ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
UNIVERSITY OF MICHIGAN

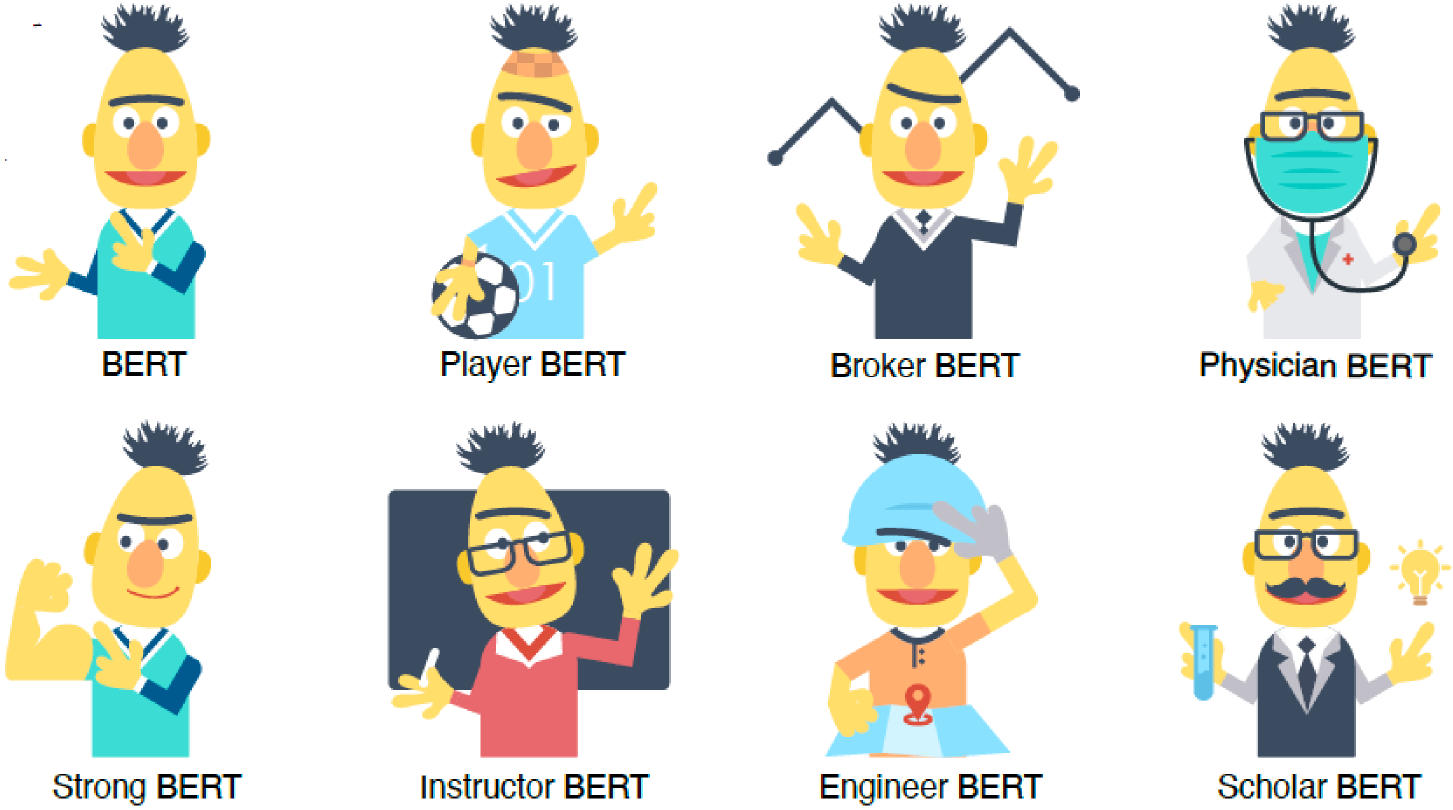
ICLR 22'

