DoRA: Weight-Decomposed Low-Rank Adaptation

Gerardo Montemayor, Chris Fernandes, Rolando Rodríguez, Ishaan Nanal
Cornell University

Introduction and Motivation

- **Fine-tuning** is an essential process for adapting models for specific downstream tasks
  - Updating all model parameters is often computationally infeasible without simplifications in practical settings

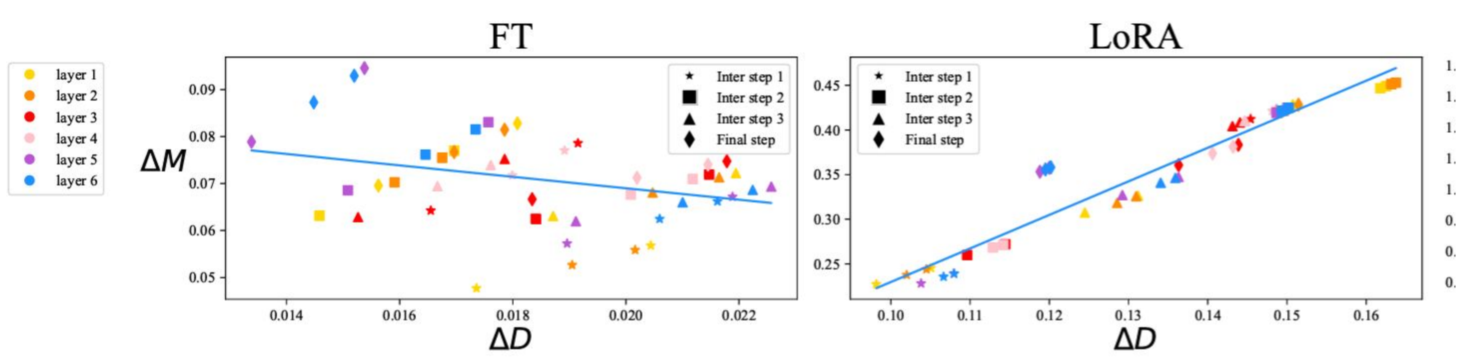
$$W_0 \in \mathbb{R}^{1024\times1024}$$

$$W = W_0 + W' \implies W' \in \mathbb{R}^{1024\times1024}$$

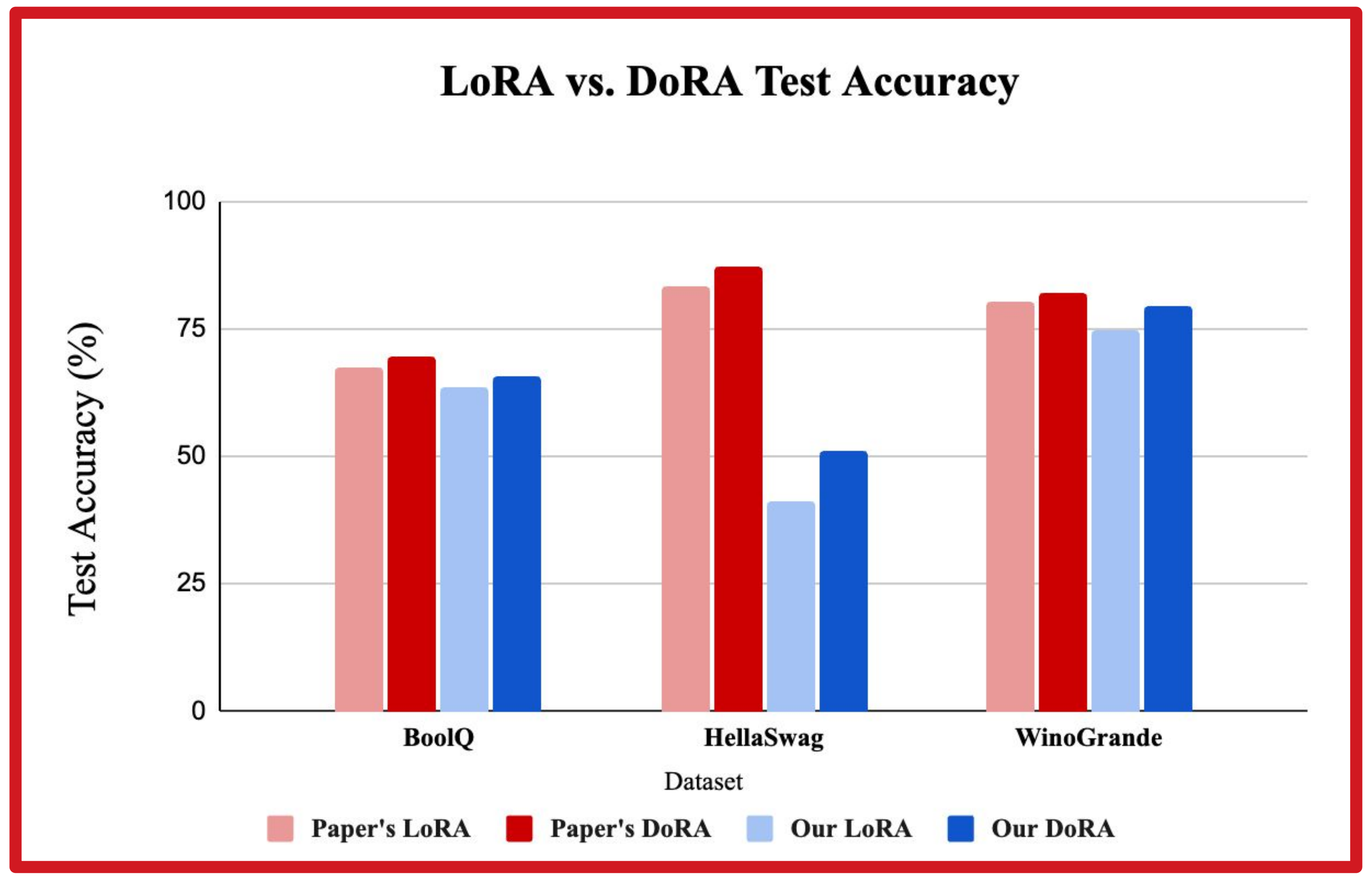
- **Parameter-Efficient Fine-Tuning (PEFT)** techniques reduce resource demands
  - LoRA → freeze weight matrices and add low-rank updates

