

# TensLoRA + Heterogeneous Allocation of Rank and Learning Rate

---

Pin-Hsuan Chen

December 4th

# Motivation 1: Heterogeneous Learning Rate

"Using the same learning rate for A and B does not allow efficient feature learning." - LoRA+

LoRA+ Approach

$$\eta_B = \lambda \eta_A, \text{ (with } \lambda > 1\text{)}$$

TensLoRA+

$$\eta_C = \lambda \eta_F, \text{ (with } \lambda > 1\text{)}$$

**Note:** Core  $C$  corresponds to  $B$  (due to zero-initialization), while Factors  $F$  correspond to  $A$ .

Hypothesis

Since TensLoRA separates the update into a Core tensor and multiple Factor matrices, applying heterogeneous learning rates may improve optimization efficiency for each component.

→ Therefore, we propose using a Heterogeneous Learning Rate.

# Motivation 2: Heterogeneous Rank

## TensLoRA Insights

- 1 **Rank is Critical:** Rank significantly impacts model performance and parameter efficiency.
- 2 **Dimensional Variance:** Different dimensions within the tensor hold varying degrees of importance.

### GAP

TensLoRA does not experimentally explore how rank should vary across modes, despite explicitly acknowledging mode specific differences.

→ **Therefore, we apply Heterogeneous Rank**

assigning different ranks to different tensor modes based on metrics describing complexity, and contribution.

# Three Phase Experimental Design

## Phase 1: Baseline

### Approaches:

- Standard LoRA
- CP LoRA
- Tucker LoRA

### Metrics Tracked:

#### • SVD\_Entropy:

Information spread.

#### • SVD\_Top1\_Energy:

Contribution magnitude.

## Phase 2: Learning Rate

We apply different  $\lambda$  values to  $\eta C$

## Phase 3: Rank

Rank allocation based on Phase 1 metrics

### High Top1 Energy / Low Entropy:

⬇ Reduce Rank

### Low Top1 Energy / High Entropy:

⬆ Increase Rank

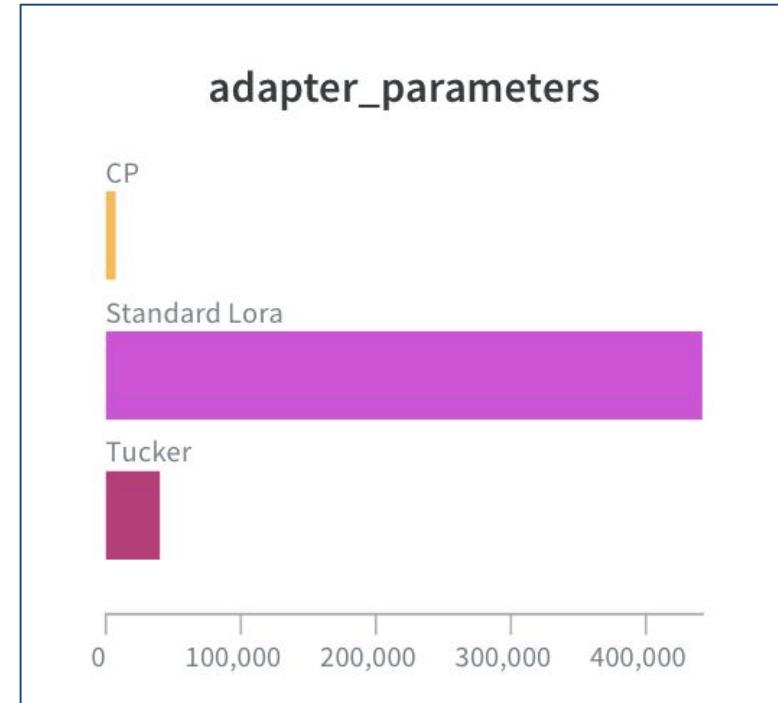
*"Information is concentrated,  
suggests that more rank is required  
to avoid information loss."*

# Phase 1 Results: Baseline Comparison

## Settings

- **Model:** RoBERTa
- **Dataset:** CoLa
- **Main Approach:** Tucker
- **Rank:** 8

Approach	Params	MCC
Standard Lora	442368	0.58
CP	6872	0.52
Tucker	39640	0.52



## ANALYSIS

- CP gives same MCC as Tucker despite smallest parameter budget.
- Perhaps because CoLA is a low capacity task, so aggressive compression does not hurt performance.

