ToPA: Block-wise Toeplitz Adaptation for Expressive and Efficient Fine-Tuning

Sicong Li[1,2] **Qianqian Xu**[3,4*] **Zhiyong Yang**[5] **Zitai Wang**[3]
Longtao Huang[6] **Xiaochun Cao**[7] **Qingming Huang**[5,3*]
[1] Institute of Information Engineering, CAS
[2] School of Cyber Security, University of Chinese Academy of Sciences
[3] State Key Laboratory of AI Safety, Institute of Computing Technology, CAS
[4] Beijing Academy of Artificial Intelligence
[5] School of Computer Science and Tech., University of Chinese Academy of Sciences
[6] Alibaba Group
[7] School of Cyber Science and Tech., Shenzhen Campus of Sun Yat-sen University
lisicong@iie.ac.cn, {xuqianqian,wangzitai}@ict.ac.cn,
{yangzhiyong21,qmhuang}@ucas.ac.cn, kaiyang.hlt@alibaba-inc.com,
caoxiaochun@mail.sysu.edu.cn

Abstract

Parameter-Efficient Fine-Tuning (PEFT) has emerged as a prevalent strategy for adapting large pre-trained models. Among PEFT techniques, reparameterization-based methods, particularly those that employ low-rank or sparsity structures, have attracted significant attention. However, these approaches often underperform full fine-tuning. Specifically, low-rank methods constrain the update rank, which is misaligned with the inherently high-rank nature of full fine-tuning. In contrast, sparsity-based techniques limit the flexibility of the parameter space and compromise connectivity across weights. To overcome these limitations, we explore the use of Toeplitz matrices, whose entries are constant along each diagonal, therefore providing a compact parameterization without imposing explicit rank or sparsity constraints. A straightforward approach is to update weights via a product of Toeplitz matrices. However, similar to RNNs, long multiplicative chains often lead to gradient instability. To address this issue, we propose **ToPA**, a more refined version of this idea that replaces standard Toeplitz matrices with block-wise Toeplitz, greatly increasing each factor's capacity and shortening the chain, thereby stabilizing training and improving expressiveness. We theoretically show that products of block-wise Toeplitz matrices can approximate arbitrary matrices under shorter chain length, justifying the structural design of ToPA. Furthermore, we demonstrate that ToPA offers greater expressivity than existing low-rank and sparse parameterizations. Empirical evaluations across 21 NLP and CV datasets, spanning 5 model architectures, consistently validate the effectiveness of ToPA.

1 Introduction

In recent years, large language models (LLMs) have made significant advances in a wide range of fields [4, 53, 8, 31, 63]. Despite their strong generalization capabilities, applying LLMs to downstream tasks often requires additional fine-tuning. Fully fine-tuning all parameters, however, is computationally expensive and memory-intensive. To mitigate this challenge, parameter-efficient fine-tuning (PEFT) [21, 27, 77, 83] has emerged as a practical paradigm that adapts pretrained models by updating only a small subset of parameters.

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

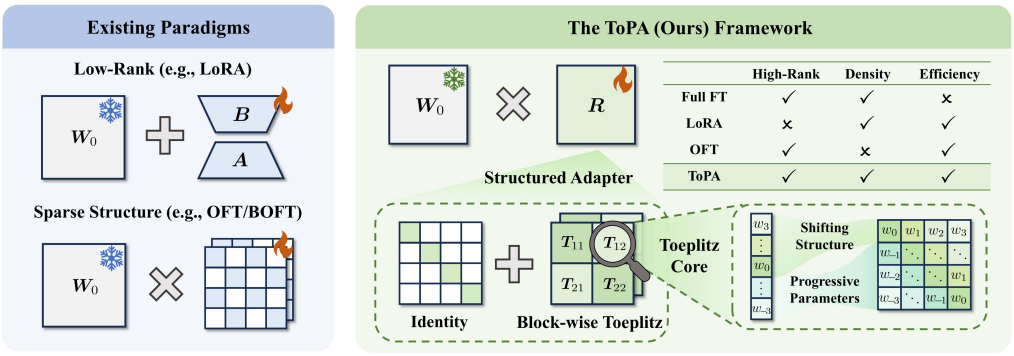


Figure 1: Low-rank methods (e.g., LoRA) adapt pretrained weights via additive low-rank updates, while sparsity-structured methods (e.g., OFT/BOFT) apply multiplicative sparse adapters. In contrast, ToPA adapts the model based on the block-wise Toeplitz matrices with identity residual design, enabling dense and high-rank adaptation while maintaining parameter efficiency.

