The Multiple Ticket Hypothesis: Random Sparse Subnetworks Suffice for RLVR

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

The Lottery Ticket Hypothesis demonstrated that sparse subnetworks can match full-model performance, suggesting parameter redundancy. Meanwhile, in Reinforcement Learning with Verifiable Rewards (RLVR), recent work has shown that updates concentrate on a sparse subset of parameters, which further lends evidence to this underlying redundancy. We study a minimal way to exploit this redundancy by training only a randomly selected subset of parameters at extreme sparsities. Empirically, we find that training just 1% of parameters matches or exceeds full-parameter RLVR finetuning across 3 models and 2 task domains. Moreover, different random masks show minimal overlap (≤ 0.005 Jaccard similarity) and yet all succeed, suggesting pretrained models contain many viable sparse subnetworks rather than one privileged set. We term this the *Multiple Ticket Hypothesis*. We explain this phenomenon through the implicit per-step KL constraint in RLVR, which restricts updates to a low-dimensional subspace, enabling arbitrary sparse masks to succeed.

1. Introduction

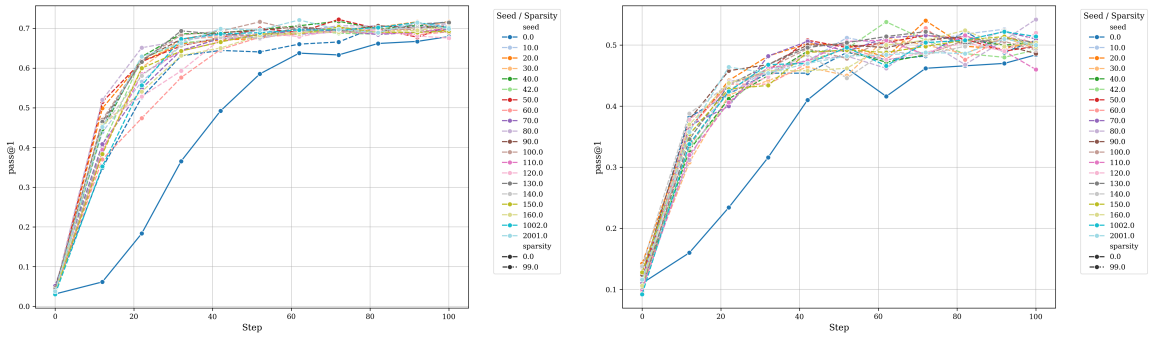
Reinforcement Learning with Verifiable Rewards (RLVR) has become a standard route for improving LLM reasoning, yet recent work shows that RLVR updates concentrate on only a small fraction of model parameters [32, 52]. This raises a simple question. If RLVR already changes sparse parameter subsets, can we choose such a subset randomly before training and update only those parameters?

We show that the answer is yes. Training a fixed random 1% of parameters matches or exceeds full-parameter RLVR finetuning across Qwen2.5-0.5B Base, Qwen2.5-0.5B Instruct, and Qwen2.5-1.5B on mathematical and logical reasoning tasks. More importantly, 20 independent 1% masks all succeed while sharing only about 0.5% Jaccard overlap, so success is not tied to a single privileged subnetwork.

We call this the *Multiple Ticket Hypothesis*. In this view, pretrained LLMs contain many sparse subnetworks that are sufficient for RLVR finetuning, and random sampling at sufficient density reliably finds one. We explain the effect geometrically. RLVR's implicit per-step KL constraint limits policy changes to the Fisher-sensitive directions that matter most. If those directions form a low-dimensional subspace and are spread across parameters, then many different random sparse masks can express useful updates in that same subspace. Appendix J gives the full theoretical outline and proofs.

We make three contributions.

1. We show that random sparse training at $\geq$99% sparsity matches full RLVR finetuning across multiple models and tasks.



GSM8K (20 seeds) MATH-500 (20 seeds)

Figure 1: **Multiple random parameter subsets match or exceed full finetuning at 99% sparsity on Qwen-2.5-1.5B.** Performance of 20 random parameter subsets of Qwen-2.5-1.5B across 100 training steps for GSM8K and MATH-500. 0% sparsity means full parameter finetuning. 99% sparsity indicates that 1% of parameters were trained.

2. We show that many nearly disjoint random masks succeed, supporting the **Multiple Ticket Hypothesis**.
3. We connect this redundancy to the low-dimensional geometry induced by KL-constrained policy optimization.

Appendix roadmap. For convenience, the core appendix sections most relevant to the main paper are:

- Appendix B describes the experimental setup and masking procedure in more detail.
- Appendix C discusses the related work.
- Appendix D expands the discussion, practical implications, and limitations.
- Appendix J states the theoretical assumptions and gives the proofs.

Other appendix figures, tables, and supplementary analyses are cited at the points where they support the corresponding result.

2. Method and Experimental Setup

Random masked RLVR. We use GRPO-style RLVR training [17, 40] in the zero-RL setting, where training starts directly from the pretrained model without supervised fine-tuning and uses verifiable task rewards. For each run, we sample a binary mask once at initialization and keep it fixed. Gradients are computed normally, but only entries selected by the mask update; all other parameters remain at their pretrained values. Masking is applied per parameter tensor so each tensor has approximately the requested keep ratio. Full objective and masking details are in Appendix B.

Models and tasks. We evaluate three Qwen2.5 models, namely 0.5B Base, 0.5B Instruct, and 1.5B [36]. Mathematical reasoning experiments train on Hendrycks MATH and evaluate on MATH-500 and GSM8K [12, 19]. Logical reasoning experiments use Alphabet Sort, a multi-turn name-ordering task. Appendix Table 2 summarizes the model/task configurations, and Appendix E and Table 4 give the full hyperparameters, prompt templates, and evaluation details.

Mask protocol. For the main multi-ticket test, we train 20 independent 99% sparse masks. For sparsity sweeps, we train five masks per sparsity level (seeds 0, 10, 42, 1002, and 2001) and tune learning rates per sparsity. We report validation pass@1 with variation across masks. Active parameter counts and learning rates are listed in Appendix A and Table 3.

3. Results

Many disjoint random masks succeed. At 99% sparsity, random sparse RLVR reaches full-finetuning performance for Qwen2.5-1.5B on GSM8K and MATH-500 (Figure 1) and for Qwen2.5-0.5B-Instruct on Alphabet Sort and GSM8K (Appendix Figure 3). The successful masks have near-random overlap, with mean pairwise Jaccard similarity ≈ 0.005 at 99% sparsity and ≈ 0.0005 at 99.9% sparsity (Appendix Table 1). This rules out a single privileged winning ticket and instead supports many viable tickets.

Performance remains strong until extreme sparsity. Figure 2 sweeps sparsity across tasks and model scales. Random masks match, exceed, or only slightly underperform full finetuning from 99% to 99.95% sparsity (1% to 0.05% trainable parameters). Performance drops sharply around 99.99% and beyond, suggesting a lower bound on the effective trainable dimensionality needed for these RLVR tasks. Appendix H discusses the learning-rate sweep used for these sparse runs.

