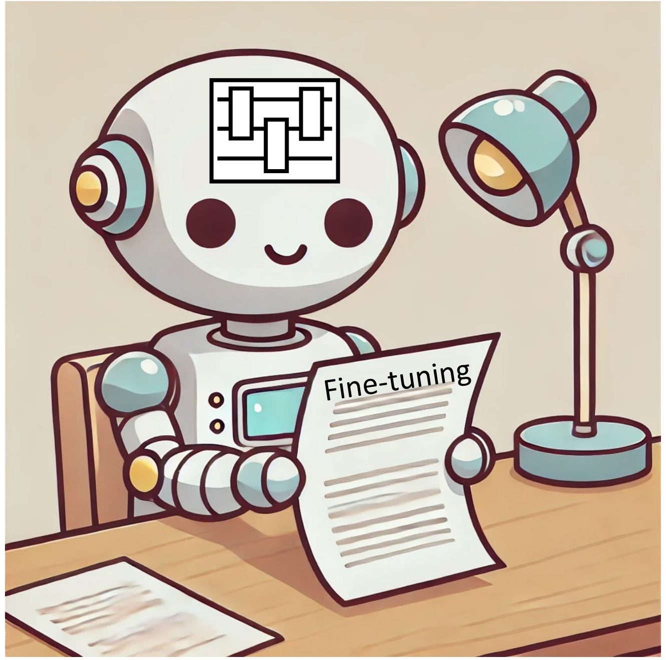
QuanTA: Efficient High-Rank Fine-Tuning of LLMs with Quantum-Informed Tensor Adaptation

Zhuo Chen[12], Rumen Dangovski[13], Charlotte Loh[13],
Owen Dugan[12], Di Luo[124*], Marin Soljacic[12]

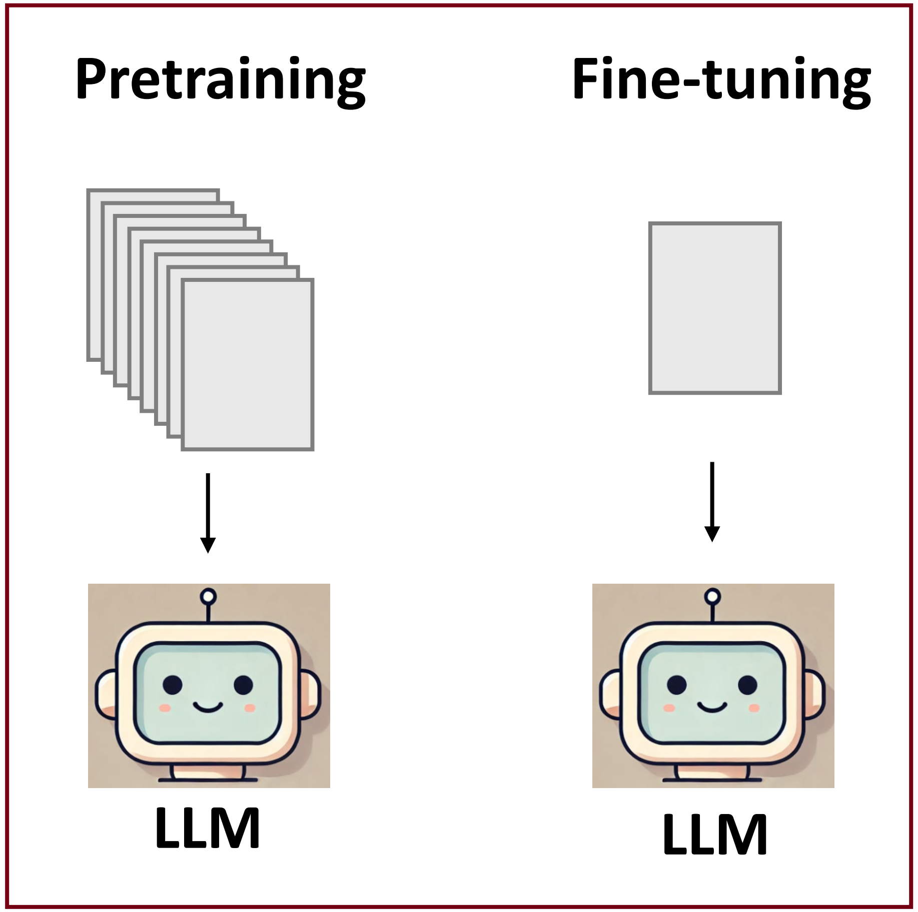
[1]The NSF AI Institute for Artificial Intelligence and Fundamental Interactions
[2]Department of Physics, Massachusetts Institute of Technology
[3]Department of EECS, Massachusetts Institute of Technology
[4]Department of Physics, Harvard University