LoRA: Low-Rank Adaptation of Large Language Models

Presented by

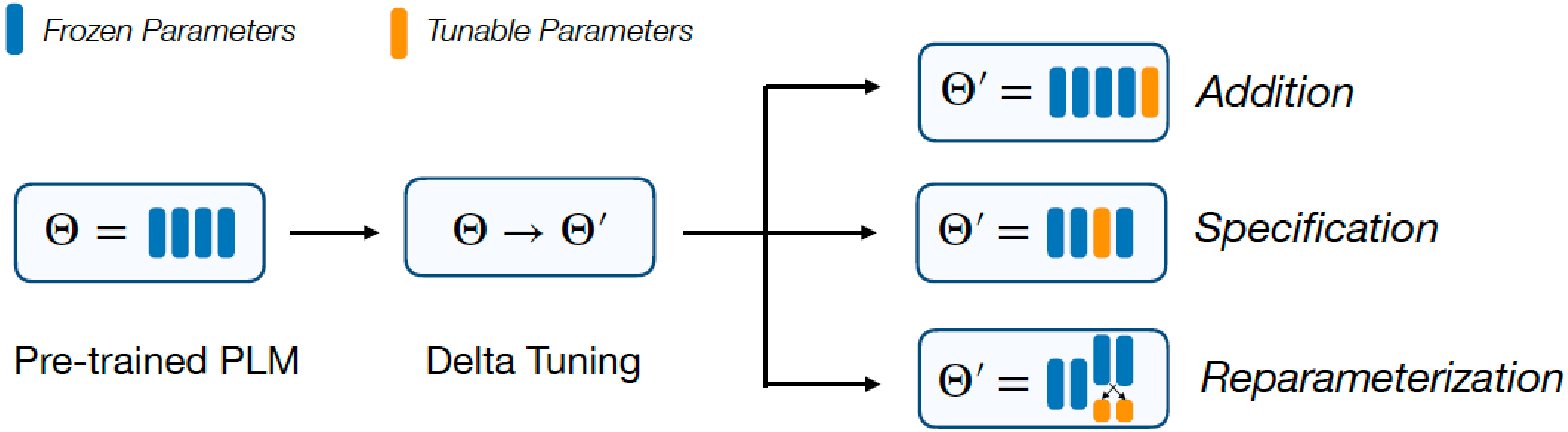
Tianyu Zhang | zhtianyu@umich.edu

Parameter Efficient Fine Tuning For LLMs



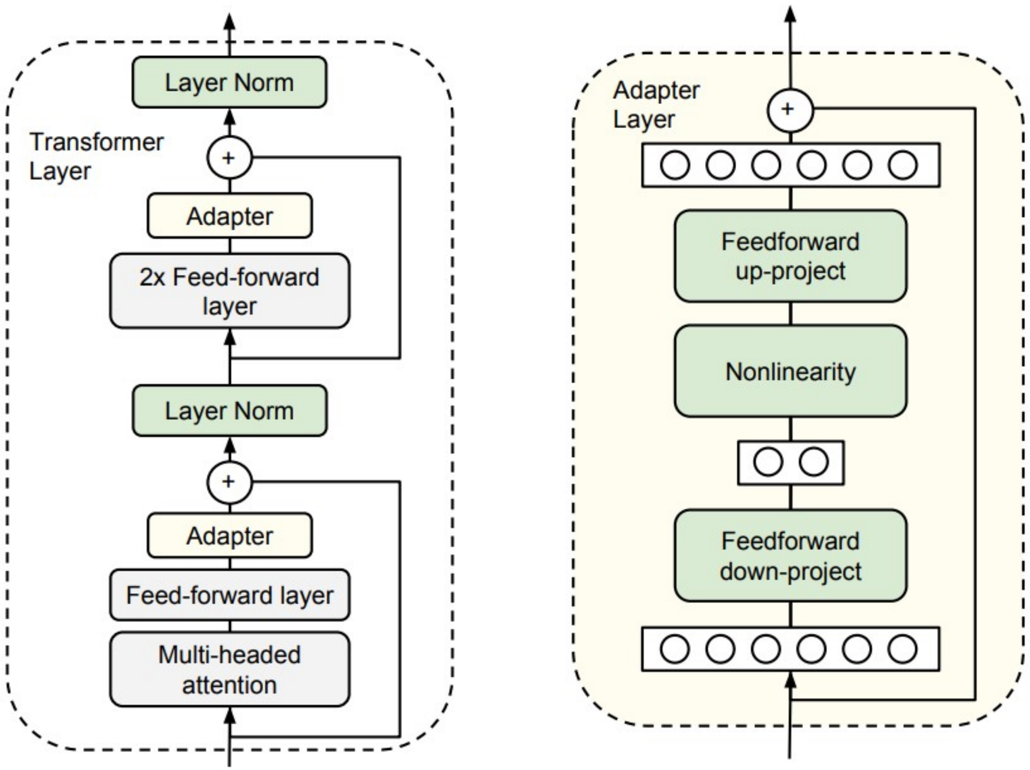
PEFT seeks to adapt and specialize PLMs with changes of a small portion of parameters.

