab
King Abdullah University of Science and Technology, Saudi Arabia
{khouloud.saadi,di.wang}@kaust.edu.sa

ABSTRACT

Large language models achieved state-of-the-art performance in generative tasks but are computationally expensive, making them impractical for deployment in resource-constrained environments. Knowledge distillation (KD) is a promising technique for compressing LLMs by transferring knowledge from a large teacher to a more efficient student model. However, existing task-based KD methods distill all teacher model components indiscriminately. Since teacher models are typically pre-trained for versatility across a broad range of tasks, this approach can introduce unnecessary complexity when distilling for a specific downstream task, potentially limiting the student's ability to specialize. Furthermore, previous work showed that only a subset of the LLM components significantly contribute to a given task, making indiscriminate distillation inefficient. Motivated by these insights, we propose task-aware selective KD (TASKD-LLM), a novel approach that transfers only task-relevant knowledge from the teacher to the student, simplifying the distillation process and maintaining the student's focus. Our method is flexible and can be combined with other distillation techniques in a plug-and-play manner. Empirical results demonstrate that TASKD-LLM outperforms existing methods, achieving higher performance on several benchmark datasets.

1 INTRODUCTION

Recently, there has been a surge in using large language models (LLMs) for generative tasks (OpenAI, 2023), where they achieved good performance across diverse applications (Zhuge et al., 2024; OpenAI, 2023; Touvron et al., 2023; Wang et al., 2023). Despite their remarkable success, these models are computationally intensive and often impractical for deployment in resource-constrained environments. Hence, there has been an interest in making LLMs more efficient in terms of storage and computation through knowledge distillation (KD) (Zhu et al., 2024; Xu & McAuley, 2022).

In the standard LLMs KD framework, the teacher model is typically a large, versatile language model (Gu et al., 2024), pre-trained on a diverse dataset (OpenAI, 2023). During distillation, all components of the teacher are transferred uniformly to the student model (Gu et al., 2024; Peng et al., 2023; Kim & Rush, 2016; Sanh et al., 2019). While this approach is beneficial for training generally capable student models (Jiao et al., 2020; Sanh et al., 2019), many real-world applications prioritize performance on a specific downstream task (Ge et al., 2023). In such cases, the broad versatility of the teacher model may introduce unnecessary complexity, potentially hindering the student model's ability to specialize effectively (Ojha et al., 2023).

Furthermore, in the context of LLMs, it was shown in Hase et al. (2024); Gromov et al. (2024a); Luo et al. (2024); Dai et al. (2021) that only a part of the LLM components significantly contribute to a given task. To further confirm this, Figure 1 illustrates the sparsity of the activation map for the last hidden state on a downstream example, along with the percentage of small activation magnitudes (< 2), averaged over all fine-tuning data, across all hidden layers of the fine-tuned gpt2-xlarge. As shown, many neuron activations are either zero or have low magnitudes, and thus contribute weakly to the final model output. Additional illustrations with different thresholds are provided in Appendix A. This highlights the inefficiency of conventional LLMs KD, which transfers all teacher components indiscriminately, even though many do not contribute meaningfully to a specific task.