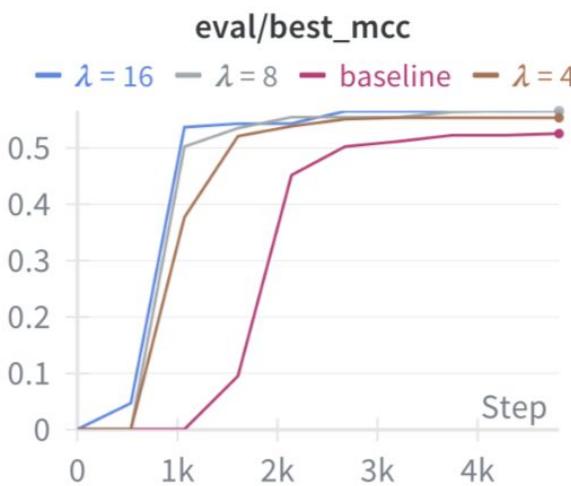
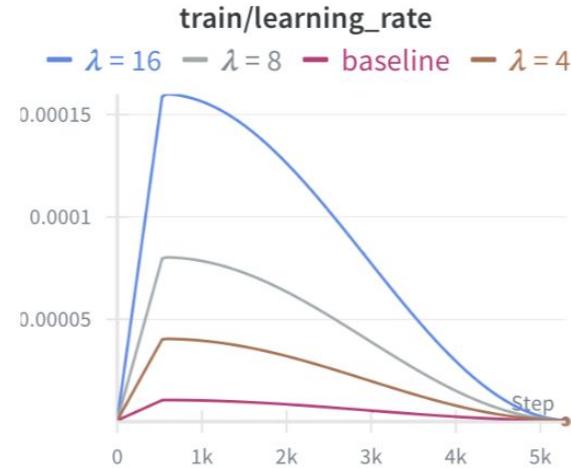
# Phase 1 Results: Metric Analysis

Metric	Input	Layers	Heads	HeadDim	QKV
SVD Entropy	0.6817	1.6908	2.0049	1.999	0.8918
SVD Energy (Top 1)	0.8284	0.4656	0.213	0.2321	0.6188

## METRIC INTERPRETATION

- *Green: High Entropy & Low Energy → Information Density → Increase Rank*
- *Yellow: Low Entropy & High Energy → Spectral Redundancy → Decrease Rank*

