GIVA: Gradient-Informed Bases for Vector-Based Adaptation

Neeraj Gangwar[1], Rishabh Deshmukh[2], Michael Shavlovsky[2], Hancao Li[2], Vivek Mittal[2], Lexing Ying[3], Nickvash Kani[1]

May 2, 2026

[1]University of Illinois Urbana-Champaign [2]Amazon [3]Stanford University

Vector-Based Adaptation

For a pre-trained weight matrix $W_{\text{pt}} \in \mathbb{R}^{m \times d}$, the weight update ΔW is defined as

$$\Delta W = \Gamma B \Lambda A$$

$$W' = W_{\text{pt}} + \Delta W = W_{\text{pt}} + \Gamma B \Lambda A$$

Here,

- $B \in \mathbb{R}^{m \times r}$
- $A \in \mathbb{R}^{r \times d}$
- $\Gamma = \text{diag}(\gamma_1 \dots \gamma_m)$
- $\Lambda = \text{diag}(\lambda_1 \dots \lambda_r)$
- $r < \min(m, d)$

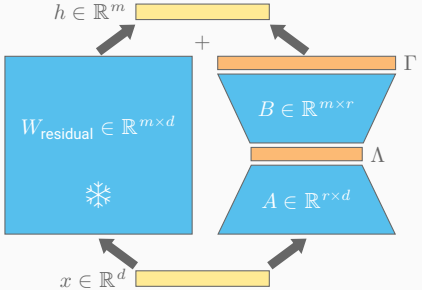


Figure 1: trainable parameters; frozen parameters. W_{residual} is the difference between W_{pt} and the initial value of ΔW.

Initializing Vector-Based Adaptation Methods

- VeRA (Kopiczko et al., 2024) initializes A and B as random matrices shared across layers.
- OSoRA (Han et al., 2025) derives A and B from the singular value decomposition of W_{pt}.

Kopiczko et al., "VeRA: Vector-based Random Matrix Adaptation", 2024.

Han et al., "OSoRA: Output-Dimension and Singular-Value Initialized Low-Rank Adaptation", 2025.

Initializing Vector-Based Adaptation Methods

- VeRA (Kopiczko et al., 2024) initializes A and B as random matrices shared across layers.
- OSoRA (Han et al., 2025) derives A and B from the singular value decomposition of W_{pt}.

As A and B are not updated during fine-tuning and contain no information about the downstream task, VeRA and OSoRA require higher ranks than LoRA to achieve comparable performance. Our work focuses on the following problem:

Can we reduce the required rank in vector-based adaptation methods to achieve training times comparable to LoRA while preserving their performance?

Kopiczko et al., "VeRA: Vector-based Random Matrix Adaptation", 2024.

Han et al., "OSoRA: Output-Dimension and Singular-Value Initialized Low-Rank Adaptation", 2025.

Our Approach

We employ a gradient-based initialization for A and B. In our method, A and B are chosen such that the first-step update closely approximates the first-step full fine-tuning update.

Our Approach

We employ a gradient-based initialization for A and B. In our method, A and B are chosen such that the first-step update closely approximates the first-step full fine-tuning update.

Setting $\Gamma_0 = \mathbf{0}^{m\times m}$ and $\Lambda_0 = \mathbb{I}^{r\times r}$ reduces the initialization to the following optimization problem (η and $\mathcal{L}$ are the learning rate and the loss function):

$$\arg\min_{A,B} \| \eta \nabla_W \mathcal{L}(W_{\text{pt}}) A^T B^T B A - \eta \nabla_W \mathcal{L}(W_{\text{pt}}) \|_F \quad (1)$$

Approx. first-step GIVA update

First-step full fine-tuning update

Our Approach

We employ a gradient-based initialization for A and B. In our method, A and B are chosen such that the first-step update closely approximates the first-step full fine-tuning update.

Setting $\Gamma_0 = \mathbf{0}^{m\times m}$ and $\Lambda_0 = \mathbb{I}^{r\times r}$ reduces the initialization to the following optimization problem (η and $\mathcal{L}$ are the learning rate and the loss function):

$$\arg\min_{A,B} \| \eta\nabla_W \mathcal{L}(W_{\text{pt}})A^T B^T BA - \eta\nabla_W \mathcal{L}(W_{\text{pt}}) \|_F \quad (1)$$

Approx. first-step GIVA update

First-step full fine-tuning update

Theorem 1

The optimization problem in Equation 1 reaches its minimum when

$$A = V_r^T, B^T B = \mathbb{I}^{r\times r}$$

V_r denotes the right singular vectors of $\nabla_W \mathcal{L}(W_{\text{pt}})$, corresponding to the r largest singular values.

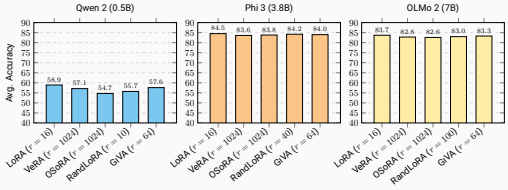
Experiments

GIVA requires lower ranks than VeRA and OSoRA to achieve comparable performance.

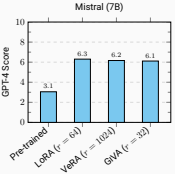
Figure 2: Performance of GIVA on various benchmarks.

GIVA requires lower ranks than VeRA and OSoRA to achieve comparable performance.

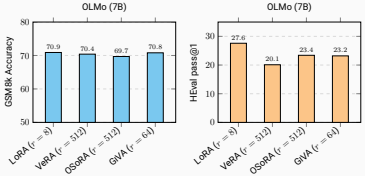
(a) GLUE



(b) Commonsense reasoning



(c) MT-Bench



(d) Math Reasoning and Code Generation

(e) Image Classification

Figure 2: Performance of GIVA on various benchmarks.

Experiments

GIVA requires lower ranks than VeRA and OSoRA to achieve comparable performance.

Figure 2: Performance of GIVA on various benchmarks.

Experiments

GIVA trains faster than the existing vector-based adaptation methods to achieve comparable results.

Method	DINOv2 (87M)	CLIP (303M)	Qwen 2 (494M)	OLMo 2 (7B)
LoRA	1h21m	4h59m	11m	2h54m
VeRA	1h55m	7h51m	27m	3h26m
OSoRA	-	-	23m	3h25m
RandLoRA	2h05m	7h55m	23m	-
GIVA	1h49m	6h53m	12m	2h57m

Table 1: Time taken for fine-tuning Qwen 2 on Commonsense-15K, OLMo 2 on MetaMathQA, and DINOv2-ViT-B/14 and CLIP-ViT-L/14 on the four image classification datasets.

Summary

GIVA uses the first-step full fine-tuning gradient to initialize its parameters.

$\Gamma = \mathbf{0}^{m \times m}$,

$\Lambda = \mathbb{I}^{r \times r}$,

$A = V_r^T$, and

B such that $B^T B = \mathbb{I}^{r \times r}$

- Requires lower ranks to match the performance of existing vector-based adaptation methods.
- Consequently, GIVA trains faster than the existing vector-based adaptation methods to achieve comparable results.
- Comparable performance to LoRA and RandLoRA.

Paper

Code

Questions?

Poster Location: 175