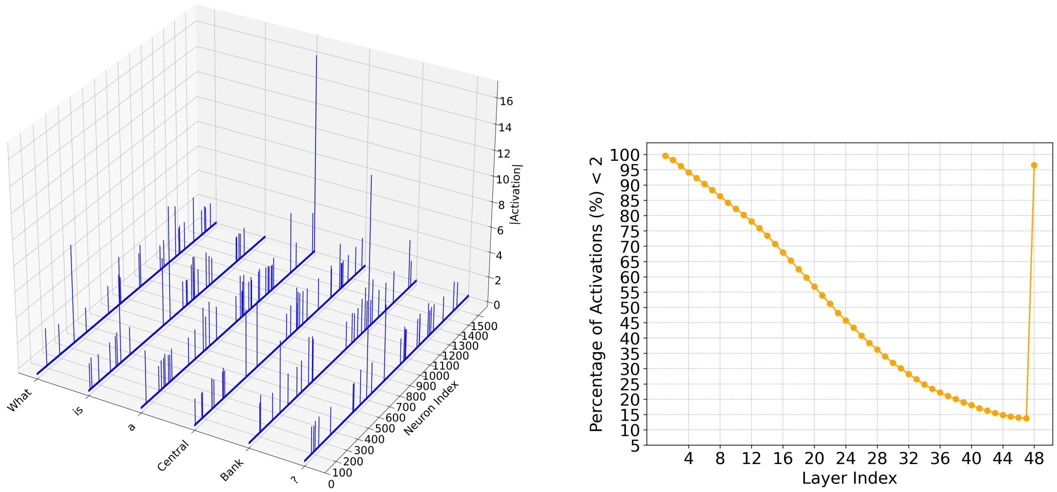


Figure 1: **(Left)**: Last hidden state activation magnitudes (z-axis) for a fine-tuned gpt2-xlarge on a downstream example, with magnitude values < 2 set to zero for visualization. The x/y axes represent sequence/features. **(Right)**: Percentage of small activation magnitudes (< 2) across all hidden layers. See Appendix A for more illustrations with different thresholds.

Motivated by these findings, we propose TASKD-LLM, a task-aware knowledge distillation approach for LLMs. Specifically, we locate and target only the task-relevant knowledge in the teacher's hidden layers during the distillation process. Unlike existing KD methods that transfer knowledge indiscriminately from all units, we focus on distilling the information from the components of the teacher model most responsible for the task at hand. In summary, our contributions are as follows:

- We leverage the inherent versatility of teacher models, recognizing that only a fraction of their nodes are relevant to a given task in the context of knowledge distillation.
- We propose a novel knowledge distillation approach namely Task-Aware Selective Knowledge Distillation for LLMs (TASKD-LLM) that locates and transfers task-relevant knowledge from the teacher model to the student model.
- Our proposed approach can be combined in a plug and play manner with other KD methods. Experimental results on various language generation benchmarks show that our approach outperforms existing methods on several datasets, achieving performance improvements of up to 0.96% in Rouge-L score on S-NI dataset.

For a detailed related work, see Appendix B.

2 METHOD

In this work, we propose TASKD-LLM, where we identify and distill only the most task-relevant teacher components to the student. Our method addresses the limitations in the existing task-based KD approaches (Gu et al., 2024; Sun et al., 2019a), which compress a versatile teacher model indiscriminately, transferring all its knowledge to the student without distinguishing between task-relevant and irrelevant information. These traditional methods often lead to inefficiencies, as not all teacher components contribute meaningfully to every task (Dai et al., 2021).

Given a teacher hidden layer, we first locate the task-relevant neurons. To achieve that, we employ the gradient attribution method (Krishna et al., 2024; Simonyan et al., 2014; Baehrens et al., 2010) to calculate the gradient of the output with respect to each neuron's activation. The magnitude of the gradient reflects how much each node affects the model's output, providing a measure of task relevance score for each unit. Next, neurons with higher gradient magnitudes are deemed more influential and are thus prioritized for distillation. Formally, given a teacher model T modeled by a function F and a student model S, the importance score on the input x of the i-th neuron in the hidden layer l of the teacher model T is computed as $\text{score}_i(x) = \frac{\partial F(x)}{\partial a_i^l}$, where $F(x)$ is the output logit of T on the input x and a_i^l is the activation of the corresponding neuron in the layer l. The scores of all the units in the layer l of T are computed as follows:

$$\text{scores}^l(x) = \{\frac{\partial F(x)}{\partial a_1^l}, \frac{\partial F(x)}{\partial a_2^l}, ..., \frac{\partial F(x)}{\partial a_N^l}\}, \tag{1}$$