# Phase 2 Results: Heterogeneous Learning Rate

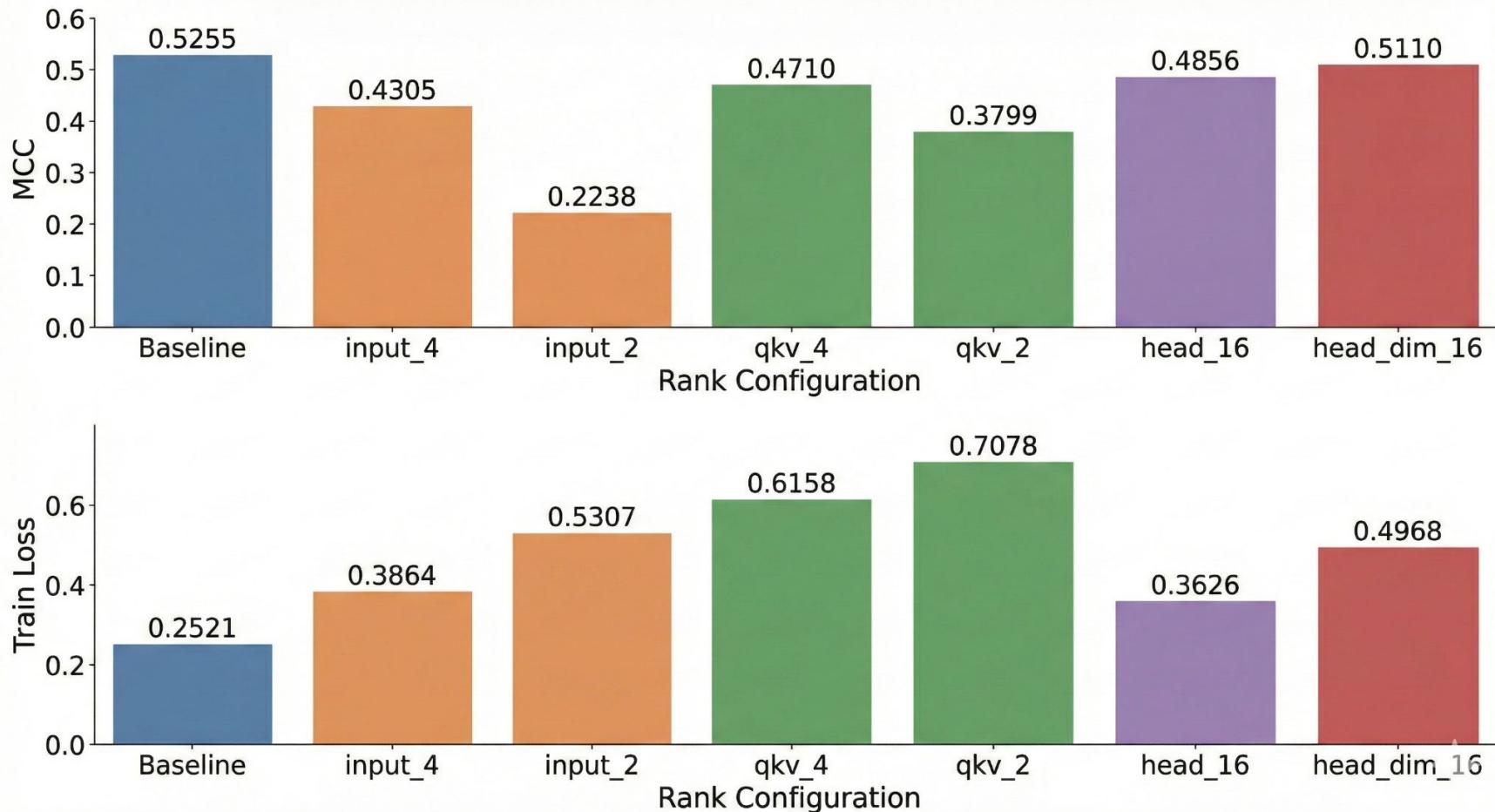


Experiment	Accuracy	MCC
Baseline	0.8063	0.5255
$\lambda=4$	0.8092	0.5536
$\lambda=8$	0.8159	0.5656
$\lambda=16$	0.8187	0.5658

## LR ANALYSIS

- Larger  $\lambda$  gives faster convergence and higher MCC.
- These results confirm the TensLora+'s heterogeneous LR hypothesis.

# Phase 3: Heterogeneous Rank



- *Input, qkv: collapse when rank is sharply reduced, indicating that extreme compression removes essential information these modes must preserve.*
- *Head, head\_dim: degrade when rank is doubled, meaning extra capacity adds noise or leads to overfitting rather than improving representation quality.*

# Limitation & Future Directions

---

- Theoretical Rigor
  - Mathematical Proof for Heterogeneous LR
  - Justification for Rank Selection Metrics
- Experimental Optimization
  - Extensive Hyperparameter Tuning
  - Complex Rank Combinations

# Reference

---

Marmoret, Axel, et al. "TensLoRA: Tensor Alternatives for Low-Rank Adaptation." *arXiv preprint arXiv:2509.19391* (2025).

Hayou, Soufiane, Nikhil Ghosh, and Bin Yu. "Lora+: Efficient low rank adaptation of large models." *arXiv preprint arXiv:2402.12354* (2024).



Thank you