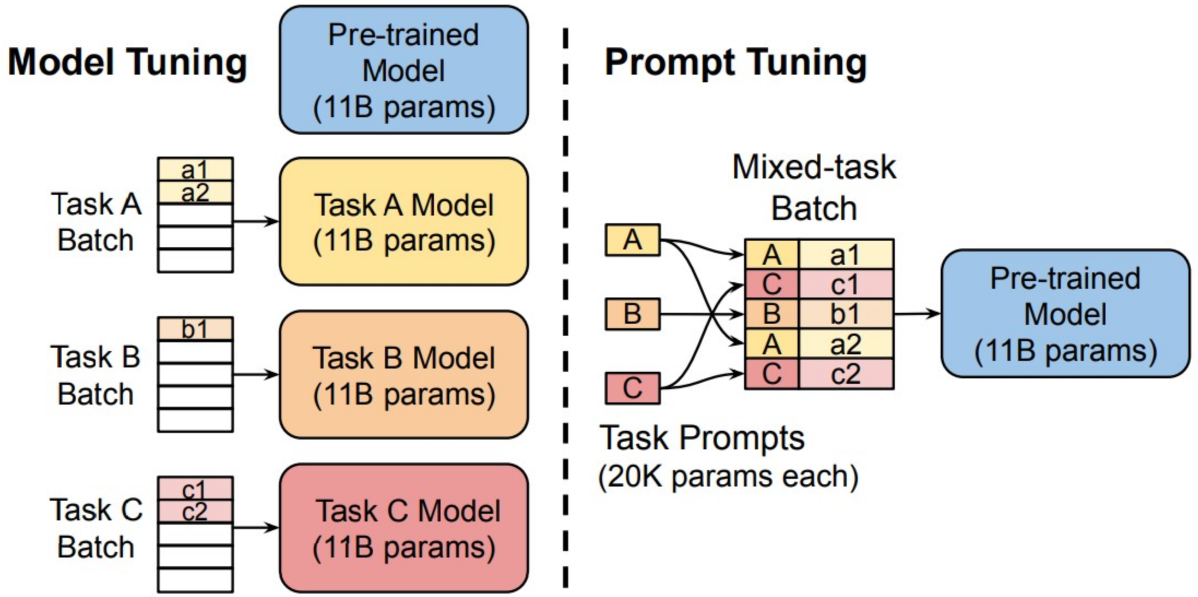
Categorization Of Peft



denote the pre-trained parameters, and represent the well-tuned parameters.

ADDITION-BASED METHOD





(1) Adapter-based tuning

(2) Prompt-based tuning

Parameter-efficient transfer learning for NLP. **ICML 2019**
The Power of Scale for Parameter-Efficient Prompt Tuning. **EMNLP 2021**

Specification-Based Methods

$$\mathbf{Q}^{m,\ell}(\mathbf{x}) = \mathbf{W}_q^{m,\ell}\mathbf{x} + \mathbf{b}_q^{m,\ell}$$
$$\mathbf{K}^{m,\ell}(\mathbf{x}) = \mathbf{W}_k^{m,\ell}\mathbf{x} + \mathbf{b}_k^{m,\ell}$$
$$\mathbf{V}^{m,\ell}(\mathbf{x}) = \mathbf{W}_v^{m,\ell}\mathbf{x} + \mathbf{b}_v^{m,\ell}$$

$$\mathbf{h}_2^\ell = \text{Dropout}\big(\mathbf{W}_{m_1}^\ell \cdot \mathbf{h}_1^\ell + \mathbf{b}_{m_1}^\ell\big)$$
$$\mathbf{h}_3^\ell = \mathbf{g}_{LN_1}^\ell \odot \frac{(\mathbf{h}_2^\ell + \mathbf{x}) - \mu}{\sigma} + \mathbf{b}_{LN_1}^\ell$$
$$\mathbf{h}_4^\ell = \text{GELU}\big(\mathbf{W}_{m_2}^\ell \cdot \mathbf{h}_3^\ell + \mathbf{b}_{m_2}^\ell\big)$$
$$\mathbf{h}_5^\ell = \text{Dropout}\big(\mathbf{W}_{m_3}^\ell \cdot \mathbf{h}_4^\ell + \mathbf{b}_{m_3}^\ell\big)$$
$$\text{out}^\ell = \mathbf{g}_{LN_2}^\ell \odot \frac{(\mathbf{h}_5^\ell + \mathbf{h}_3^\ell) - \mu}{\sigma} + \mathbf{b}_{LN_2}^\ell$$

Heuristic Specification

$$\boldsymbol{\theta}_\tau = \boldsymbol{\theta} + \boldsymbol{\delta}_\tau,$$

$$\min_{\boldsymbol{\delta}_\tau} \; L(\mathcal{D}_\tau, f_\tau, \boldsymbol{\theta} + \boldsymbol{\delta}_\tau) + \lambda R(\boldsymbol{\theta} + \boldsymbol{\delta}_\tau),$$

$$L(\mathcal{D}_\tau, f_\tau, \boldsymbol{\theta}_\tau) = \frac{1}{N}\sum_{n=1}^{N} C\left(f_\tau(x_\tau^{(n)}; \boldsymbol{\theta}_\tau), y_\tau^{(n)}\right)$$

$$R(\boldsymbol{\theta} + \boldsymbol{\delta}_\tau) = \|\boldsymbol{\delta}_\tau\|_0 = \sum_{i=1}^{d} \mathbb{1}\{\boldsymbol{\delta}_{\tau,i} \neq 0\}$$

Learn the Specification

Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. **ACL 2022**
Parameter-efficient transfer learning with diff pruning. **ACL 2021**

Reparameterization-Based Method

What is Intrinsic Dimensionality?