Random beats simple structured sparsity. At the same 99% parameter budget, random masks outperform first-layer-only and last-layer-only baselines across the tested model/task combinations (Appendix G; Appendix Figures 5 and 6). We also observe more model collapse and variance at the highest sparsities; we treat this as an optimization-stability issue rather than evidence for a unique mask.

4. Theory Sketch

The empirical result is surprising only if parameter identity matters. Our explanation is that RLVR constrains policy movement more than parameter movement. For small updates, the per-step KL constraint has the Fisher form

$$D_{\mathrm{KL}}(\pi_{\theta+\Delta}\|\pi_\theta) \approx \tfrac{1}{2}\Delta^\top F(\theta)\Delta \le K.$$

If the Fisher spectrum has low effective rank, then only the top-r eigenspace changes the policy appreciably; components in the tail have little effect on log-probabilities. If these top eigenvectors are delocalized over coordinates, then a random mask with $k > r$ active coordinates preserves enough of this subspace with high probability. Thus many different sparse masks can realize nearly the same policy-level update, explaining why independent masks succeed despite negligible parameter overlap.

Appendix J states the assumptions and proves two formal claims, low-dimensional policy sensitivity (Proposition 4) and sufficiency of random masks (Proposition 5). The eigenspectrum

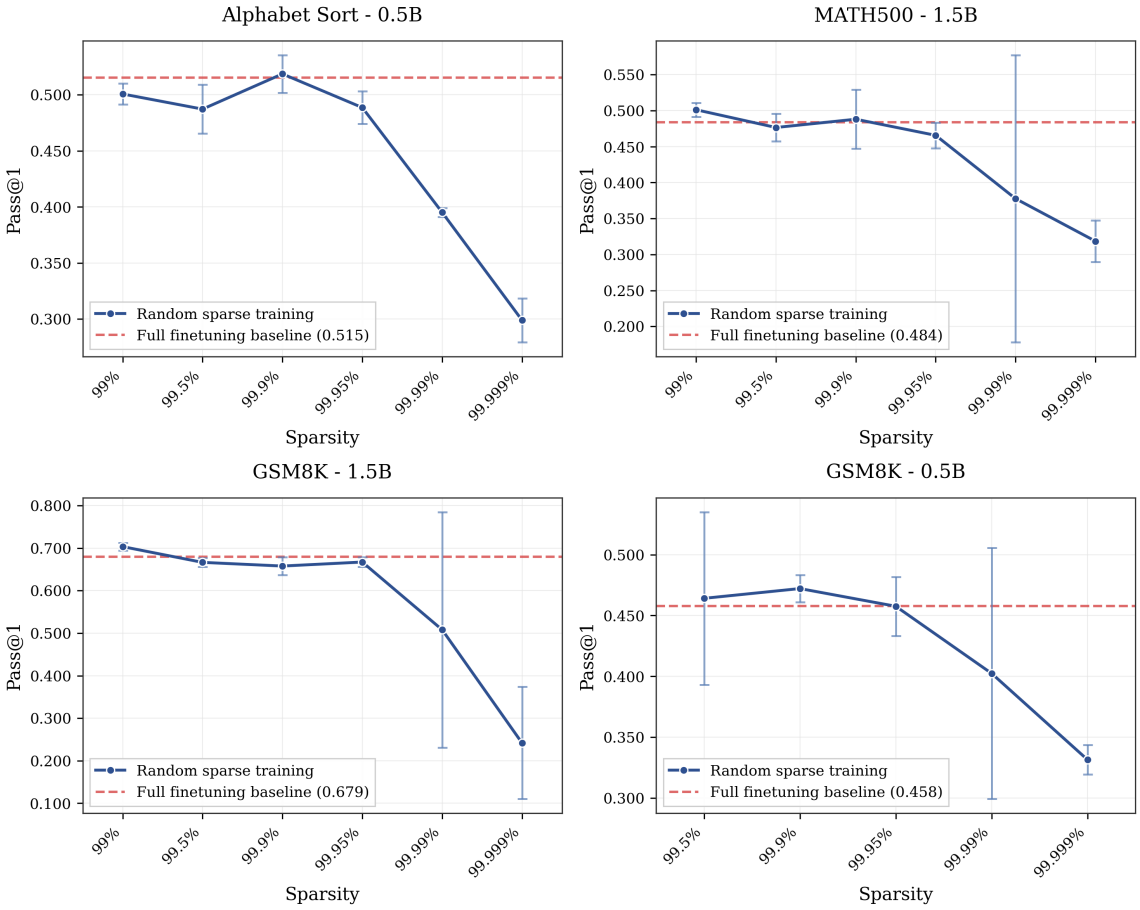


Figure 2: **Random sparse training matches full finetuning at different sparsities.** Validation performance across sparsity levels for three tasks. Error bars show variation across five random masks. Horizontal dashed lines indicate full-parameter baselines. All results use best learning rate from sweep for each configuration.

evidence appears in Appendix F, where Figure 4 shows rapid gradient-eigenspectrum decay with effective rank about 44 for Qwen2.5-0.5B on Alphabet Sort.

5. Discussion and Conclusion

Random sparse RLVR is a strong, simple baseline because training only 1% of parameters can match full finetuning while using substantially less optimizer memory (Appendix I; Table 5). Conceptually, the results suggest that RLVR operates in a low-dimensional policy-relevant subspace of a highly redundant pretrained model. Detailed related work, extended discussion, and limitations are in Appendices C and D.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pages 7319–7328, 2021.

[3] Paul Albert, Frederic Z Zhang, Hemanth Saratchandran, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. Randlora: Full-rank parameter-efficient fine-tuning of large models. *arXiv preprint arXiv:2502.00987*, 2025.

[4] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

[5] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[6] Alan Ansell, Ivan Vulić, Hannah Sterz, Anna Korhonen, and Edoardo M Ponti. Scaling sparse fine-tuning to large language models. *arXiv preprint arXiv:2401.16405*, 2024.

[7] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-short.1. URL https://aclanthology.org/2022.acl-short.1/.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Jingjing Liu, and Zhangyang Wang. The elastic lottery ticket hypothesis. *Advances in Neural Information Processing Systems*, 34: 26609–26621, 2021.

[10] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm:

Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113, 2023.

[11] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[13] Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

[14] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

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

[16] Laura Graesser, Utku Evci, Erich Elsen, and Pablo Samuel Castro. The state of sparse training in deep reinforcement learning. In *International Conference on Machine Learning*, pages 7766–7792. PMLR, 2022.

[17] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou,

Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

[18] Haiyang Guo, Fanhu Zeng, Fei Zhu, Jiayi Wang, Xukai Wang, Jingang Zhou, Hongbo Zhao, Wenzhuo Liu, Shijie Ma, Xu-Yao Zhang, et al. A comprehensive survey on continual learning in generative models. *arXiv preprint arXiv:2506.13045*, 2025.

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

[20] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

[21] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1 (2):3, 2022.

[23] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[24] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, and Jianfeng Gao. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends in Computer Graphics and Vision*, 16(1-2):1–214, 2024.

[25] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.353. URL https://aclanthology.org/2021.acl-long.353/.

[26] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[27] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[28] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

[29] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR, 2020.

[30] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075, 2023.