$$W = W_0 + AB \implies A \in \mathbb{R}^{1024\times r}, B \in \mathbb{R}^{r\times1024}$$

  - We can decrease parameter count by 98% if rank = 10!
- **LoRA vs. DoRA:**
  - LoRA's test accuracy is not as good as full fine-tuning
  - DoRA aims to approximate FT better by decomposing weights into magnitude and direction
- **Our Goal:**
  - **Implement DoRA's low-rank decomposition** and **evaluate against LoRA**



Results



PEFT Strategy	BoolQ	HellaSwag	WinoGrande
Paper's LoRA	67.5	83.4	80.4
Our LoRA	63.5	41.2	74.7
Paper's DoRA	69.7	87.2	81.9
Our DoRA	65.8	51.2	79.5

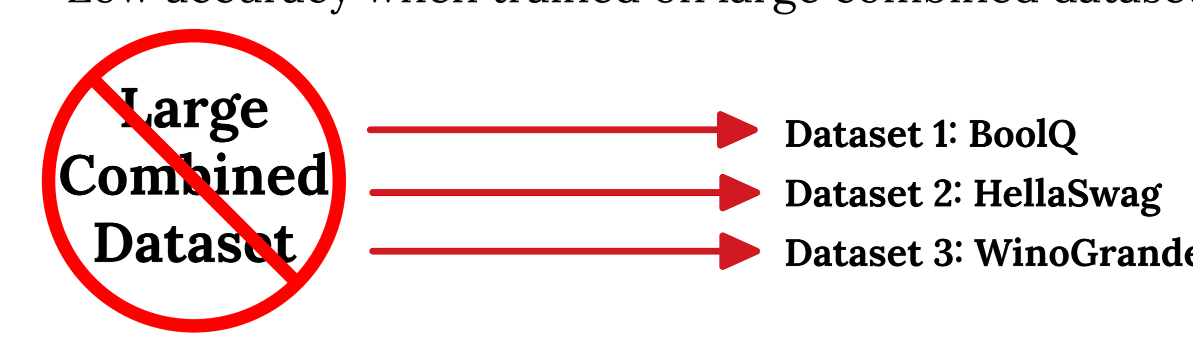
Challenges and Future Work

Future Work

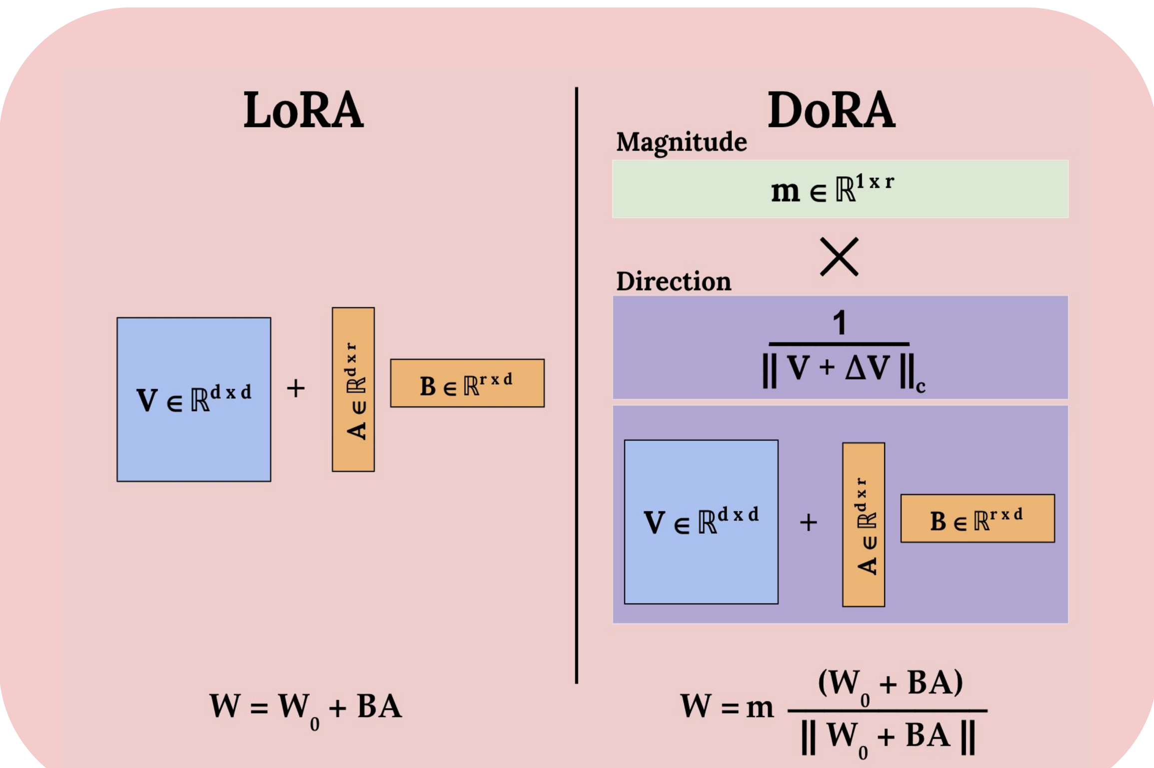
- **Adaptive rank hyperparameterization**
  - Introduce learnable gates that scale only low-rank components
  - The Hessian eigenstructure with respect to the low-rank matrices indicates how large their rank should be to capture meaningful directions of loss curvature
- **Hybrid models**
  - Combining DoRA and other adaptation techniques
  - Ex. Model pruning to make the model even more efficient
  - Does DoRA change how aggressively we can use quantization?

Challenges

- **Memory**
  - A-100's 40GB GPU RAM wasn't enough without dataset simplifications
- **Separated Dataset**
  - Low accuracy when trained on large combined dataset



Methodology



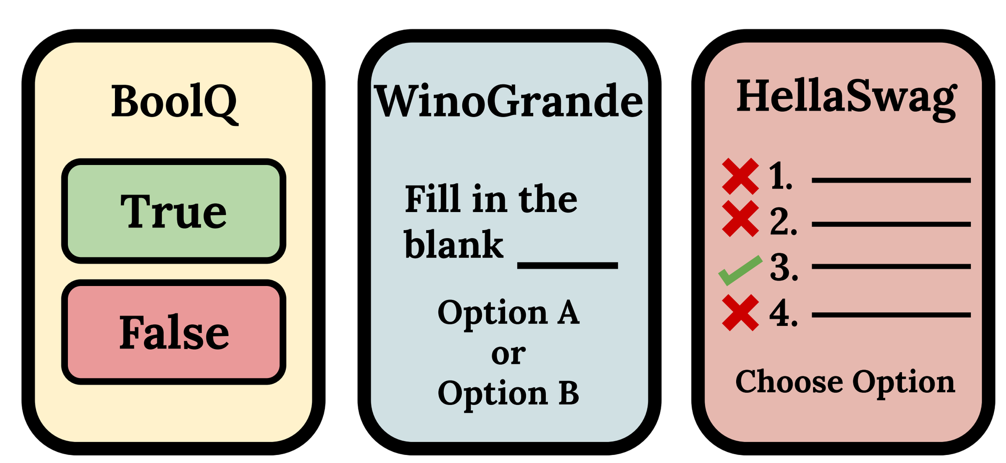