where N is the size of the hidden state l of T. Lastly, we rank the scores of all the units in layer l and select the n highest scores to target their corresponding nodes for distillation, where n is the corresponding hidden size of the student model S. Unlike previous methods that require identical hidden layer sizes (Sun et al., 2019b) or an additional projector training (Jiao et al., 2020) to align the teacher and the student hidden representations, TASKD-LLM allows the teacher and student models to have different hidden layers dimensions without requiring extra computations. The selected task-relevant neurons, from T are obtained as follows:

$$R^l(B) = \left\{ \sum_{b=1}^{B} \frac{\partial F(x)}{\partial a_{i_1}^l}, \ldots, \sum_{b=1}^{B} \frac{\partial F(x)}{\partial a_{i_N}^l} \;\middle|\; \sum_{b=1}^{B} \frac{\partial F(x)}{\partial a_{i_1}^l} > \cdots > \sum_{b=1}^{B} \frac{\partial F(x)}{\partial a_{i_N}^l} \right\}, \tag{2}$$

where B is the batch size and b is the sample index in the batch B. The set of selected neurons $E^l(B) = \{u_{i_1}^l, u_{i_2}^l, ..., u_{i_n}^l\}$ is the set of units that correspond to the first n scores in the set R(B), where u is namely for unit.

Distillation Loss Function: To transfer the knowledge of these carefully selected neurons $E^l(B)$ to the student model S, we employ a correlation loss function, which was shown to be more effective than traditional mean squared error (MSE) and cosine distance in capturing meaningful relationships in the feature space (Saadi et al., 2023; Fard & Mahoor, 2022). In the context of LLMs, Dai et al. (2021) showed that factual knowledge in transformer models is primarily stored in MLP layers rather than in attention. Moreover, Gromov et al. (2024a); Men et al. (2024) emphasized the significance of the last hidden state over other layers in generative tasks for LLMs. Thus, in TASKD-LLM, we focus on distilling the teacher's last hidden state L to the student's last hidden state. Formally, we maximize the cross-correlation between the set of units $E^L(B)$ from T and their corresponding units in S as follows:

$$L_{TASKD-LLM} = \sum_{j}^{n} (1 - C_{jj})^2, \tag{3}$$

where $C_{jj} = \frac{\sum_{b=1}^{B} a_{b,j}^T a_{b,j}^S}{\sqrt{\sum_{b=1}^{B} (a_{b,j}^T)^2} \sqrt{\sum_{b=1}^{B} (a_{b,j}^S)^2}}$ which represents the cross-correlation value between the feature representation a_j^T, which corresponds to the j-th neuron activation from $E^L(B)$ of T, and a_j^S, which is the j-th neuron activation in the student's last hidden state. The final training loss of the student model is: $Loss = \alpha L_{TASKD-LLM} + \beta L_2 + L_{LM}$, where L_2 is the logit distillation loss, e.g., Gu et al. (2024), and L_{LM} is the supervised language modeling loss (Radford et al., 2019).

3 Experimental results and Discussion

In this section, we present our experimental setup and discuss our results. The conducted experiments follow a similar setup to the one outlined in Gu et al. (2024). In all experiments, the teacher is the gpt2-xlarge model, with 1.5 billion parameters, after being fine-tuned on the instruction-following Dolly[1] dataset. In this work, our target task is instruction-following. We experiment with two student models, namely gpt2-base (120 million) and gpt2-medium (340 million). For evaluation metrics, similar to Gu et al. (2024), we report the Rouge-L (Lin, 2004) score on the following benchmark datasets: Dolly test dataset, SelfInst (Wang et al., 2022a), Vicuna (Chiang et al., 2023), S-NI (Wang et al., 2022b), and UnNI (Honovich et al., 2023) datasets. The Rouge-L score measures the precision of the model generation and it was shown by Wang et al. (2022b) that it is suitable for large-scale instruction-following evaluation. For L_2, we use the proposed logit loss in Gu et al. (2024), which applies a reverse Kullback-Leibler divergence (Kullback, 1951) between the teacher and the student logits. Full experimental details are available in Appendix C.