[31] Sagnik Mukherjee. Who is adam? sgd might be all we need for rlvr in llms. *Manuscript in preparation*, 2025. Available at https://www.notion.so/sagnikm/Who-is-Adam-SGD-Might-Be-All-We-Need-For-RLVR-In-LLMs-1cd2c74770c080de9cbbf

[32] Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models, 2025. URL https://arxiv.org/abs/2505.11711.

[33] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[34] Ashwinee Panda, Berivan Isik, Xiangyu Qi, Sanmi Koyejo, Tsachy Weissman, and Prateek Mittal. Lottery ticket adaptation: Mitigating destructive interference in llms. *arXiv preprint arXiv:2406.16797*, 2024.

[35] Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. Task-specific skill localization in fine-tuned language models. In *International Conference on Machine Learning*, pages 27011–27033. PMLR, 2023.

[36] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

[37] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[38] John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[41] Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[42] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[43] Marc Aurel Vischer, Robert Tjarko Lange, and Henning Sprekeler. On lottery tickets and minimal task representations in deep reinforcement learning. *arXiv preprint arXiv:2105.01648*, 2021.

[44] Xinyi Wang, Shawn Tan, Mingyu Jin, William Yang Wang, Rameswar Panda, and Yikang Shen. Do larger language models imply better reasoning? a pretraining scaling law for reasoning. *arXiv preprint arXiv:2504.03635*, 2025.

[45] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[46] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024.

[47] Jing Xu and Jingzhao Zhang. Random masking finds winning tickets for parameter efficient fine-tuning. *arXiv preprint arXiv:2405.02596*, 2024.

[48] Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*, 2019.

[49] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

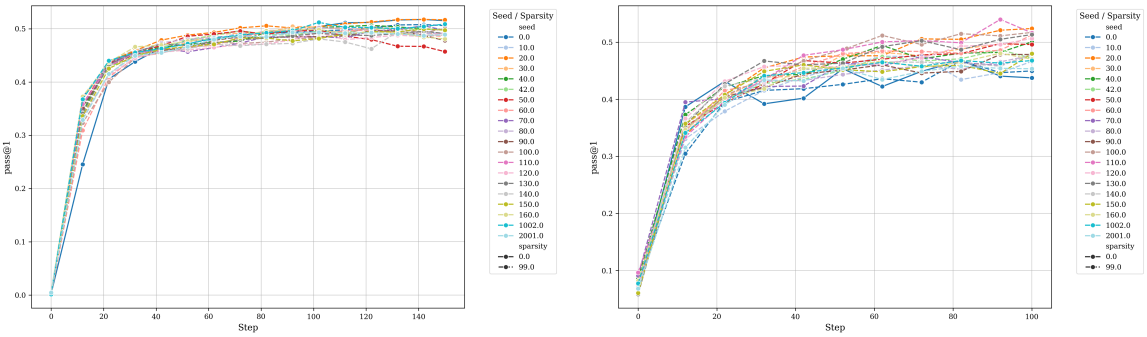
[50] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023.

[51] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[52] Hanqing Zhu, Zhenyu Zhang, Hanxian Huang, DiJia Su, Zechun Liu, Jiawei Zhao, Igor Fedorov, Hamed Pirsiavash, Zhizhou Sha, Jinwon Lee, et al. The path not taken: Rlvr provably learns off the principals. *arXiv preprint arXiv:2511.08567*, 2025.

[53] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendix A. Additional Results and Setup Tables



Alphabet sort (20 seeds) GSM8K (20 seeds)

Figure 3: **Multiple random parameter subsets match or exceed full finetuning at 99% sparsity on Qwen-2.5-0.5B.** Performance of 20 random parameter subsets of Qwen-2.5-0.5B across 100 training steps for GSM8K and 150 steps for Alphabet sort.

Table 1: Jaccard similarity between pairs of successful masks. Values show mean across all mask pairs between the 5 masks for each model. Expected overlap for random masks are also shown.

Configuration	**99%**	**99.9%**	**99.99%**
Qwen2.5-0.5B	0.005	0.0005	0.000055
Qwen2.5-1.5B	0.005031	0.000498	0.0000528
Expected (random)	0.005	0.0005	0.00005

Table 2: Summary of models, training tasks, and evaluation tasks.

MODEL	TRAIN SET	EVAL SET
QWEN2.5-0.5B-IT	ALPHABET SORT	ALPHABET SORT
QWEN2.5-0.5B	GSM8K	GSM8K
QWEN2.5-1.5B	MATH	GSM8K, MATH-500

Table 3: Sparsity levels, active parameter counts, and used learning rates.

Sparsity (%)	Active parameters		Learning rate	
	0.5B	1.5B	0.5B	1.5B
0	$\sim$ 0.5B	$\sim$ 1.5B	1×10^{-6}	1×10^{-6}
99	$\sim$ 4.9M	$\sim$ 15M	1×10^{-4}	1×10^{-4}
99.5	$\sim$ 2.4M	$\sim$ 7.7M	1×10^{-4}	1×10^{-4}
99.9	$\sim$ 490K	$\sim$ 1.5M	1×10^{-3}	1×10^{-3}
99.95	$\sim$ 247K	$\sim$ 770K	1×10^{-3}	1×10^{-3}
99.99	$\sim$ 49K	$\sim$ 150K	1×10^{-3}	4×10^{-3}
99.999	$\sim$ 4K	$\sim$ 15K	5×10^{-3}	1×10^{-2}

Appendix B. RLVR and Masking Details

B.1. Reinforcement Learning with Verifiable Rewards

Following Guo et al. [17], we adopt Group Relative Policy Optimization (GRPO) [40], an on-policy reinforcement learning algorithm that extends Proximal Policy Optimization (PPO) [39] while eliminating the need for a separate value critic.

GRPO estimates advantages using relative rewards within a group of sampled responses. For each prompt x, the current policy π_θ generates G candidate outputs $\{y_1, \ldots, y_G\}$. Their verifiable rewards $\{R_1, \ldots, R_G\}$ are normalized to obtain group-relative advantages $\hat{A}_i = (R_i - \mu)/\sigma$, where μ and σ are the group mean and standard deviation.

The policy is updated by maximizing a clipped surrogate objective that favors higher-reward responses, with a KL-divergence penalty against a reference policy π_{ref}.

$$\mathcal{L}(\theta) = \mathbb{E}_{x,y_i}\Bigg[\min\Big(r_i(\theta)\hat{A}_i,\ \text{clip}\big(r_i(\theta), 1-\varepsilon, 1+\varepsilon\big)\hat{A}_i\Big) - \beta\,\text{KL}\big(\pi_\theta(\cdot|x)\ \|\ \pi_{\text{ref}}(\cdot|x)\big)\Bigg] \tag{1}$$

with $r_i(\theta) = \pi_\theta(y_i|x)/\pi_{\text{ref}}(y_i|x)$ the importance ratio and β controlling regularization strength.

Zero-RL Training. Following [17], we perform "zero-RL" training, starting directly from the pretrained base model without supervised fine-tuning. We also follow [49] in setting $\beta = 0$ and using token-level policy gradients which removes explicit KL regularization.

B.2. Sampling Random Parameters

To sample a random subset of parameters at x% sparsity, we iterate through all layers of the model and sample a random subset of the parameters with a fixed seed.