$$W = m\frac{V}{||V||_c} = ||W||_c \frac{W}{||W||_c}$$

$$A \sim \mathcal{N}(0, 0.01), B = 0$$

DoRA Algorithm

1) **Freeze** pre-trained weights W_0
2) **Decompose** W_0 into magnitude vector m and matrix of unit vectors V
3) **Update** V with LoRA: $V' = V + AB$
4) **Compute** $W' = m\frac{V'}{||V'||_c}$

Evaluation



Training Pipeline

Model: LLaMa-7B

+

🤗 **Datasets**

→ BoolQ
→HellaSwag
→ WinoGrande

Modifications

The authors of the paper trained on a large commonsense reasoning dataset, whereas we trained and then evaluated on subtasks of commonsense reasoning one at a time.

Conclusion

- **DoRA outperforms LoRA** while keeping marginal parameter count low
- **Decoupling direction and magnitude** allows gradient updates to optimize both **independently and asymmetrically**
- Training stability and sample efficiency significantly affect the quality of fine-tuning
- Normalization as a remedy for instability can be extended to fine-tuning
- While DoRA adds slightly more parameters than LoRA, the **performance benefits outweigh the costs**

References

Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., & Chen, M.-H. (2024). *DoRA: Weight-Decomposed Low-Rank Adaptation.* arXiv preprint arXiv:2402.09353.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). *LoRA: Low-Rank Adaptation of Large Language* Models. arXiv preprint arXiv:2106.09685