{chenzhuo,rumenrd,cloh,odugan,diluo,soljacic}@mit.edu

* corresponding author

Workflow with Training LLMs



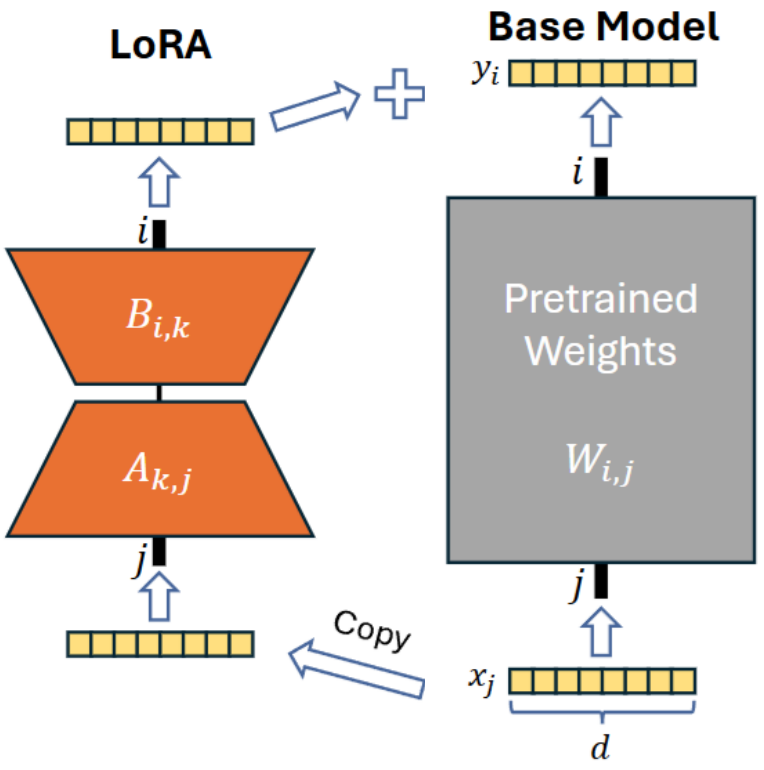
Full fine-tuning:

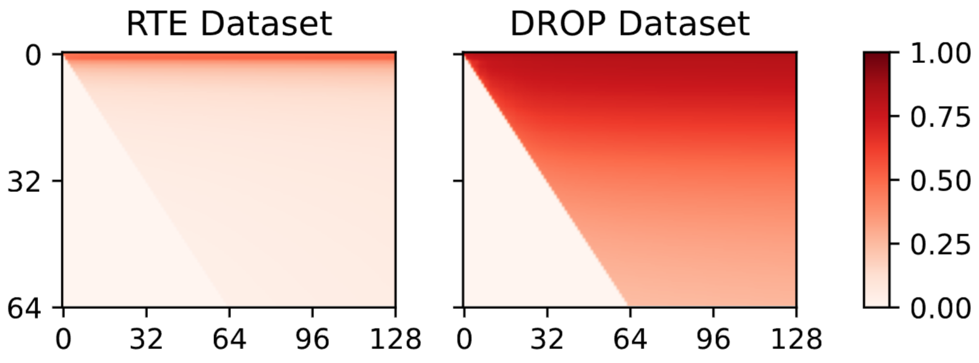
- **Pro**
  - Most flexible
  - High score
- **Con**
  - High cost
  - Prone to overfitting
  - Catastrophic forgetting

LoRA: Low-Rank Adaptation of Large Language Models

Edward Hu* Yelong Shen* Phillip Wallis Zeyuan Allen-Zhu
Yuanzhi Li Shean Wang Lu Wang Weizhu Chen