As shown in Table 1, our proposed distillation approach outperforms state-of-the-art methods, including FT (fine-tuning), KD (word-level KD), SeqKD (sequence-level KD), and MiniLLM, on

[1]https://github.com/databrickslabs/dolly/tree/master

most datasets. For gpt2-base (120 M), TASKD-LLM achieves the highest performance on S-NI (23.07%), UnNI (25.32%), and Vicuna (17.91%) and ranks second on Dolly and SelfInst. For gpt2-medium (340 M), TASKD-LLM outperforms all methods on four out of five datasets, often by a significant margin. For instance, on UnNI, TASKD-LLM surpasses MiniLLM by 0.79% and SeqKD by 6.98%. These results demonstrate that focusing on task-relevant components is a promising direction for task-based KD, leading to improved student model performance.

Notably, as shown in Table 1, the distilled gpt2-base medium (340 M) model with our TASKD-LLM outperforms our trained teacher model on SelfInst, Vicuna, S-NI, and UnNI. This phenomenon, where a distilled model surpasses its teacher, has been observed in prior work (Gu et al., 2024; Stanton et al., 2021; Furlanello et al., 2018), highlighting the student model's strong generalization capabilities.

Table 1: The Rouge-L score (%) of the different approaches. * Results reported from Gu et al. (2024). Results are averaged over 3 random seeds. M for million and B for billion.

#params	Method	Dolly	SelfInst	Vicuna	S-NI	UnNI
1.5 B	Teacher*	27.60	14.03	16.30	27.60	31.80
	Teacher	26.67	13.72	16.18	25.14	28.56
120 M	FT*	23.30	10.00	14.70	16.30	18.50
	KD* (Sanh et al., 2019)	22.80	10.08	13.40	19.70	22.00
	SeqKD* (Taori et al., 2023)	22.70	10.10	14.30	16.40	18.80
	MiniLLM Gu et al. (2024)	**24.43**	**12.47**	17.89	22.11	24.48
	TASKD-LLM (ours)	24.31	12.29	**17.91**	**23.07**	**25.32**
340 M	FT*	25.50	13.00	16.00	25.10	28.10
	KD* (Sanh et al., 2019)	25.00	12.00	15.40	23.70	24.60
	SeqKD* (Taori et al., 2023)	25.30	12.60	16.90	22.90	23.30
	MiniLLM (Gu et al., 2024)	25.62	14.19	**18.03**	24.93	29.49
	TASKD-LLM (ours)	**26.02**	**14.57**	17.68	**25.61**	**30.28**

Ablation Study: As explained in Section 2, TASKD-LLM focuses exclusively on distilling the last hidden state of the teacher to the last hidden state of the student. In Table 2 of Appendix D, we experiment with different layer combinations. As shown in Table 2, the performance across all layer combinations is generally comparable. However, distilling the last two layers from the teacher model to the last two layers of gpt2-base yields superior performance on four datasets, i.e., SelfInst, Vicuna, S-NI, and UnNI, compared to other configurations. This suggests that incorporating layer selection techniques, such as those in Gromov et al. (2024b); Belrose et al. (2023), could further enhance our approach. To further validate the effectiveness of our approach, which targets task-relevant neurons, in Table 3 of Appendix D, we compare TASKD-LLM with random neuron distillation. Indeed, targeting the relevant knowledge through the gradient attribution method (Grad) in the distillation process outperforms random distillation in most cases. For instance, in the case of gpt2-medium (340 M), TASKD-LLM has a rouge-L score of 26.02% on Dolly dataset while the random approach (Rand) has 25.55%. This perfectly aligns with the findings of Hase et al. (2024); Dai et al. (2021), which showed that only certain components of an LLM significantly contribute to a specific task.

4 Conclusion and Future Work

In this paper, we introduced TASKD-LLM, a novel task-based knowledge distillation approach that reduces LLMs size while preserving performance. By identifying and distilling only task-relevant components from the teacher model, our method produces more efficient student models. Notably, TASKD-LLM is the first to explore feature distillation in LLMs for generative tasks. We showed through preliminary experiments that our approach is a promising direction for task-based KD. Future work will include incorporating different layer selection strategies, exploring alternative methods for identifying task-relevant components (e.g., integrated gradients (Sundararajan et al., 2017), LogitLens (Belrose et al., 2023)), and extending our approach to different model architectures.

