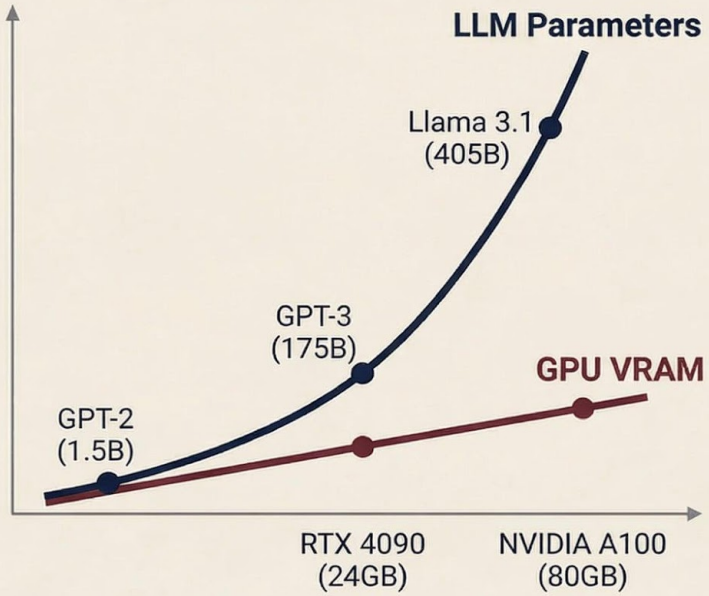
DiaBlo: Diagonal Blocks Are Sufficient For Finetuning

Selcuk Gurses, Aozhong Zhang, Yanxia Deng, Xun Dong,
Xin Li, Naigang Wang, Penghang Yin, Zi Yang

University at Albany, SUNY — IBM T.J. Watson Research Center

ICLR 2026

Background: The Growing Scale of LLMs



Full fine-tuning all parameters is increasingly impractical.

DiaBlo

Block Diagonal Reparameterization

DiaBlo fine-tunes only the diagonal blocks of the pretrained weight matrix $\mathbf{W_0}$ by adding a block-diagonal update matrix $\mathbf{D}$.

$$\mathbf{Y} = \mathbf{XW} = \mathbf{XW_0} + \mathbf{XD}$$

The adaptation matrix $\mathbf{D}$ is block diagonal

$$\mathbf{D} = \begin{pmatrix} \mathbf{D}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_N \end{pmatrix},$$

with $\mathbf{D_1}$, ..., $\mathbf{D_n}$ as trainable diagonal blocks.

Simple Initialization

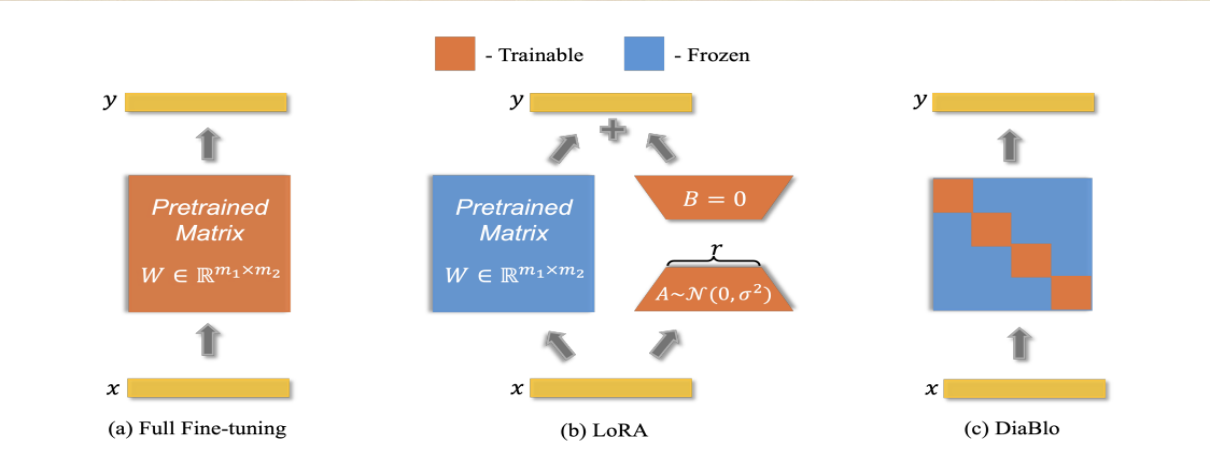
Zero initialization — no special schemes needed.

Efficient Implementation

$$\texttt{torch.einsum}(bNd_1, Nd_1d_2 \to bNd_2, \mathcal{X}, \mathcal{D})$$

ICLR 2026

Why DiaBlo Outperforms LoRA



- LoRA uses low-rank product $\Delta W = AB$ — non-convex optimization, sensitive to initialization.
- DiaBlo directly fine-tunes diagonal blocks — convex subproblems, simple zero initialization.
- DiaBlo's gradients enable more stable training.

DiaBlo eliminates the fundamental limitations of the low-rank matrix product.

ICLR 2026

Theoretical Guarantees

Theorem 1 (Linear Least Squares)

Suppose that X is a generic rank-r matrix. If the number of diagonal blocks $N \leq m_1/r$ is a common factor of m_1, m_2, then any solution to the DiaBlo-LSQ also solves the full-LSQ.

Theorem 2 (Nonlinear Stationary Point)

Suppose the activation matrix $X \in \mathbb{R}^{b \times m_1}$ and the linear output gradient $g_Y \in \mathbb{R}^{b \times m_2}$ are generic with ranks r_1 and r_2, respectively. If N is a common factor of m_1, m_2 and satisfies $N \geq \lceil r_1 N/m_1 \rceil \lceil r_2 N/m_2 \rceil$, then any stationary point of DiaBlo also yields a stationary point of the full FT objective.