LoRA Is Not Always Sufficient



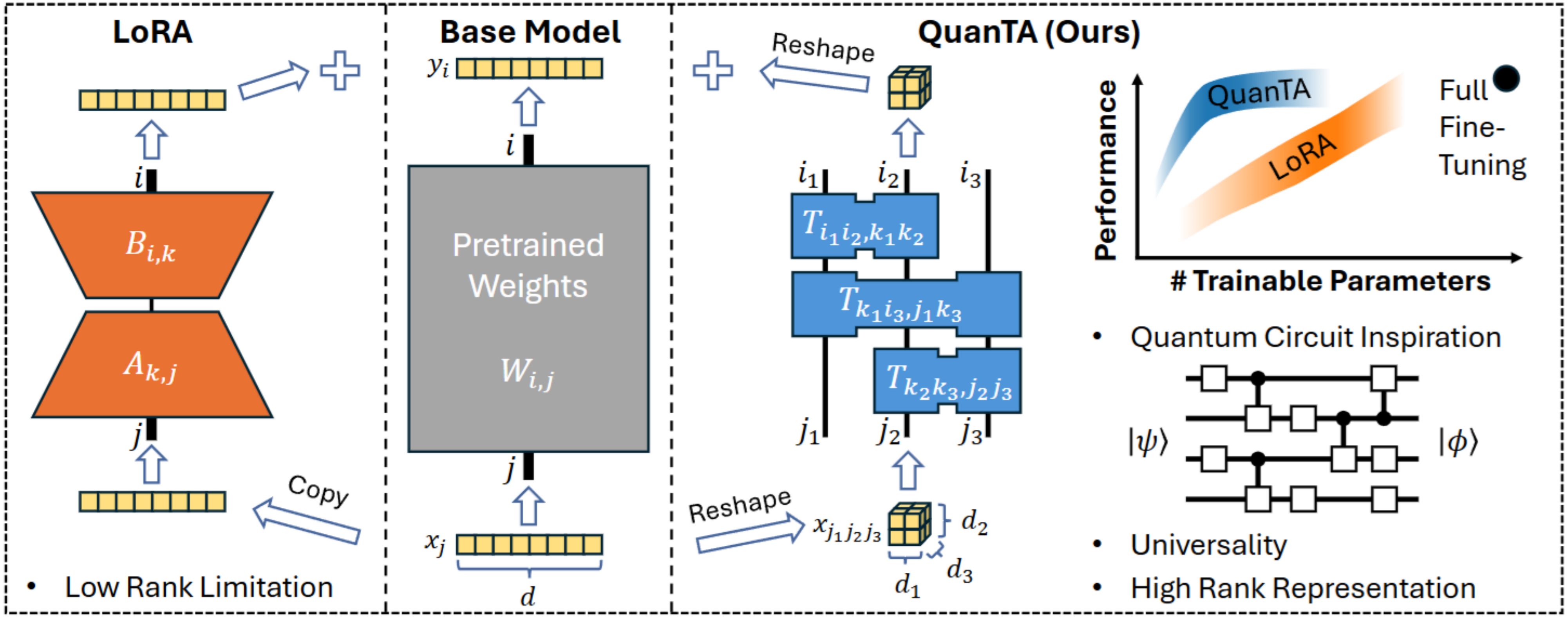


- LoRA works well for tasks with low intrinsic rank
- LoRA may struggle for tasks with high intrinsic rank

How to achieve parameter efficient high-rank fine-tuning?

Quantum-informed Tensor Adaptation

Efficient high-rank finetuning



Easy to Implement

Apply QuanTA weights $T^{(i)}$'s to input x

$$(\mathcal{T}x)_{i_1,i_2,i_3} = \sum_{k_1,k_2} T^{(1)}_{i_1,i_2;k_1,k_2} \sum_{j_1,k_3} T^{(2)}_{k_1,i_3;j_1,k_3} \sum_{j_2,j_3} T^{(3)}_{k_2,k_3;j_2,j_3} x_{j_1,j_2,j_3}$$

```
torch.einsum("...abc,efbc,diaf,ghde->...ghi", x, T_3, T_2, T_1)
```

Calculate Full QuanTA matrix

$$\mathcal{T}_{i;j} = \mathcal{T}_{i_1,i_2,i_3;j_1,j_2,j_3} = \sum_{k_1,k_2} T^{(1)}_{i_1,i_2;k_1,k_2} \sum_{k_3} T^{(2)}_{k_1,i_3;j_1,k_3} T^{(3)}_{k_2,k_3;j_2,j_3}$$

```
torch.einsum("efbc,diaf,ghde->ghiabc", T_3, T_2, T_1)
```

No inference overhead!