References

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831, 2010.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, march 2023. *URL https://lmsys. org/blog/2023-03-30-vicuna*, 3(5), 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

Ali Pourramezan Fard and Mohammad H Mahoor. Ad-corre: Adaptive correlation-based loss for facial expression recognition in the wild. *IEEE Access*, 10:26756–26768, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

Yingqiang Ge, Wenyue Hua, Kai Mei, Juntao Tan, Shuyuan Xu, Zelong Li, Yongfeng Zhang, et al. Openagi: When llm meets domain experts. *Advances in Neural Information Processing Systems*, 36:5539–5568, 2023.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers. *arXiv preprint arXiv:2403.17887*, 2024a.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A Roberts. The unreasonable ineffectiveness of the deeper layers, 2024. *URL https://arxiv. org/abs/2403.17887*, 2024b.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. Unnatural instructions: Tuning language models with (almost) no human labor. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14409–14428, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.806. URL https://aclanthology.org/2023.acl-long.806/.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372/.

Minsoo Kim, Sihwa Lee, Janghwan Lee, Sukjin Hong, Du-Seong Chang, Wonyong Sung, and Jungwook Choi. Token-scaled logit distillation for ternary weight generative language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. I-bert: Integer-only bert quantization. In *International conference on machine learning*, pp. 5506–5518. PMLR, 2021.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1139. URL `https://aclanthology.org/D16-1139/`.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. Post hoc explanations of language models can improve language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Solomon Kullback. Kullback-leibler divergence, 1951.

François Lagunas, Ella Charlaix, Victor Sanh, and Alexander Rush. Block pruning for faster transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10619–10629. Association for Computational Linguistics, November 2021.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. Mixkd: Towards efficient distillation of large-scale language models. In *International Conference on Learning Representations*, 2021.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S Yu. Learning multiple tasks with multilinear relationship networks. *Advances in neural information processing systems*, 30, 2017.

Yuqi Luo, Chenyang Song, Xu Han, Yingfa Chen, Chaojun Xiao, Zhiyuan Liu, and Maosong Sun. Sparsing law: Towards large language models with greater activation sparsity. *arXiv preprint arXiv:2411.02335*, 2024.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024.

Utkarsh Ojha, Yuheng Li, Anirudh Sundara Rajan, Yingyu Liang, and Yong Jae Lee. What knowledge gets distilled in knowledge distillation? *Advances in Neural Information Processing Systems*, 36:11037–11048, 2023.

R OpenAI. Gpt-4 technical report. arxiv 2303.08774. *View in Article*, 2(5), 2023.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*, 2023.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3208–3229. Association for Computational Linguistics, November 2020.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. Fully quantized transformer for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1–14. Association for Computational Linguistics, November 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. Subformer: Exploring weight sharing for parameter efficiency in generative transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4081–4090. Association for Computational Linguistics, November 2021.

Khouloud Saadi, Jelena Mitrović, and Michael Granitzer. Learn from one specialized sub-teacher: One-to-one mapping for feature-based knowledge distillation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13235–13245, 2023.

Victor Sanh, L Debut, J Chaumond, and T Wolf. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. arxiv 2019. *arXiv preprint arXiv:1910.01108*, 2019.

K Simonyan, A Vedaldi, and A Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR, 2014.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919, 2021.

S. Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Conference on Empirical Methods in Natural Language Processing*, 2019a.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4323–4332, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1441. URL `https://aclanthology.org/D19-1441`.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2140–2151, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.188. URL `https://aclanthology.org/2021.findings-acl.188`.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022a.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340/.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Canwen Xu and Julian McAuley. A survey on model compression for natural language processing. *arXiv preprint arXiv:2202.07105*, 2022.