Per-parameter-tensor masking. Let the model parameters be organized as a collection of tensors in different layers[1].

$$\{\theta^{(l)}\}_{l=1}^{L}.$$

1. In practice, the parameters are organized into tensors which are organized into layers.

For each parameter tensor $\theta^{(l)}$, we independently construct a binary mask

$$m^{(l)} \in \{0,1\}^{\text{shape}(\theta^{(l)})}.$$

Given a target sparsity level $s \in [0,1)$ with keep ratio $p = 1 - s$, we sample

$$k^{(l)} = \lfloor p \cdot |\theta^{(l)}| \rfloor$$

entries of $\theta^{(l)}$ uniformly at random *without replacement* and set the corresponding $k^{(l)}$ entries of $m^{(l)}$ to one, with all remaining entries set to zero.

This procedure results in approximately uniform sparsity across parameter tensors while the total number of active parameters is also approximately at a ratio of p.

B.3. Masked Training

Masks are sampled once at initialization and held fixed throughout training. During training, gradients are computed densely for all parameters. The effective gradient used for optimization is given by

$$\nabla^{\text{masked}}_{\theta^{(l)}} \mathcal{L} = m^{(l)} \odot \nabla_{\theta^{(l)}} \mathcal{L},$$

where $\odot$ denotes elementwise multiplication. Parameters corresponding to zero entries in the mask receive zero gradient updates at all training steps and remain fixed at their initialization values. Optimizer states (e.g., momentum terms) are also maintained only for unmasked parameters.

Appendix C. Related Work

C.1. Post-Training of Large Language Models and RLVR

Pretrained large language models [1, 5, 8, 10, 24, 37, 42] require post-training to achieve strong performance on downstream tasks. The main paradigms include supervised fine-tuning (SFT) [11, 13, 21, 45] and reinforcement learning (RL), often applied sequentially [17, 33, 53].

Recent progress in LLM reasoning [17] shows that Reinforcement Learning with Verifiable Rewards (RLVR) excels in domains with clear verification and correctness checks, such as mathematics, code generation, and logical reasoning. These methods typically build on policy optimization algorithms such as PPO [33] or the more memory-efficient GRPO [40] and variants that further improve them [26, 27, 49, 51].

RLVR's optimization dynamics have drawn attention. Mukherjee et al. [32] showed that despite updating all parameters, RLVR concentrates changes on 5-30% of them. Zhu et al. [52] revealed implicit per-step KL constraints in RLVR even without explicit regularization ($\beta = 0$), distinguishing it from SFT. These observations inspired our explicit sparse training experiments, but unlike their focus on analyzing post-hoc subnetworks, we show that random sparse subnetworks at $\geq$99% sparsity match or exceed full RLVR performance from the start, without prior training or identification.

C.2. Sparsity in Neural Network Training and Lottery Tickets

The Lottery Ticket Hypothesis [15, 29] established that dense networks contain sparse subnetworks matching full-model performance, though identifying these "winning tickets" requires iterative pruning. Subsequent work extended LTH to various domains, including deep reinforcement learning [16, 43, 48], and identified task-specific subnetworks in language models that can mitigate

catastrophic forgetting [34, 35]. Chen et. al [2021] showed that random sparse subnetworks underperformed winning tickets identified by iterative pruning on vision tasks.

In contrast to LTH's emphasis on a single privileged subnetwork identified via pruning, our Multiple Ticket Hypothesis reveals that pretrained LLMs contain many viable sparse subnetworks for RLVR and sampling a random subset of parameters at sufficient density reliably discovers one.

C.3. Parameter-Efficient Fine-Tuning (PEFT)

PEFT methods cut costs for adapting / finetuning large models and numerous studies have established that they operate in constrained parameter subspaces [3, 6, 7, 20, 22, 25, 30, 46, 50].

Of most interest is LoRA [22], which constrains update to low rank matrices i.e it learns the low-rank subspace during training via the adapter module. Schulman, John and Thinking Machines Lab [2025] further showed that rank-1 LoRA matches full RL finetuning. In this work, we sample the subspace, rather than learn it and our focus is on RLVR, while most of the works in this domain have been focused on SFT.

C.4. Sparsity in RLVR

Mukherjee et al. [2025] observed RLVR's intrinsic sparse updates (5-30%), conjecturing post-hoc subnetworks recover performance. Zhu et al. [2025] linked sparsity to model-conditioned bias, emphasizing off-principal updates and spectral preservation. We align with this geometry while making a complementary claim. Our framework, rooted in per-step KL constraints inducing low-dimensional subspaces, explains why arbitrary random masks at $>99\%$ sparsity succeed without prior training. This shifts emphasis from describing sparsity to leveraging redundancy for efficient RLVR, supporting our Multiple Ticket Hypothesis over singular subnetworks.

C.5. Random Sparse Training for Fine-Tuning

Most related, Xu & Zhang [2024] demonstrated random masks at 0.001% trainable parameters match full SFT on NLP (language understanding and comprehension tasks), attributing flatter landscapes (smaller Hessians) and higher learning rates to overparameterization, analyzed via linear regression. Concurrently, Sampreeth et al. explored using expander graph masks instead of random masks for the initial subnetworks.

We extend this to RLVR by showing that RLVR shows greater redundancy, with many independent masks succeeding. Mechanisms differ—SFT's unconstrained updates vs. RLVR's policy gradients, on-policy sampling, and implicit KL constraints [52]—leading to RLVR-specific low-rank Fisher structure from trust regions, not general flat landscapes. We provide empirical multiplicity evidence via Jaccard analysis and test on reasoning tasks with Qwen models, unlike their SFT classification task as RLVR success isn't implied by SFT due to differing dynamics.

C.6. Fisher Information, Policy Optimization Geometry, and Intrinsic Dimensionality

Fisher information aids understanding training dynamics, parameter importance for transfer, and generalization. In policy optimization, natural gradient methods use it for stable updates via metric tensors.

We build on this by showing KL constraints in RLVR create low-rank gradient Fisher matrices, restricting updates to low-dimensional subspaces that enable random sparse training. This geometric view explains mask success and multiplicity.

The success also relates to intrinsic dimensionality. Aghajanyan et al. [2021] showed that fine-tuning needs few parameters despite billions total, aligning with overparameterization theory [4, 14]. Our framework applies this to RLVR, where KL induces policy-relevant low-dimensional subspaces, and delocalization lets random masks span them. The Multiple Ticket Hypothesis follows from trust-region methods in overparameterized networks.

Appendix D. Extended Discussion and Limitations

Our investigation into random sparse training for Reinforcement Learning with Verifiable Rewards (RLVR) suggests that pretrained language models contain combinatorially many viable subnetworks capable of matching full-parameter performance. This *Multiple Ticket Hypothesis* (MTH) fundamentally shifts our understanding of parameter redundancy in the RLVR regime.

RLVR optimization We showed in Section 4 that the the Fisher is low-dim in RLVR. We conjecture that RLVR is locally optimizing a flat loss landscape. This intuition is further surported by preliminary experiments from Mukherjee et al., [2025] where they showed that even simple optimizers like SDG also match and outperform optimizers such as Adam(W).