Theoretical Guarantees

Theorem 6.1 (Universality of QuanTA). *Let W be an arbitrary matrix of shape $2^M \times 2^M$. For any collection of local dimensions $\{d_n\}$ such that each d_n is a power of 2 and $\prod_n d_n = 2^M$, it is always possible to decompose W into a finite sequence of tensors $\{T^{(\alpha)}\}$, where each tensor applies on two axes with local dimensions $d_{m(\alpha)}$ and $d_{n(\alpha)}$.*

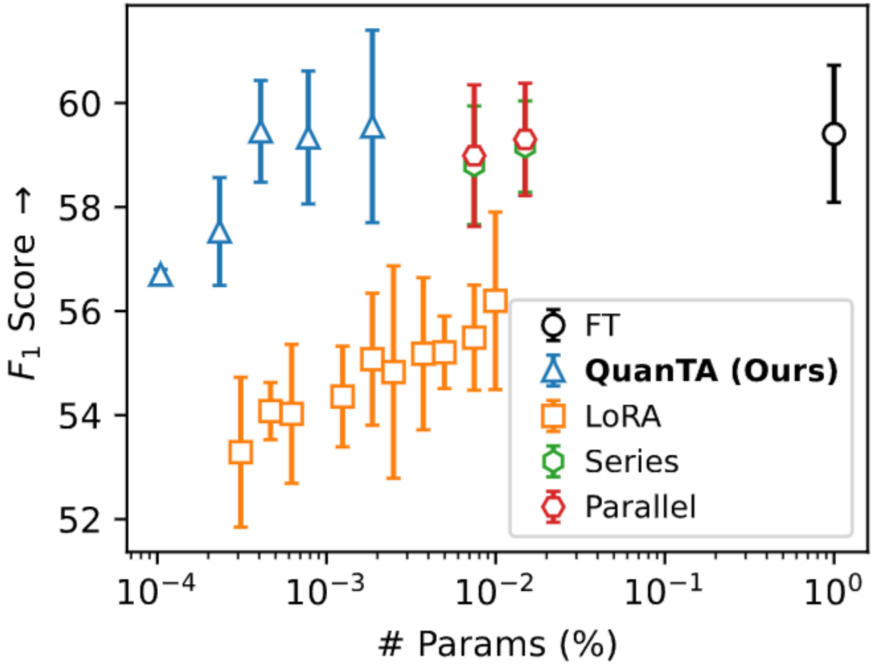
Theorem 6.2 (Rank representation). *Let $R = r(\mathcal{T})$ be the rank of the full QuanTA operator, $R^{(\alpha)} = r(T^{(\alpha)})$ be the rank of individual tensors, d be the total dimension of $\mathcal{T}$, $d^{(\alpha)} = d_{m(\alpha)} d_{n(\alpha)}$ be the total dimension of the individual tensor $T^{(\alpha)}$, and N_T be the total number of tensors. The following inequality always holds*

$$\sum_{\alpha} \frac{dR^{(\alpha)}}{d^{(\alpha)}} - d(N_T - 1) \leq R \leq \min_{\alpha} \frac{dR^{(\alpha)}}{d^{(\alpha)}}.$$

Theorem 6.3 (Composition openness). *There exists a set $\mathbb{S} = \{\mathcal{M}_k\}$ of matrices generated from a fixed QuanTA structure and two matrices $\mathcal{M}_1, \mathcal{M}_2 \in \mathbb{S}$ such that $\mathcal{M}_1 \mathcal{M}_2 \notin \mathbb{S}$.*

Benchmark on DROP Dataset

Model	PEFT Method	# Params (%)	F_1 Score (↑)
LLaMA2$_{7B}$	FT	100%	59.4
	Series	0.747%	58.8
	Parallel	0.747%	59.0
	LoRA$_{r=8}$	0.062%	54.0
	LoRA$_{r=32}$	0.249%	54.8
	LoRA$_{r=128}$	0.996%	56.2
	QuanTA$_{16\text{-}8\text{-}8\text{-}4}$ (Ours)	0.041%	59.5
	QuanTA$_{16\text{-}16\text{-}16}$ (Ours)	0.261%	**59.6**
LLaMA2$_{13B}$	LoRA$_{r=8}$	0.050%	61.0
	QuanTA$_{16\text{-}8\text{-}8\text{-}5}$ (Ours)	0.029%	**69.0**
LLaMA2$_{70B}$	LoRA$_{r=8}$	0.024%	74.3
	QuanTA$_{16\text{-}8\text{-}8\text{-}8}$ (Ours)	0.014%	**79.4**