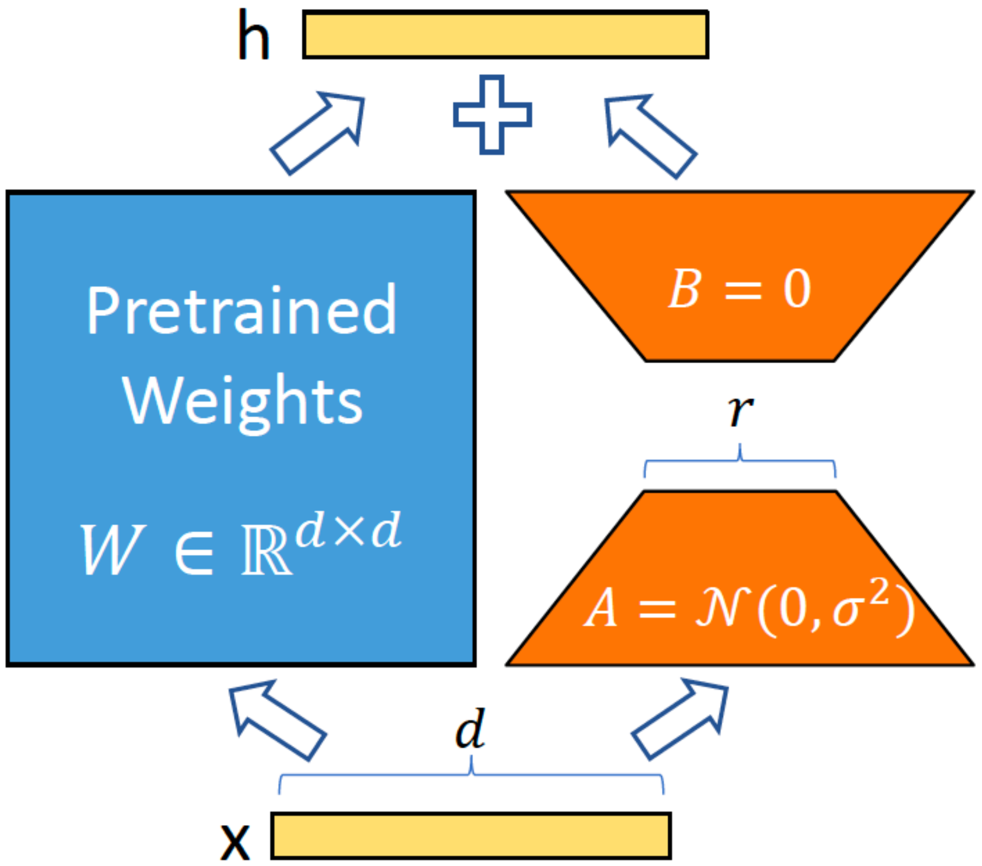
The intrinsic dimension of an objective function measures the minimum number of parameters needed to reach satisfactory solutions to the respective objective

- Pre-training implicitly reduces the intrinsic dimension.
- Larger models, after a certain number of training iterations, tend to exhibit lower intrinsic dimensions.
- Simpler downstream tasks correspond to lower intrinsic dimensions.
- Lower intrinsic dimensionality is associated with better generalization performance.

Intrinsic dimensionality explains the effectiveness of language model fine-tuning. **ACL 2021**