These findings establish random sparse training as a strong baseline for parameter-efficient RLVR and suggest new directions for understanding how reinforcement learning interacts with pretrained language model representations.

A potential point of contention is the fact that Mukherjee et al., [2025] claim that the updates during RLVR are nearly full rank. We empirically show that the gradients are effectively low-rank. This distinction comes from the fact that first, Mukherjee et al., [2025] estimate the rank from $\Delta = \theta_{final} - \theta_{init}$, while we estimate effective rank from the gradients (Appendix F). It's also possible that a matrix is nearly full rank, but has low effective rank.

Practical Implications for Efficiency. Beyond its theoretical interest, the MTH offers immediate practical benefits for RLVR research. By training only 1% of parameters, researchers can significantly reduce the memory footprint of optimizer states and gradients, enabling the finetuning of larger models on consumer-grade hardware or the use of larger batch sizes. Unlike methods like LoRA, which require learning a low-rank adapter, random sparse training utilizes the model's existing weights directly, acting as a highly efficient, unstructured Parameter-Efficient Fine-Tuning (PEFT) baseline.

Redundancy and Pretraining. Crucially, our findings highlight that this redundancy is a byproduct of the pretraining process itself. The success of random masks when training from scratch suggests that pretraining "delocalizes" knowledge across the parameter space, creating the very landscape that RLVR subsequently navigates.

D.1. Limitations and Future Work

While the Multiple Ticket Hypothesis provides a robust framework for understanding RLVR sparsity, several limitations remain.

- **Catastrophic Forgetting.** Catastrophic forgetting has been explored in deep learning. Specifically in generative AI, skills acquired during pretraining are lost during subsequent finetuning stages [18, 23, 28, 41].

  It's been established that RLVR forgets less than SFT and we tie this to Zhu et al., [2025]'s work, as Zhu et al., [2025] showed that naturally, RLVR chooses low principal weight directions.

  We conjecture that using random sparse mask for RLVR training is likely to lead to more catastrophic forgetting because random sampling doesn't guarantee that principal weights aren't selected and put under RLVR's optimization pressure, which is likely to lead to catastrophic forgetting. We leave further exploration of this to future work.

- **Task Complexity and Sparsity Thresholds.** We observed a consistent performance collapse when trainable parameters dropped below $\sim$0.01%. While this threshold held across our reasoning tasks, more complex, multi-domain and longer horizon tasks (which is what the bulk of RLVR in practise is used for today) might require a higher "intrinsic dimensionality" and thus a lower maximum sparsity.

- **Model Scale and Sparse Architectures.** Our experiments were conducted on dense models up to 1.5B parameters. While the MTH appears to hold as dense model size increases, further validation on frontier-scale models (e.g., 70B+) is necessary to confirm if the ratio of "winning tickets" remains constant or grows with scale. We conjecture however that the MTH findings will hold for larger dense models as increasingly larger models are more overparameterized [44] and they would exhibit more parameter redundancy. A significant additional limitation is that we do not evaluate mixture-of-experts (MoE) models, which already implement sparse computation by activating only a subset of experts per token. This makes MoEs a qualitatively different comparison point: our method adds sparsity by restricting which parameters are trainable, whereas MoEs already restrict which parameters participate in each forward pass. It is therefore unclear how random trainable masks should be implemented for MoEs—for example, whether masks should be sampled globally, per expert, only over routed experts, or jointly with the router—and whether such masks would complement or interfere with expert specialization, routing, and load balancing.

- **Stability and Model Collapse.** We noted a higher frequency of model collapse at extreme sparsities. This suggests that while viable tickets exist at 99.9% sparsity, the optimization path to find them becomes increasingly narrow and sensitive to hyperparameter choices like learning rate and is deserving of more attention.

Appendix E. Complete Experimental Setup

E.1. Hyperparameters

For alphabet sort, max number of turns and min number of turns are set to 2.

E.2. Prompt Template

Mathematical reasoning

We use the following instruction template for all training and evaluation rollouts; only the task-specific instruction changes.

Table 4: Hyperparameters table

Hyperparameter	Training Setup		
	0.5B (Alphabet Sort)	0.5B (GSM8k)	1.5B (Maths)
training steps	150	100	100
batch size	512	512	512
num rollouts	16	16	8
max tokens	128	1024	2048
sampling temperature	1.0	1.0	1.0
Optimizer	AdamW	AdanW	AdamW
clip ratio	0.2	0.2	0.2
KL constant	0	0	0
weight decay	0.01	0.01	0.01
max gradient norm	1.0	1.0	1.0
training seed	42	42	42
eval interval	10	10	10
eval sampling temperature	0.7	0.9	0.9
eval max tokens	128	1024	1024
eval metric	pass@1	pass@1	pass@1
num eval samples	1024	1319	1319
eval seed	2001	-	

```
SYSTEM:
You are a helpful mathematical AI assistant. Please reason step by step
and put your final answer within \boxed{}

USER:
[TASK_DESCRIPTION]

ASSISTANT:
```

Alphabet sort We do not use any system prompt template for the Alphabet sort task. The dataset itself already contains instructions.

```
SYSTEM:

USER:
Sort the following names in alphabetical order:
[List of names]

ASSISTANT:
```

E.3. Additional details on Masks training

The Qwen models have the output projection head tied to the embedding layer; we sample masks once for the embedding layer and reuse them during projection.