Benchmark on Commonsense Reasoning

Model	PEFT Method	# Params (%)	Accuracy (↑)								
			BoolQ	PIQA	SIQA	HellaS.	WinoG.	ARC-e	ARC-c	OBQA	Avg.
GPT-3_{175B}*	–	–	60.5	81.0	–	78.9	70.2	68.8	51.4	57.6	–
PaLM_{540B}*	–	–	88.0	82.3	–	83.4	81.1	76.6	53.0	53.4	–
ChatGPT*	–	–	73.1	85.4	68.5	78.5	66.1	89.8	79.9	74.8	77.0
LLaMA_{7B}	FT	100%	71.3	82.1	78.6	90.2	79.0	82.9	67.2	76.8	78.5
	Prefix*	0.11%	64.3	76.8	73.9	42.1	72.1	72.9	54.0	60.6	64.6
	Series*	0.99%	63.0	79.2	76.3	67.9	75.7	74.5	57.1	72.4	70.8
	Parallel*	3.54%	67.9	76.4	78.8	69.8	78.9	73.7	57.3	75.2	72.3
	LoRA*	0.83%	68.9	80.7	77.4	78.1	78.8	77.8	61.3	74.8	74.7
	DoRA†	0.43%	70.0	82.6	**79.7**	83.2	80.6	80.6	65.4	77.6	77.5
	DoRA†	0.84%	69.7	**83.4**	78.6	87.2	81.0	81.9	66.2	79.2	78.4
	QuanTA (Ours)	0.041%	**71.6**	83.0	**79.7**	**91.8**	**81.8**	**84.0**	**68.3**	**82.1**	**80.3**
LLaMA_{13B}	Prefix*	0.03%	65.3	75.4	72.1	55.2	68.6	79.5	62.9	68.0	68.4
	Series*	0.80%	71.8	83.0	79.2	88.1	82.4	82.5	67.3	81.8	79.5
	Parallel*	2.89%	72.5	84.8	79.8	92.1	84.7	84.2	71.2	82.4	81.5
	LoRA*	0.67%	72.1	83.5	80.5	90.5	83.7	82.8	68.3	82.4	80.5
	DoRA†	0.35%	72.5	85.3	79.9	90.1	82.9	82.7	69.7	83.6	80.8
	DoRA†	0.68%	72.4	84.9	81.5	92.4	84.2	84.2	69.6	82.8	81.5
	QuanTA (Ours)	0.029%	**73.2**	**85.4**	**82.1**	**93.4**	**85.1**	**87.8**	**73.3**	**84.4**	**83.1**
LLaMA2_{7B}	FT	100%	**72.9**	83.0	79.8	92.4	83.0	**86.6**	72.0	80.1	81.2
	LoRA†	0.83%	69.8	79.9	79.5	83.6	82.6	79.8	64.7	81.0	77.6
	DoRA†	0.43%	72.0	83.1	**79.9**	89.1	83.0	84.5	71.0	81.2	80.5
	DoRA†	0.84%	71.8	83.7	76.0	89.1	82.6	83.7	68.2	82.4	79.7
	QuanTA (Ours)	0.041%	72.4	**83.8**	79.7	**92.5**	**83.9**	85.3	**72.5**	**82.6**	**81.6**

Benchmark on Arithmetic Reasoning

Model	PEFT Method	# Params (%)	Accuracy (↑)				
			AQuA	**GSM8K**	**MAWPS**	**SVAMP**	**Avg. W/O AQuA**
GPT-3.5$_{175B}$*	–	–	38.9	56.4	87.4	69.6	71.1
LLaMA2$_{7B}$	FT	100%	*19.3*	65.2	92.0	80.7	79.3
	LoRA	0.83%	*17.5*	65.7	91.2	**80.8**	79.6
	QuanTA (Ours)	0.19%	*16.7*	**67.0**	**94.3**	80.3	**80.5**
LLaMA2$_{13B}$	LoRA	0.67%	*16.7*	72.3	90.8	84.3	82.5
	QuanTA (Ours)	0.13%	*18.9*	**72.4**	**94.5**	**84.8**	**83.9**

Conclusion and Outlooks

Conclusion:

- QuanTA is an efficient, easy-to-implement, high-rank fine-tuning method with no inference overhead
- QuanTA leverages quantum-inspired techniques to achieve high-rank adaptations
- QuanTA is guaranteed by universality theorem and rank representation theorem
- QuanTA demonstrates better performance with extremely few parameters on various tasks

Outlook:

- Apply QuanTA in other domains such as image or video generation
- Integrate QuanTA with other fine-tuning methods such as quantization
- Explore additional optimization techniques tailored specifically for QuanTA
- Design new fine-tuning methods based on principles from quantum computing