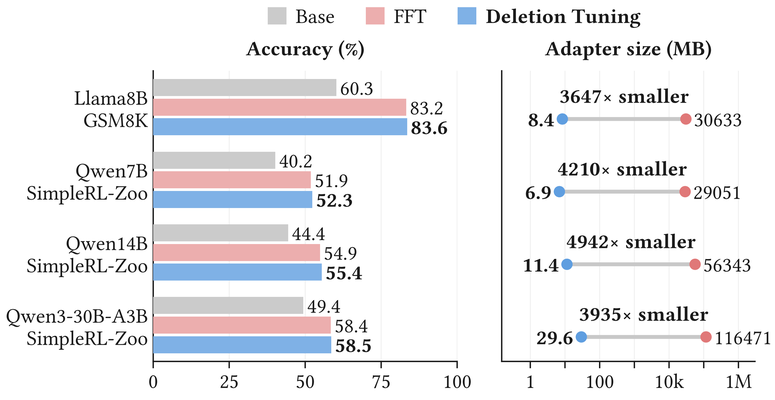
The Surprising Effectiveness of Deleting Weights in LLM Post-Training

Jack Lu[1*], Zhenbang Yang[1*], Mike Lasby[2], Tejas Pote[2], Yani Ioannou[2], Mengye Ren[1]
[1]New York University, [2]University of Calgary ([*]Equal contribution)

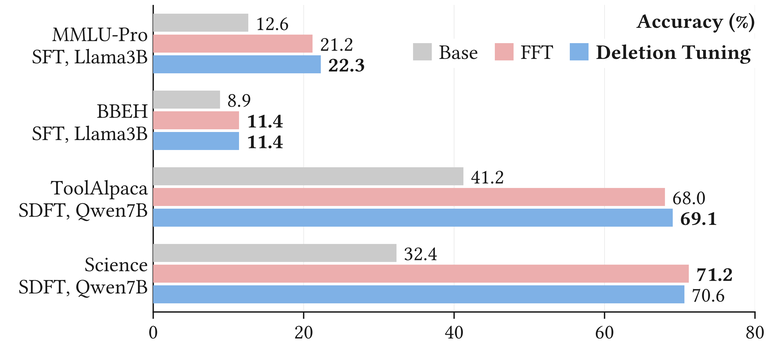
Matches FFT Under RLVR at the Largest Backbones

From 7B to 30B, dense and MoE, with adapters thousands of times smaller.



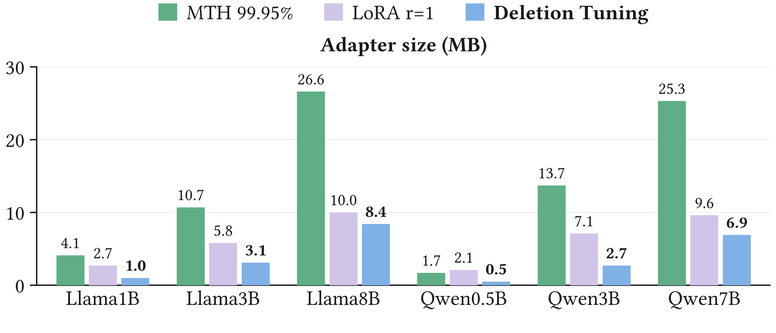
Even Matches FFT Under SFT and SDFT

SFT at 3B and on-policy distillation (SDFT) at 7B, across four benchmarks.



Smallest Adapter at Every Scale

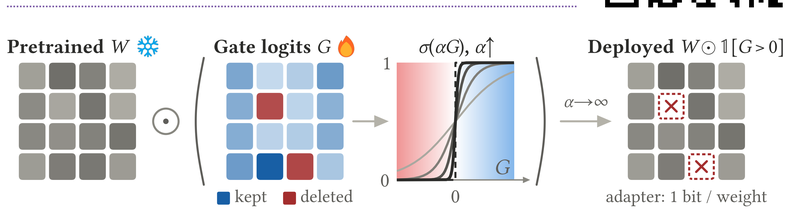
Smaller than even LoRA r=1 and MTH 99.95% (random sparse fine-tuning).