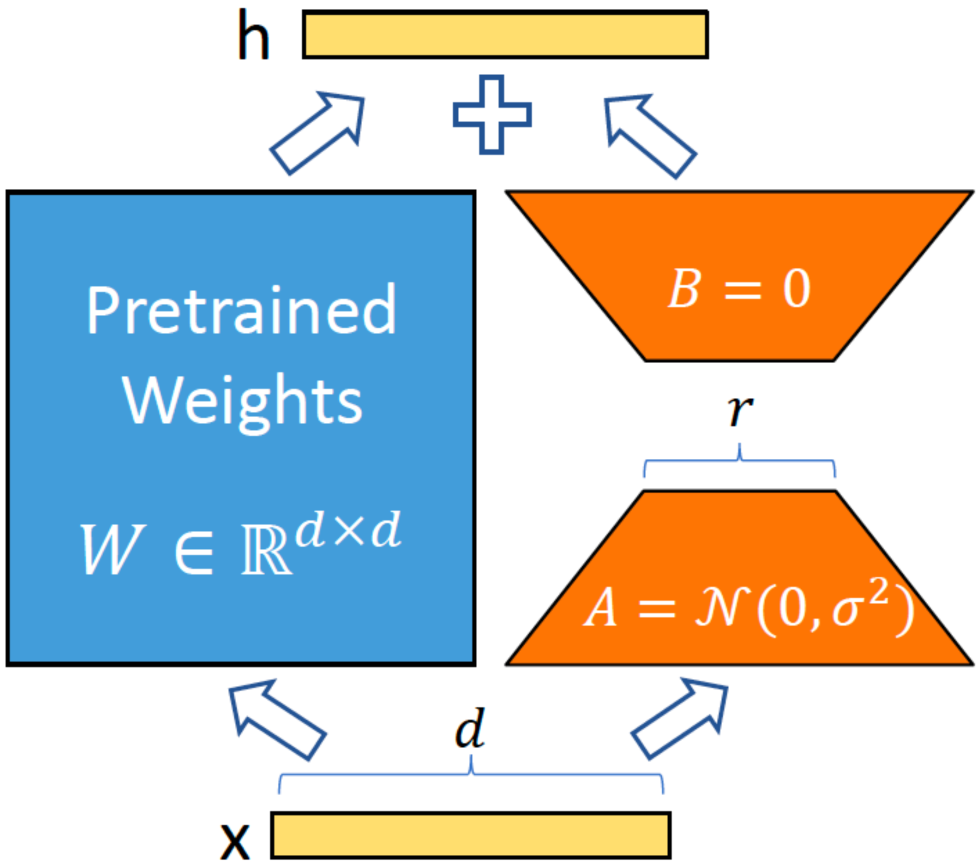
Motivation1: Low Rank

The learned over-parametrized models in fact reside on a low intrinsic dimension. We hypothesize that the change in weights during model adaptation also has a low "intrinsic rank".



Motivation2: Comparision

- *There is no direct ways to bypass the extra compute in **adapter** layers.*
- ***Prefix tuning** is difficult to optimize and that its performance changes non-monotonically in trainable parameters.*
- *Reserving a part of the sequence length for adaptation reduces the sequence length available to process a downstream task.*

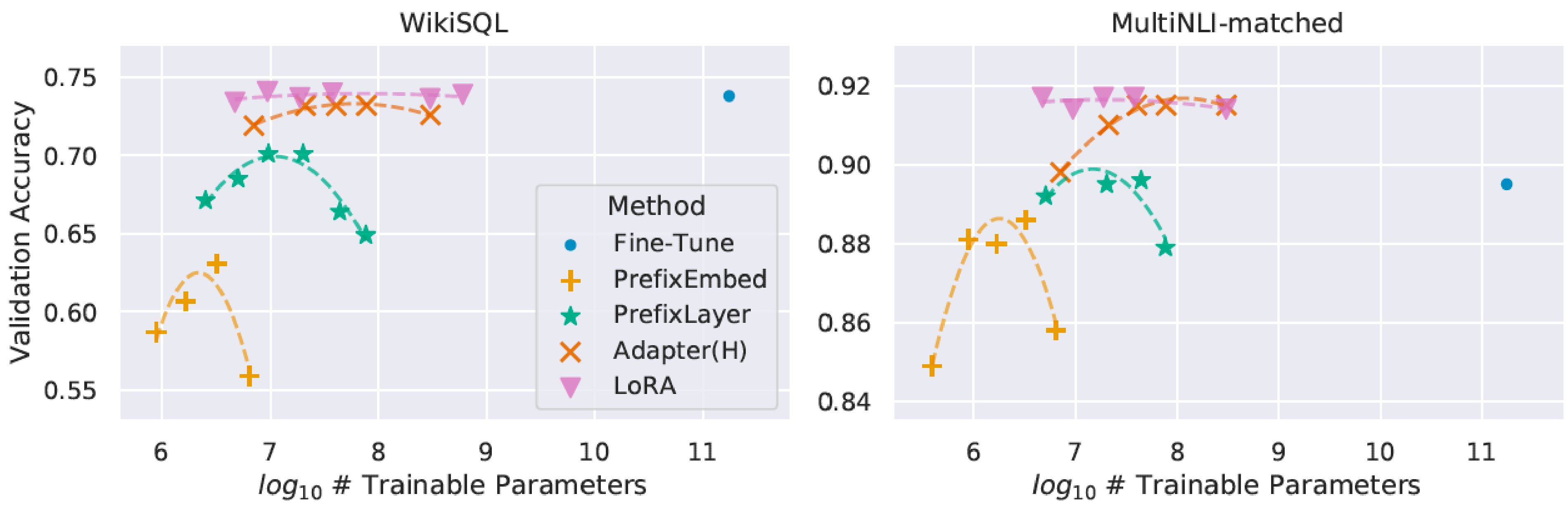


METHODOLOGY

Only adapting the ***attention weights*** *for downstream tasks and freeze the MLP modules both for simplicity and parameter-efficiency.*