Rongzhi Zhang, Jiaming Shen, Tianqi Liu, Jialu Liu, Michael Bendersky, Marc Najork, and Chao Zhang. Do not blindly imitate the teacher: Using perturbed loss for knowledge distillation. *arXiv preprint arXiv:2305.05010*, 2023.

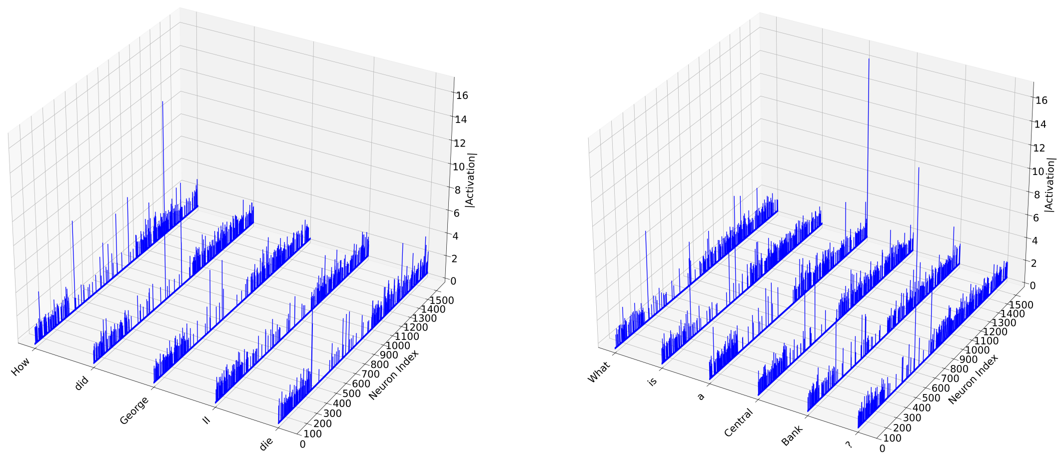
Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044*, 2017.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577, 2024.

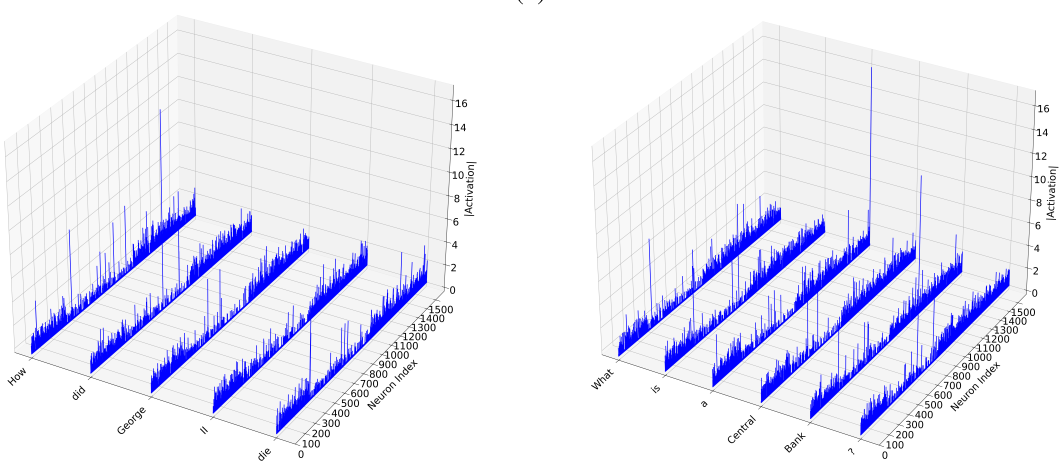
Mingchen Zhuge, Wenyi Wang, Louis Kirsch, Francesco Faccio, Dmitrii Khizbullin, and Jürgen Schmidhuber. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

A Additional Figures

(a)



(b)



(c)