Can deleting <0.05% of weights unlock full post-training performance?

Yes! Deletion Tuning **learns which pretrained weights to delete**, writes no new values, and matches full fine-tuning (FFT) at scale.

Post-Training | Reasoning | Continual Learning | Sparsity

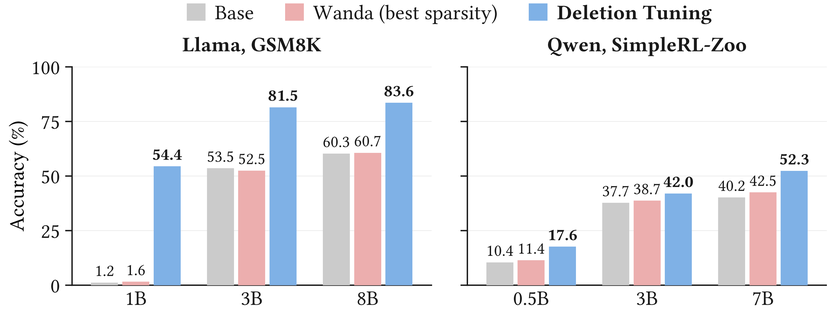


Takeaways

- **Performance**: matches FFT at scale in three post-training regimes
- **Storage**: a compressed bitmask, the smallest adapter at every scale
- **Retention**: keeps prior-task accuracy while the other methods drift
- **Mechanism**: learns weight updates closest to FFT's geometry

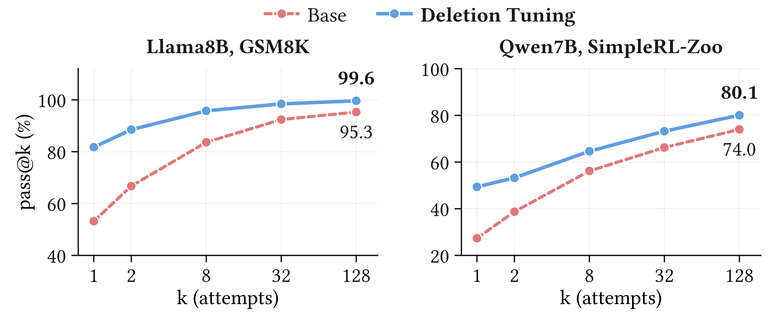
Learning the Mask Drives the Gains

Wanda pruning (the initialization) stays near the base; Deletion Tuning gains up to 53.2 points.



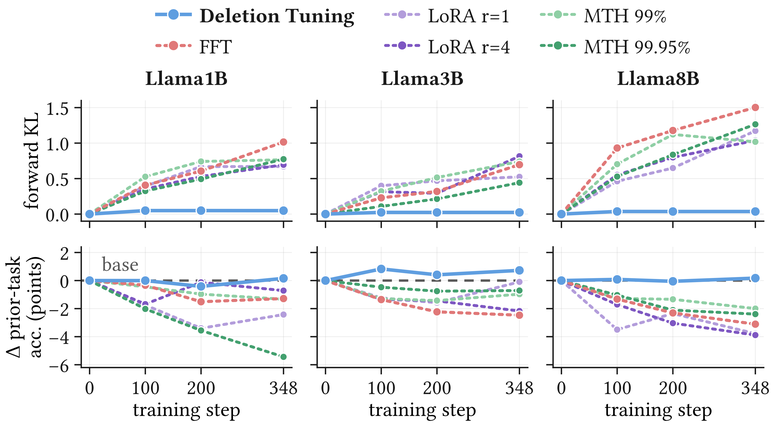
Acquires New Capabilities

Solves problems the base model misses in all 128 attempts.



Keeps Prior-Task Accuracy at Every Scale

Stays within one point of the base model, while the baselines lose up to 5.4 points.



Tracks FFT's Update Geometry Most Closely

Avoids the principal weights and barely shifts the spectrum, as FFT does.