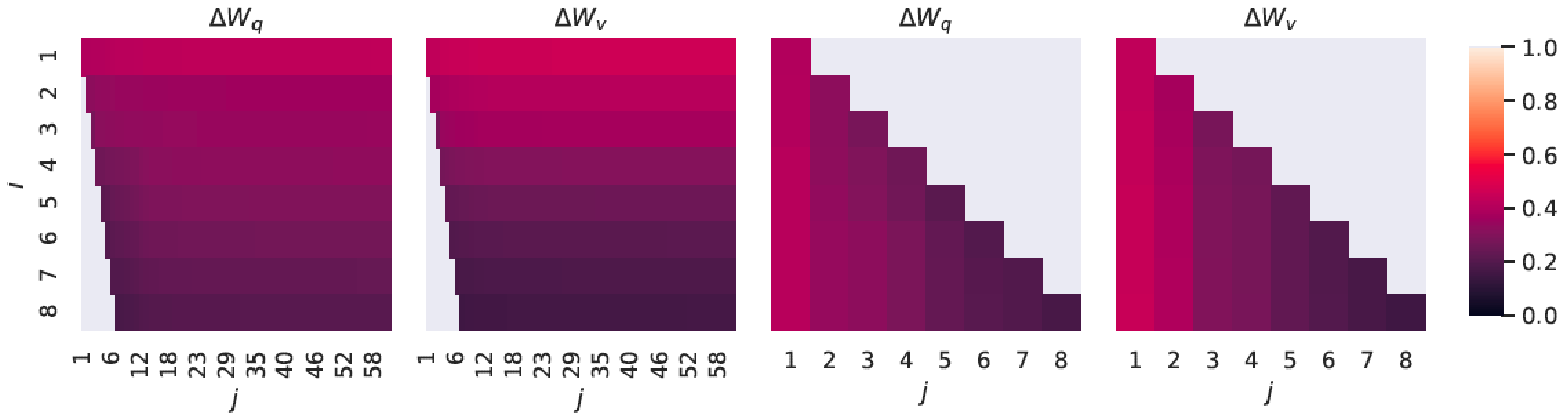
	# of Trainable Parameters = 18M						
Weight Type	W_q	W_k	W_v	W_o	W_q, W_k	W_q, W_v	W_q, W_k, W_v, W_o
Rank r	8	8	8	8	4	4	2
WikiSQL ($\pm 0.5\%$)	70.4	70.0	73.0	73.2	71.4	**73.7**	**73.7**
MultiNLI ($\pm 0.1\%$)	91.0	90.8	91.0	91.3	91.3	91.3	**91.7**