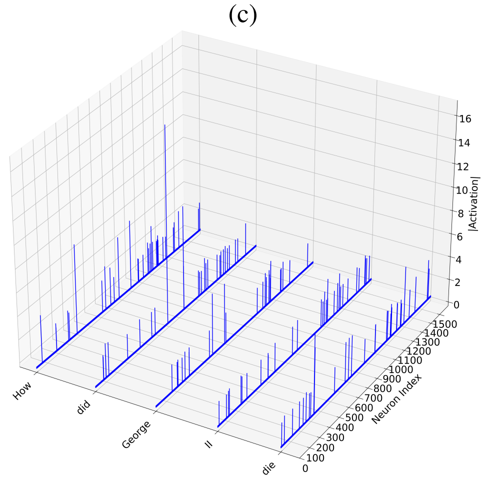


Figure 2: Activation magnitudes (z-axis) after feeding training samples from the downstream task to a fine-tuned gpt2-xlarge. x and y axes are sequence and feature dimensions: (a) We threshold values below 1 to zero. (b) We threshold values below 0.5 to zero. (c) We threshold values below 2 to zero.

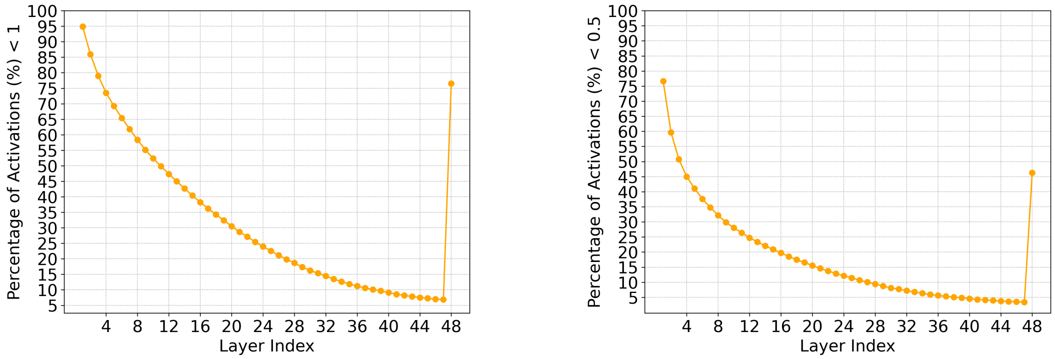


Figure 3: Percentage of small (< 1 and < 0.5, respectively) activation magnitudes (averaged across all fine-tuning data) across different layers of the fine-tuned gpt2-xlarge.

B Related Work

Model compression reduces neural network size while preserving performance. Common techniques include pruning, which removes unimportant weights, neurons, or layers (Frankle & Carbin, 2018; Lagunas et al., 2021; Prasanna et al., 2020); weight sharing, where different model parts reuse the same weights (Long et al., 2017; Lan et al., 2019; Reid et al., 2021); and quantization, which represents weights and activations with lower-bit integers instead of high-precision floats (Zhou et al., 2017; Kim et al., 2021; Prato et al., 2020).

Knowledge Distillation (KD) is also a widely used model compression technique that transfers knowledge from a large teacher model to a small, efficient student model (Sanh et al., 2019; Gou et al., 2021). In natural language processing (NLP), KD has been predominantly applied to text classification tasks by aligning the student model with the teacher's output distributions (Liang et al., 2021; Zhang et al., 2023), hidden representations (Sun et al., 2019b; Jiao et al., 2020), or attention matrices (Wang et al., 2020; 2021). These approaches effectively reduce model size while preserving performance, making them suitable for resource-constrained setups.

However, the application of KD in language generation tasks is more complex than in classification tasks (Gu et al., 2024). Unlike the fixed-label space of classification, open-ended text generation involves producing discrete token sequences of varying lengths, which adds inherent complexity. Existing KD techniques for generative models primarily minimize the forward Kullback-Leibler divergence (KLD) (Kullback, 1951) between the teacher and student model distributions (Sanh et al., 2019; Kim et al., 2024). This may involve supervision using the teacher's outputs at each generation step (Kim & Rush, 2016; Taori et al., 2023), training on teacher-generated text (Peng et al., 2023), or employing reverse KLD (Gu et al., 2024), which has shown promise in significantly improving student model performance.