What Do the Theorems Mean?

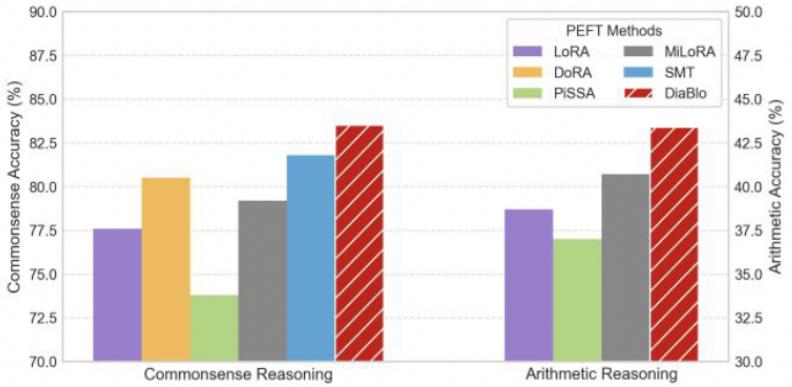
Theorem 1 Summary

In the linear setting, DiaBlo recovers the exact same solution as full fine-tuning — while being more parameter-efficient than LoRA.

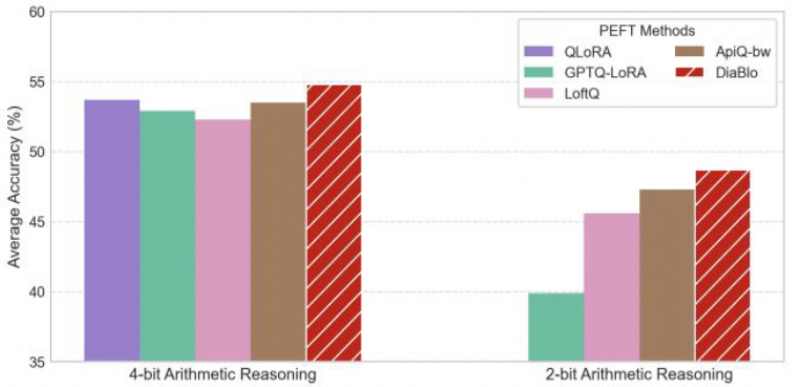
Theorem 2 Summary

Even in nonlinear deep networks, DiaBlo converges to critical points of the full fine-tuning objective with fewer trainable parameters.

Experiments: Reasoning Tasks



(a) Finetuning full-precision LLaMA2-7B



(b) Finetuning quantized LLaMA2-7B

DiaBlo achieves state-of-the-art on commonsense and arithmetic reasoning benchmarks.

Experiments: Code Generation & Safety

Model	Method	r/N	#Params	HumanEval Pass@1	HumanEval Pass@5	HumanEval Pass@10	HEx-PHI
Llama3-8B	Zero-shot	N/A	0%	28.5	39.8	45.2	78.7
	LoRA	$r = 32$	1.12%	34.7	46.4	50.8	91.6
	DoRA	$r = 32$	1.12%	33.1	44.0	48.6	93.6
	LoRI	$r = 32$	0.56%	**43.2**	**57.6**	<u>63.2</u>	92.8
	DiaBlo	$N = 64$	1.51%	**43.2**	<u>57.4</u>	**63.5**	**97.6**
	DiaBlo	$N = 128$	0.76%	<u>39.4</u>	55.0	61.9	<u>96.3</u>
Mistral-7B	Zero-shot	N/A	0%	28.6	38.2	41.9	80.5
	LoRA	$r = 32$	1.25%	33.8	42.4	45.3	91.9
	DoRA	$r = 32$	1.25%	33.7	42.6	46.8	<u>95.3</u>
	LoRI	$r = 32$	0.63%	33.8	42.0	45.1	94.7
	DiaBlo	$N = 64$	1.68%	**34.4**	<u>44.8</u>	<u>48.7</u>	**98.8**
	DiaBlo	$N = 128$	0.84%	<u>34.0</u>	**45.7**	**49.1**	**98.8**

DiaBlo outperforms LoRA variants on code generation and safety alignment.

Experiments: Sparsity Patterns

Method	Retained Sparse Update Accuracy (%)			Fine-tuning (Sparsity 1/64)	
	Sparsity 1/32	Sparsity 1/64	Sparsity 1/128	Accuracy (%)	Time (min)
DiaBlo	**67.30**	**65.47**	**63.11**	**67.68**	17.26
Random Entries	65.98	64.05	62.19	65.35	26.51
Random Block	65.96	63.75	62.98	64.86	29.76
Random Column	65.73	62.93	61.66	65.19	**17.01**
Random Row	49.14	24.08	2.85	61.71	17.76

DiaBlo's diagonal block pattern achieves the best accuracy across all sparsity levels.

Thank You!

Key Takeaways

- DiaBlo replaces LoRA's low-rank product with trainable diagonal blocks, eliminating gradient instability.
- Theoretically guaranteed to match full fine-tuning solutions in both linear and nonlinear settings.
- State-of-the-art results across reasoning, code generation, and safety alignment benchmarks.

Selcuk Gurses, Aozhong Zhang, Yanxia Deng, Xun Dong, Xin Li, Naigang Wang, Penghang Yin, Zi Yang

University at Albany, SUNY — IBM T.J. Watson Research Center