Experiments



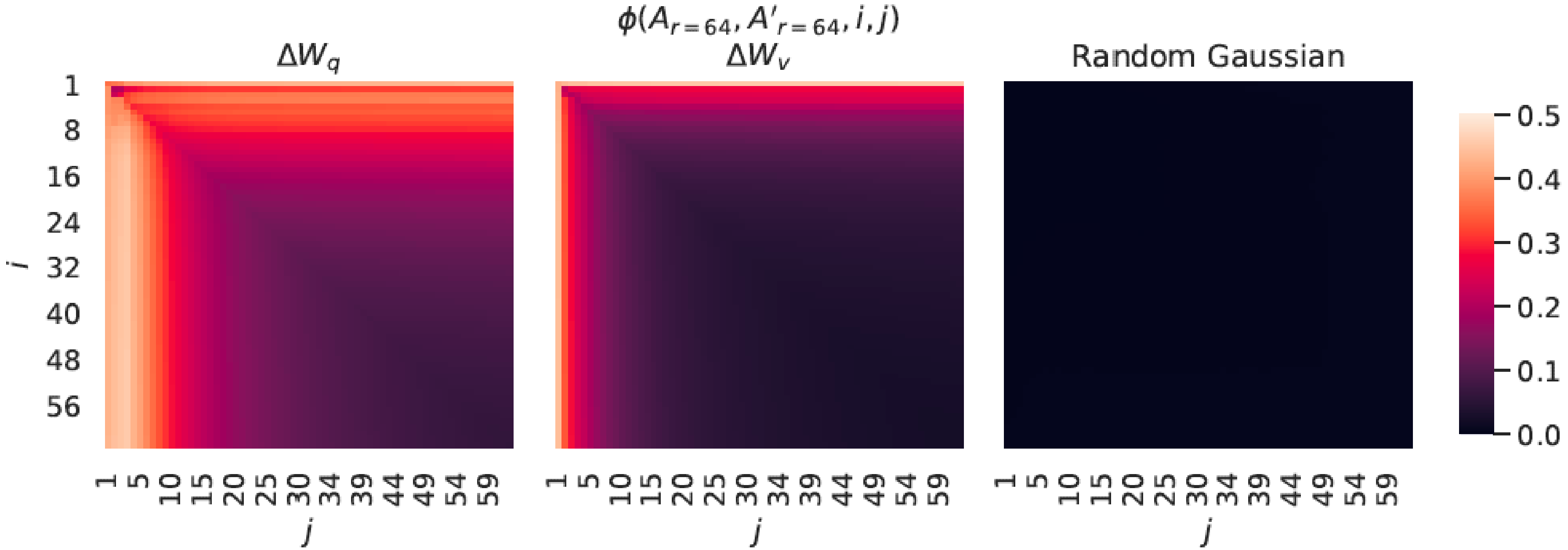
What Is The Optimal Rank

	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL($\pm 0.5\%$)	W_q	68.8	69.6	70.5	70.4	70.0
	W_q, W_v	73.4	73.3	73.7	73.8	73.5
	W_q, W_k, W_v, W_o	74.1	73.7	74.0	74.0	73.9
MultiNLI ($\pm 0.1\%$)	W_q	90.7	90.9	91.1	90.7	90.7
	W_q, W_v	91.3	91.4	91.3	91.6	91.4
	W_q, W_k, W_v, W_o	91.2	91.7	91.7	91.5	91.4



How Does The Adaption Compared To W

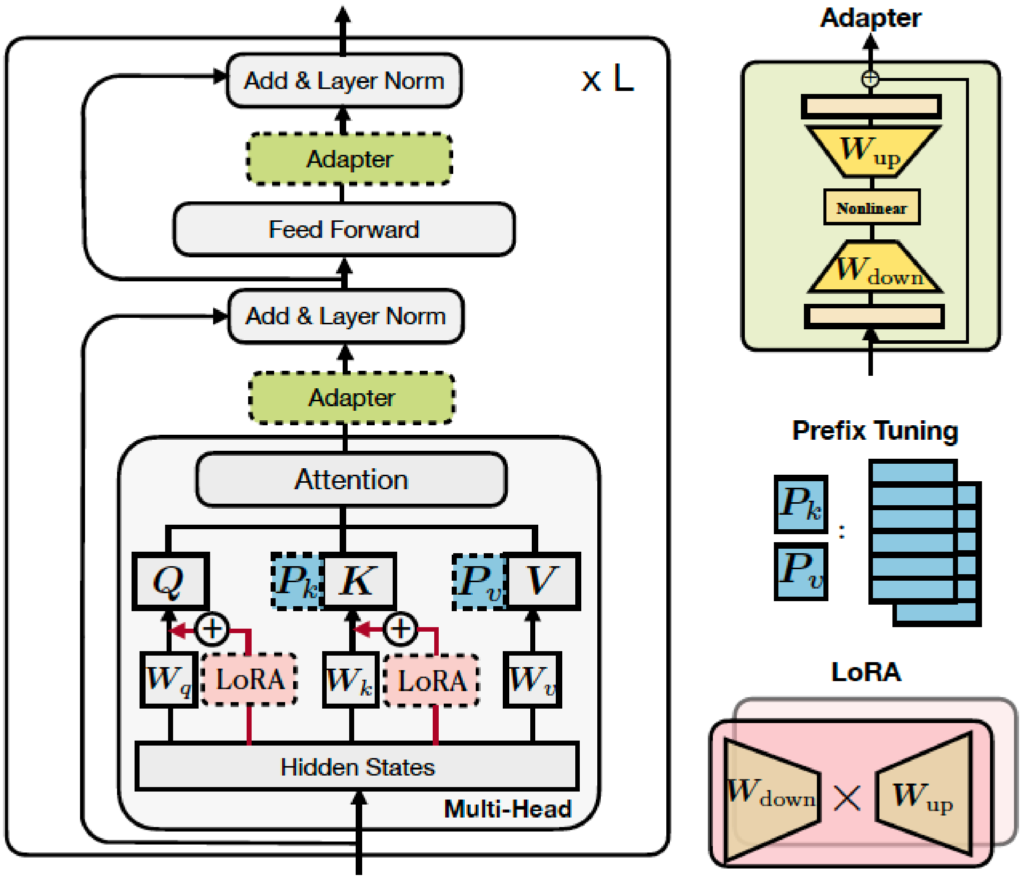
	$r=4$			$r=64$		
	ΔW_q	W_q	Random	ΔW_q	W_q	Random
$\|\|U^\top W_q V^\top\|\|_F =$	0.32	21.67	0.02	1.90	37.71	0.33
$\|\|W_q\|\|_F = 61.95$	$\|\|\Delta W_q\|\|_F = 6.91$			$\|\|\Delta W_q\|\|_F = 3.57$		