C Experimental Details

The teacher model is GPT-2 xlarge, with 1.5 billion parameters, fine-tuned on the Dolly dataset for 10 epochs, using a learning rate of $1e - 5$. The Dolly dataset is constructed from `databricks-dolly-15K3`, which consists of 15K human-written instruction-response pairs (Gu et al., 2024). For all experiments, the batch size for both training and evaluation is set to 8. During training, all experiments are repeated for 3 random seeds. The models are trained for 7000 steps. The learning rate is set to $5e-6$, α and β are 0.5 and 1.0, respectively. For efficient computation of the selection process using the gradient attribution method, we focus on three tokens during fine-tuning: The first, the middle, and the last ones of the generated response text. Specifically, for each selected token, we compute the gradient of its probability distribution with respect to the last hidden state and average the attribution scores across the three token positions. For the L_{LM} loss, we use the OpenWebText (Gokaslan et al., 2019) dataset as in Gu et al. (2024). For all test sets, we sample the responses and report the average scores of 5 generations for each prompt with different random seeds as in Gu et al. (2024).

We compare our approach to the following competitive methods:

- **FT** refers to standard fine-tuning.
- **KD** (Sanh et al., 2019) namely, word-level KD, where the student model is trained on the teacher model's output at each token step.
- **SeqKD** (Taori et al., 2023) refers to sequence-level knowledge distillation, where the student model is trained on data generated by the teacher model.
- **MiniLLM** (Gu et al., 2024) employs reverse KL divergence to distill knowledge from the teacher model's logits.

We evaluate our models on the following instruction-following datasets:

- **Dolly**: 500 samples from the `databricks-dolly-15K` dataset used as test set.
- **SelfInst** (Wang et al., 2022a): A user-oriented instruction-following set consisting of 252 samples.
- **Vicuna** (Chiang et al., 2023): The set of 80 difficult questions used for the Vicuna evaluation.
- **S-NI** (Wang et al., 2022b): The SUPER-NATURALINSTRUCTIONS test set comprises 9K samples spanning 119 tasks. Following Gu et al. (2024), we divide it into three subsets based on ground truth response lengths: $[0, 5], [6, 10], [11, +\infty]$ and we use the $[11, +\infty]$ subset.
- **UnNI** (Honovich et al., 2023): The core set of UNNATURALINSTRUCTIONS comprises 60K samples. Following a similar approach to S-NI, we evaluate on a randomly selected subset of 10K examples from the $[11, +\infty]$ range.

D Ablation Study

Table 2: Ablation study: Performance of TASKD-LLM applied to different layers configurations. Results are reported over 3 random seeds. L refers to Last layer, F refers to First layer, and M refers to Middle layer. M for million.

	#params: 120 M					#params: 340 M				
	Dolly	SelfInst	Vicuna	S-NI	UnNI	Dolly	SelfInst	Vicuna	S-NI	UnNI
L	24.31	12.29	17.91	23.07	25.32	**26.02**	14.57	17.68	25.61	30.20
F+L	24.30	12.10	17.55	23.11	25.02	25.83	14.65	17.79	**26.43**	**30.58**
2L	24.50	**12.37**	**17.95**	**23.47**	**25.54**	25.60	14.50	17.71	26.07	30.09
M+L	**24.51**	12.17	17.54	23.19	24.98	25.83	**14.74**	**17.89**	26.34	30.50

Table 3: Ablation study: Comparing different selection approaches. Results are reported over 3 random seeds. Rand refers to random selection of units. Grad is the gradient attribution method for units selection. M for million.

	#params: 120 M					#params: 340 M				
	Dolly	SelfInst	Vicuna	S-NI	UnNI	Dolly	SelfInst	Vicuna	S-NI	UnNI
Grad	24.31	12.29	**17.91**	**23.07**	**25.32**	**26.02**	**14.57**	17.68	**25.61**	**30.28**
Rand	**24.45**	**12.33**	17.58	22.90	25.31	25.55	14.34	**18.02**	24.96	29.68