Appendix F. Eigenspectrum Analysis Methodology

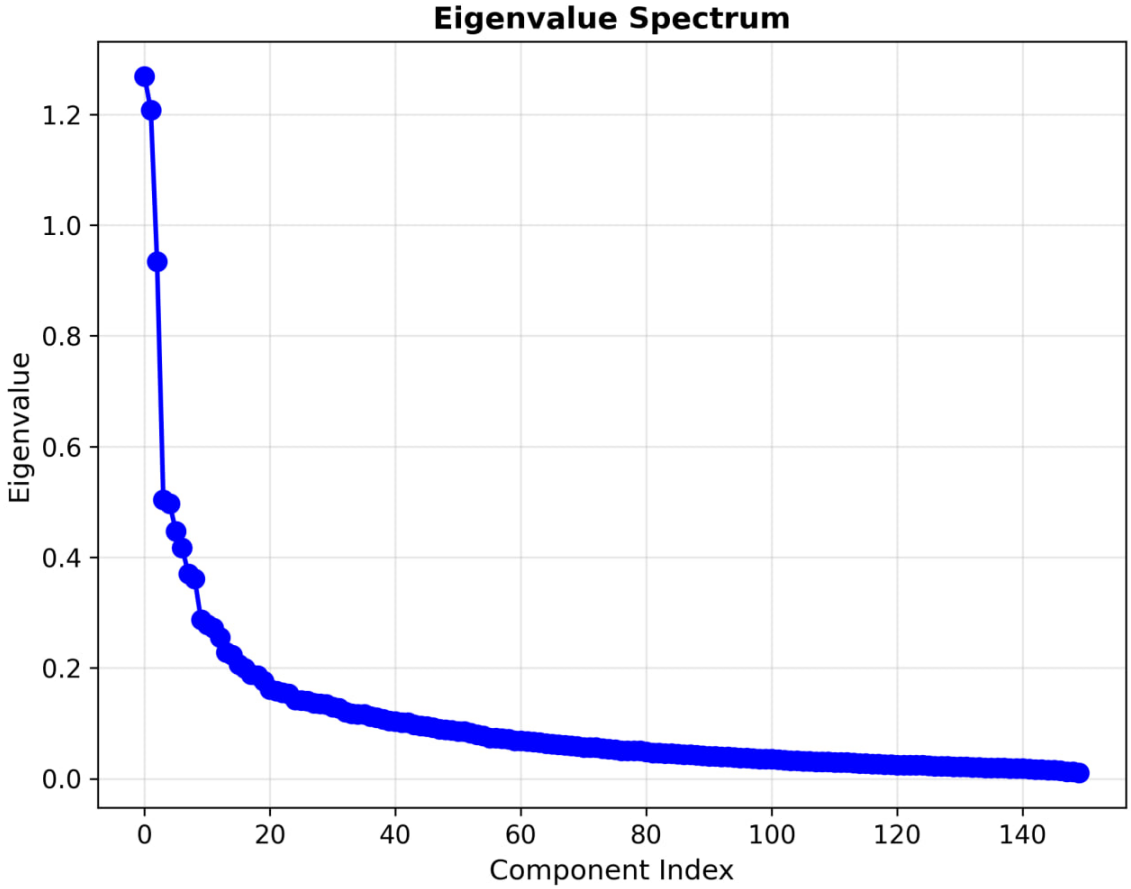


Figure 4: Eigenspectrum analysis of the gradients.

To compute the eigenspectrum of the gradient Fisher information matrix (Figure 4), we used the following procedure.

1. **Gradient collection:** Using the Qwen2.5-0.5B model on the Alphabet Sort task, we ran 150 training steps and saved the complete gradient vector at each step.

2. **Gradient matrix construction:** We flattened each gradient tensor into a 1D vector of dimension $d \approx 490{,}000{,}000$ (corresponding to the total number of model parameters). Stacking all 150 gradient vectors produced a matrix $G \in \mathbb{R}^{150\times 490M}$.

3. **Gram matrix computation:** Rather than computing the full Fisher matrix $F = G^\top G \in \mathbb{R}^{490M\times 490M}$ (which would be computationally infeasible), we computed the Gram matrix $GG^\top \in \mathbb{R}^{150\times 150}$.

4. **Eigendecomposition:** We performed eigenvalue decomposition on $GG^\top$ to obtain the eigenspectrum. The non-zero eigenvalues of $GG^\top$ are identical to those of $G^\top G$, allowing us to characterize the effective rank of the gradient space.

This procedure reveals that the gradient updates lie in a low-dimensional subspace, as evidenced by the rapid eigenvalue decay shown in Figure 4. The top few eigenvalues capture most of the variance, supporting Assumption 1 (low effective rank) in our theoretical framework.

Appendix G. Baselines

We run structured sparsity baselines experiments (first and last layer) against full parameter finetune and random seed (seed = 0).

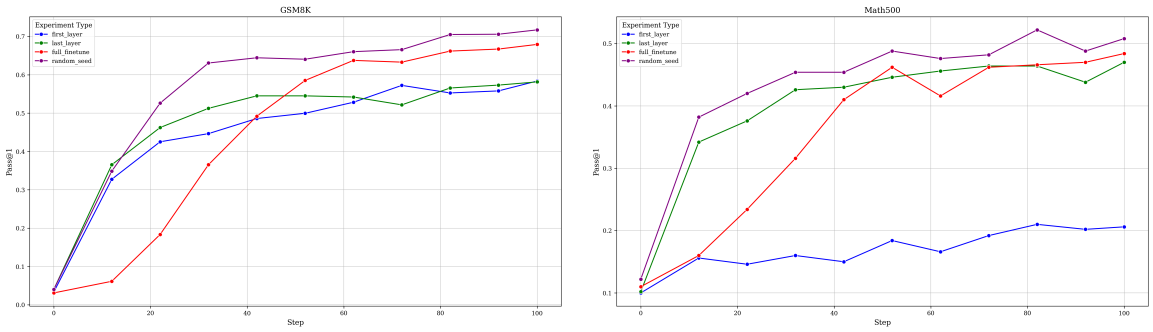


Figure 5: **Comparison of random sparse training, full parameter finetuning and structured sparsity training on Qwen-2.5-1.5B.**

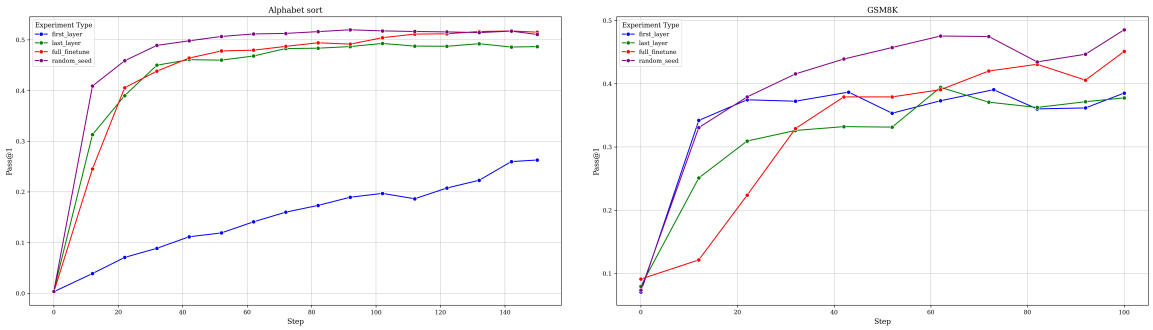


Figure 6: **Comparison of random sparse training, full parameter finetuning and structured sparsity training on Qwen-2.5-0.5B (Instruct / Base).**

Appendix H. Learning Rate Puzzle

As noted by Xu & Zhang, [2024] and evident from Table 3, we also observe that at increasing sparsities, higher learning rates are needed for the performance to match full RLVR finetuning.

Figure 7 shows the learning rate sweep across multiple masks on Alphabet sort task, for Qwen2.5-0.5B-Instruct.

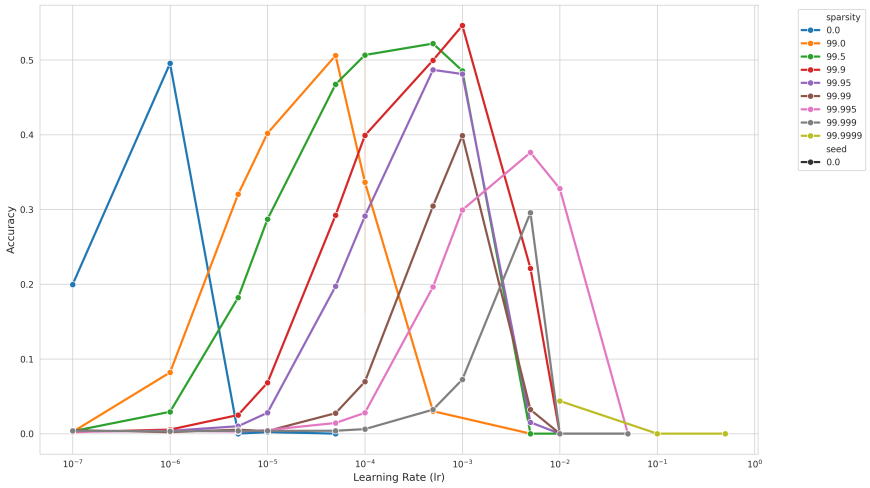


Figure 7: **Learning-rate sweep on Alphabet Sort for Qwen2.5-0.5B-Instruct.** Validation performance across sparsity levels and learning rates. All the masks for the various sparsities are seeded to 0. See the Appendix for the lr. sweep for the other seeded random masks.