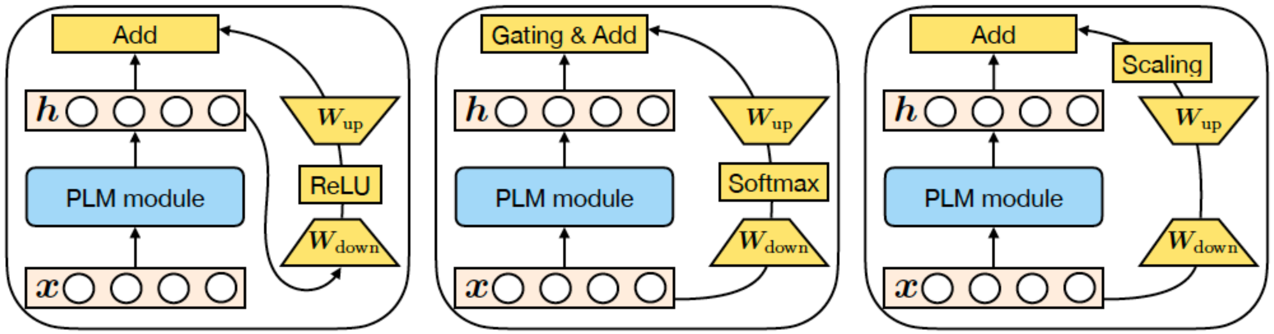


FUTURE WORK

1. *LoRA can be combined with other efficient adaptation methods, potentially providing orthogonal improvement.*
2. *The mechanism behind fine-tuning or LoRA is far from clear – how are features learned during pre-training transformed to do well on downstream tasks? We believe that LoRA makes it more tractable to answer this than full fine-tuning.*
3. *We mostly depend on heuristics to select the weight matrices to apply LoRA to. Are there more principled ways to do it?*
4. *Finally, the rank-deficiency of the adaption matrix suggests that W could be rank-deficient as well, which can also be a source of inspiration for future works.*

A Unified View





(a) Adapter (b) Prefix Tuning (c) LoRA

Towards a unified view of parameter-efficient transfer learning. **ICLR 2022**

Potential Impact Of Lora

README.md

QLoRA: Efficient Finetuning of Quantized LLMs

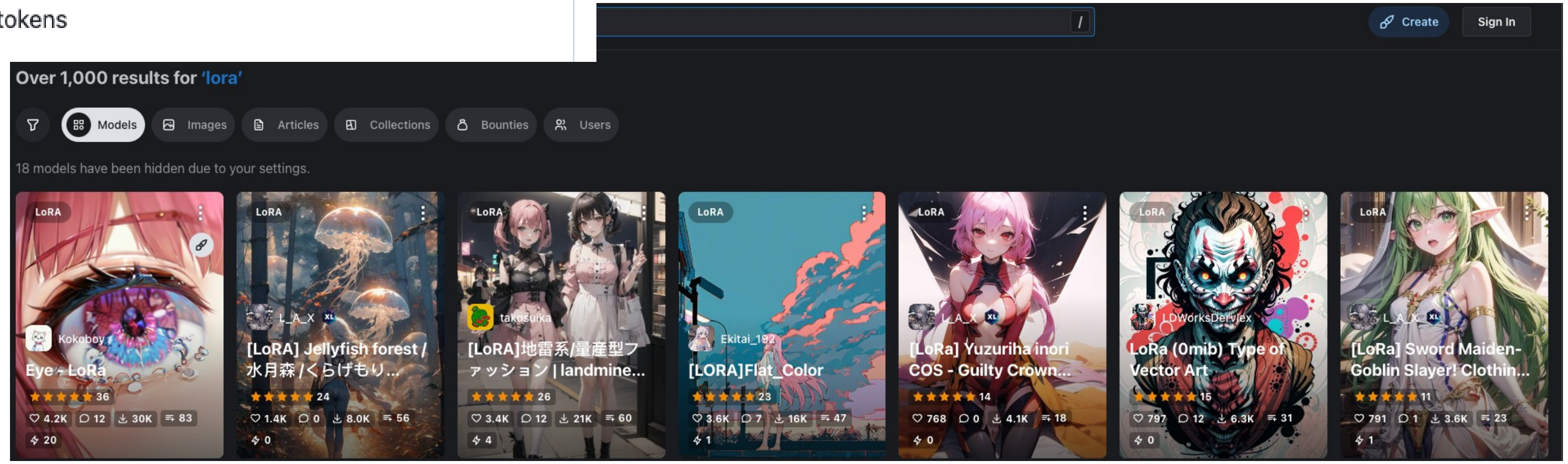
| Paper | Adapter Weights | Demo |

This repo supports the paper "QLoRA: Efficient Finetuning of Quantized LLMs", an effort to democratize access to LLM research.

QLoRA uses bitsandbytes for quantization and is integrated with Hugging Face's PEFT and transformers libraries. QLoRA was developed by members of the University of Washington's UW NLP group.

Updates

- 7/19/2023 - Added LLaMA 2 example script and updated version requirements
- 7/18/2023 - Fixed non-frozen embeddings when adding new tokens



QUESTIONS

1. What are the different types of PEFT?
2. What are the advantages of LORA compared to adapter and prompt tuning?
3. (OPEN) Why is LORA considered better for training than prompt tuning?

THANKS!