Appendix I. Memory Savings Across Sparsities

We update and track optimizer state for only the parameters being updated in a particular run. In the table below, we show the memory savings from tracking only the parameters being updated.

Table 5: Optimizer memory footprint during training (MiB). Full finetuning shown for reference.

Model	Full	99%	99.5%	99.9%	99.95%	99.99%	99.999%
Qwen2.5-1.5B	11776	235.8	118	23.8	12	2.5	0.39
Qwen2.5-0.5B	3769.5	75.6	37.86	7.67	3.9	0.9	0.2

Appendix J. Theoretical Proofs

This appendix provides complete proofs for the theoretical results stated in Section 4.

J.1. Detailed Assumptions and Justifications

We restate the assumptions from the main text with full justification.

Assumption 1 (Low Effective Rank) *There exists a small integer $r \ll d$ and a small constant $\epsilon > 0$ satisfying*

$$\frac{\sum_{i=1}^{r} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \geq 1 - \epsilon.$$

Justification. This assumption states that the top r eigenvectors capture nearly all the "energy" of the Fisher matrix. The empirical eigenvalue spectrum (Figure 4), showing rapid decay after the first few components, directly supports this assumption. Details of the eigenspectrum computation are provided in Appendix F.

Assumption 2 (Delocalization of Eigenvectors) *There exists a constant $\mu > 0$ such that each eigenvector v_i satisfies*

$$\|v_i\|_\infty \leq \frac{\mu}{\sqrt{d}}.$$

Justification. This condition states that no single parameter dominates an eigenvector; the eigenvector's mass is spread across many coordinates. This is a common property in large random matrices and is empirically plausible for well-trained neural networks, where gradient information is typically distributed across many parameters rather than concentrated in a few.

Assumption 3 (Small-Step Regime) *The per-step update Δ satisfies $\|\Delta\| = O(\sqrt{K})$, where K is the KL bound.*

Justification. This ensures that second-order Taylor expansions are accurate and higher-order terms are negligible. In practice, the clipping mechanism in PPO/GRPO and the on-policy sampling procedure naturally enforce small per-step policy changes.

J.2. Proof of Proposition 4 (Low-Dimensional Policy Sensitivity)

Proposition 4 (Low-Dimensional Policy Sensitivity) *Under Assumptions 1 and 3, for any update Δ satisfying the per-step KL constraint $D_{\mathrm{KL}}(\pi_{\theta+\Delta} \| \pi_\theta) \leq K$, the second-order change in the policy depends only on the projection of Δ onto the subspace $U = \mathrm{span}\{v_1, \ldots, v_r\}$. Components orthogonal to U have negligible impact on the policy.*

Proof We proceed in five steps.

Step 1. KL Constraint in Quadratic Form. Using a second-order Taylor expansion and the definition of the Fisher matrix, the KL divergence can be approximated as

$$D_{\mathrm{KL}}(\pi_{\theta+\Delta}\|\pi_\theta) = \frac{1}{2}\Delta^\top F(\theta)\Delta + O(\|\Delta\|^3).$$

Under the small-step regime (Assumption 3), the cubic term is negligible, so the constraint is essentially

$$\Delta^\top F(\theta)\Delta \le 2K. \tag{$\star$}$$

Step 2. Decomposition of the Update. Decompose Δ into two orthogonal components

$$\Delta = \Delta_\parallel + \Delta_\perp,$$

where $\Delta_\parallel \in U$ and $\Delta_\perp \in U^\perp$ (the orthogonal complement, spanned by $v_{r+1}, \ldots, v_d$). Substitution into ($\star$) gives

$$\Delta^\top F(\theta)\Delta = \Delta_\parallel^\top F(\theta)\Delta_\parallel + \Delta_\perp^\top F(\theta)\Delta_\perp \le 2K.$$

Step 3. Bounding the Contribution of $\Delta_\perp$. Since $\Delta_\perp$ lies in the span of the tail eigenvectors,

$$\Delta_\perp^\top F(\theta)\Delta_\perp = \sum_{i=r+1}^{d} \lambda_i \langle \Delta_\perp, v_i\rangle^2 \le \lambda_{r+1}\|\Delta_\perp\|^2,$$

where λ_{r+1} is the largest eigenvalue in the tail. By Assumption 1, λ_{r+1} is very small relative to the total variance. More precisely, let $\Lambda_{\mathrm{tail}} = \sum_{i=r+1}^{d} \lambda_i$. Then

$$\lambda_{r+1} \le \frac{\Lambda_{\mathrm{tail}}}{d-r} \le \frac{\epsilon \cdot \mathrm{Tr}(F(\theta))}{d-r}.$$

Since $\mathrm{Tr}(F(\theta))$ is typically $O(d)$ in neural networks, $\lambda_{r+1} = O(\epsilon)$. Therefore, even if $\|\Delta_\perp\|^2$ is as large as $O(K/\lambda_{\min})$ (where $\lambda_{\min}$ is the smallest eigenvalue), the product $\lambda_{r+1}\|\Delta_\perp\|^2$ remains $O(\epsilon K/\lambda_{\min})$. Given that ϵ is small and $\lambda_{\min}$ is not extremely small in practice, this term is negligible compared to the KL budget K.

Step 4. Policy Change Depends Primarily on $\Delta_\parallel$. Consider the change in log-probability for a specific output y.

$$\log \pi_{\theta+\Delta}(y|x) - \log \pi_\theta(y|x) = \Delta^\top g(y) + \frac{1}{2}\Delta^\top H(y)\Delta + O(\|\Delta\|^3),$$

where $g(y) = \nabla_\theta \log \pi_\theta(y|x)$ and $H(y) = \nabla_\theta^2 \log \pi_\theta(y|x)$. The expected square of the linear term is $\Delta^\top F(\theta)\Delta$, which we have already bounded. The linear term decomposes as

$$\Delta^\top g(y) = \Delta_\parallel^\top g(y) + \Delta_\perp^\top g(y).$$

The variance of the second term is

$$\mathbb{E}_{y\sim\pi_\theta}\left[(\Delta_\perp^\top g(y))^2\right] = \Delta_\perp^\top F(\theta)\Delta_\perp,$$

which is negligible as argued above. Moreover, since $g(y)$ lies in the span of the Fisher eigenvectors (by definition of $F(\theta)$), the component $\Delta_\perp^\top g(y)$ is only excited by tail eigenvectors, which have small eigenvalues and hence small typical magnitudes. Therefore, the change in log-probability—and thus the policy itself—is dominated by $\Delta_\parallel$.

Step 5. Conclusion. To second order, the policy update depends only on $\Delta_{\|}$. The orthogonal component $\Delta_{\perp}$ neither significantly affects the KL divergence nor the policy output. This establishes that the policy-relevant subspace is effectively low-dimensional. ■

J.3. Proof of Proposition 5 (Sufficiency of Random Masks)

Proposition 5 (Sufficiency of Random Masks) *Let $S \subset \{1, \dots, d\}$ be a random subset of indices of size k, chosen uniformly. Under Assumptions 1 and 2, if $k > r$, then with high probability there exists an update Δ_S supported on S (i.e., $\Delta_{S,i} = 0$ for $i \notin S$) satisfying*

$$\|\Delta_{\|} - \Delta_S\|_F \leq \eta,$$

where $\|\cdot\|_F$ denotes the Fisher norm $\|u\|_F = \sqrt{u^\top F(\theta) u}$, and η is a small constant that decreases as k increases. Consequently, optimizing only over parameters in S can achieve policy improvement equivalent to full-parameter optimization within the KL-reachable region.

Proof We proceed in six steps.

Step 1. Setup and Notation. Let $V_r = [v_1, \dots, v_r] \in \mathbb{R}^{d \times r}$ be the matrix whose columns are the top r eigenvectors. Any vector in U can be written as $V_r c$ for some coefficient vector $c \in \mathbb{R}^r$. Let P_S be the projection operator that zeros out coordinates not in S, so $(P_S u)_i = u_i$ if $i \in S$, and 0 otherwise.

Step 2. Goal. We want to approximate a given $\Delta_{\|} = V_r c$ by a vector Δ_S supported on S. Equivalently, we want to find coefficients $c' \in \mathbb{R}^r$ such that $\Delta_S = P_S(V_r c')$ is close to $\Delta_{\|}$ in Fisher norm.

Step 3. Delocalization and Random Masks. By Assumption 2, each eigenvector v_i has bounded infinity norm. This delocalization property implies that when we sample a random subset S of coordinates, the restricted vectors $\tilde{v}_i = P_S(v_i)$ are likely to preserve the geometric structure of the original subspace.

Formally, consider the matrix $\tilde{V}_r = P_S(V_r) \in \mathbb{R}^{d \times r}$ (which has zeros in rows outside S). The product $\tilde{V}_r^\top F(\theta) \tilde{V}_r$ measures how well the restricted eigenvectors capture the Fisher metric on the subspace. Because the eigenvectors are delocalized, each row of V_r has small norm. A standard concentration argument (see Lemma 6 below) shows that with high probability,

$$\left\| \frac{d}{k} \tilde{V}_r^\top F(\theta) \tilde{V}_r - V_r^\top F(\theta) V_r \right\|_2 \leq \delta,$$

where δ decreases with k. Since $V_r^\top F(\theta) V_r = \mathrm{diag}(\lambda_1, \dots, \lambda_r)$ is diagonal with large entries, the restricted Gram matrix is also well-conditioned when k is sufficiently larger than r.

Step 4. Existence of a Good Approximation. Because the restricted Gram matrix is well-conditioned, the linear map $c' \mapsto P_S(V_r c')$ is injective on $\mathbb{R}^r$. Thus, for any desired $\Delta_{\|} = V_r c$, we can solve the least-squares problem

$$\min_{c' \in \mathbb{R}^r} \|V_r c - P_S(V_r c')\|_F.$$

The solution satisfies

$$\|V_r c - P_S(V_r c')\|_F \leq \kappa \cdot \|V_r c\|_F,$$

where κ depends on the condition number of the restricted Gram matrix. As k increases, $\kappa \to 0$. Setting $\Delta_S = P_S(V_r c')$ yields the required approximation.

Step 5. Connection to Policy Improvement. Since the policy change depends continuously on the update (as shown in Proposition 4), and since the Fisher norm dominates the change in log-probabilities, a small error η in Fisher norm translates to a small error in policy improvement. Therefore, optimizing over the mask S can achieve essentially the same policy improvement as full-parameter optimization, provided $k > r$.

Step 6. Threshold Effect. The quality of approximation undergoes a phase transition. When $k < r$, the restricted Gram matrix becomes singular, and approximation fails. When $k > r$, the error decreases as k increases. This explains the empirical observation that random masks work well above a certain sparsity threshold. ■

J.4. Technical Lemmas

Lemma 6 (Concentration of Restricted Gram Matrix) *Let $V_r \in \mathbb{R}^{d\times r}$ have orthonormal columns with $\|v_i\|_\infty \leq \mu/\sqrt{d}$. Let S be a random subset of size k. Then with probability at least $1-\delta$,*

$$\left\| \frac{d}{k} \tilde{V}_r^\top \tilde{V}_r - I_r \right\|_2 \leq C\mu^2 r \sqrt{\frac{\log(1/\delta)}{k}},$$

where $\tilde{V}_r = P_S(V_r)$ and C is an absolute constant.

Proof sketch. This follows from matrix Bernstein inequalities applied to the sum of independent random matrices $X_j = \frac{d}{k}\mathbf{1}_{j\in S}(V_r)_{j:}^\top (V_r)_{j:}$, where $(V_r)_{j:}$ is the j-th row of V_r. The delocalization assumption ensures each term has bounded norm $\|X_j\|_2 \leq \frac{d}{k}\cdot\frac{\mu^2}{d} = \frac{\mu^2}{k}$. Applying matrix Bernstein with variance proxy $\sigma^2 = O(r/k)$ yields the stated bound.

Remark on the Concentration Bound. The proof relies heavily on the delocalization assumption (Assumption 2). While this is plausible for large neural networks, it is difficult to verify rigorously. However, empirical studies of eigenvectors in trained networks often show diffuse weight distributions, supporting this assumption. Additionally, the concentration bound requires $k = \Omega(r \log r)$, which is consistent with our empirically observed sparsity threshold.

J.5. Synthesis of the Random-Mask Argument

Propositions 4 and 5 together explain the empirical success of random sparse fine-tuning in RLVR.

1. **KL constraints create a low-dimensional trust region.** The per-step KL bound restricts updates to a region defined by the Fisher matrix's quadratic form.

2. **The Fisher matrix has low effective rank.** Due to Assumption 1, the policy-relevant subspace is only r-dimensional, where $r \ll d$.

3. **Delocalization enables random sampling.** Due to Assumption 2, a random mask of size $k > r$ captures this subspace with high probability.

4. **Multiple masks succeed.** The combinatorial number of ways to choose k parameters from d—each capable of spanning the same r-dimensional subspace—directly yields the Multiple Ticket Hypothesis.

J.6. Connection to Neural Tangent Kernel Theory

Our theoretical framework connects to Neural Tangent Kernel (NTK) theory. In the infinite-width limit, neural networks operate in a "lazy training" regime where the kernel remains approximately constant. While our setting involves finite-width networks with potentially evolving representations, the low effective rank of the Fisher matrix suggests a similar phenomenon in which the policy-relevant directions are determined early and remain stable, allowing arbitrary parameter subsets to navigate this low-dimensional landscape.

The delocalization assumption (Assumption 2) is particularly natural in the NTK regime, where eigenvectors of the kernel matrix tend to be spread across many input dimensions rather than localized. This provides theoretical grounding for our empirical observation that random masks work across different model architectures and scales.