Among PEFT approaches, one of the most representative methods is based on reparameterization, which represents the update matrix into an efficient structure based on a specific inductive bias. According to the selected structure, existing techniques typically follow two directions. One line is **low-rank decomposition**, exemplified by LoRA and its variants [22, 33, 38], which adds the product of two low-rank matrices. The other line adopts **sparse structure**, such as OFT [43] and BOFT [34], which multiplies one or multiple sparse orthogonal matrices to modulate pretrained weights.

Despite their empirical success, both paradigms still **fall short of full fine-tuning**. Low-rank methods are motivated by the assumption that over-parameterized models admit low intrinsic dimension [28, 1]. However, some recent work suggests that **low intrinsic dimension does not imply low rank** [25] and in fact full fine-tuning typically yields a high-rank update [26, 84]. Hence, the low-rank constraint does not lead to the desired inductive bias and thus could become a structural bottleneck, limiting expressiveness and potentially leading to underfitting. Sparsity structure methods impose a **fixed, predetermined sparsity pattern** coupled with orthogonality constraints on the adapter. Under this design, each column of the adapted weight is restricted to a predefined low-dimensional subspace spanned by **only a small set of** original columns. Moreover, sparsity masks most entries, disrupting connectivity among parameters and preventing interactions across dimensions, thereby blocking information propagation. Seeing the weakness of these two mainstream paradigms, we come to the following question:

Can we design a new compact parameterization to unleash the expressiveness of the PEFT method to catch up to full fine-tuning?

To this end, we turn to the **Toeplitz matrix**, a structured matrix whose entries are identical along each descending diagonal. As shown in Fig. 1, it inherently exhibits a shifting pattern with only $2n-1$ free parameters to form an $n \times n$ matrix. Intuitively, the shifting structure ensures that each row introduces an independent parameter, which almost surely avoids linear dependence and achieves full-rank. The absence of a fixed zero pattern ensures that efficiency herein is not tied to sparsity. Overall, Toeplitz matrices are **free from rank or sparsity constraints** yet maintain compactness. Moreover, matrix–vector multiplication with a Toeplitz matrix can be implemented efficiently in $\mathcal{O}(n \log n)$ time via Fast Fourier Transform (FFT). A classical result further shows that any matrix can be represented as a product of Toeplitz matrices. These properties make the Toeplitz matrix a promising candidate for expressive and efficient fine-tuning.

Building on these advantages, a naive implementation is to reparameterize pretrained weights by multiplying them with a chain of learnable Toeplitz matrices, whose parameter capacity can only be increased by stacking more Toeplitz factors. However, our theoretical and empirical analyses indicate that an overly long multiplication chain tends to cause gradient explosion, resembling the situation in RNNs [40]. To address this issue, we introduce two key improvements. First, we replace the plain Toeplitz matrix with its **block-wise variant**. By partitioning the matrix into p^2 blocks, each factor contains $p(2n-p)$ learnable parameters, which is approximately p times that of a standard Toeplitz

matrix. Consequently, under the same parameter budget, the required chain length can be reduced by roughly a factor of p. Second, we incorporate an **identity matrix** to enable residual learning, which further stabilizes optimization. Together, these refinements lead to our proposed **ToPA** framework, as illustrated in Fig. 1.

Theoretically, we show that ToPA enables high-rank and dense adaptation, effectively addressing the limitations of previous methods. Moreover, we prove that the product of block-wise Toeplitz matrices can represent any matrix under a shorter chain length than the classical ones. Empirically, we conduct extensive experiments on diverse tasks, including natural language understanding, subject-driven generation, commonsense reasoning, and math & code generation, using multiple models such as DeBERTaV3, Stable-Diffusion, LLaMA-2/3, and Qwen3. The results show that ToPA consistently outperforms strong baselines, validating its effectiveness.

In summary, our key contributions are as follows:

- We propose ToPA, a new PEFT technique that exploits block-wise Toeplitz reparameterization to achieve high expressiveness with a compact parameter budget.
- Theoretical analysis shows that ToPA can achieve high-rank and dense adaptation, enabling stronger expressiveness than prior approaches.
- Extensive experiments across multiple benchmarks and models show that ToPA consistently surpasses multiple baselines, highlighting its effectiveness.

2 Preliminaries

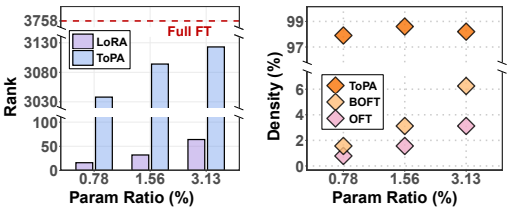
This section briefly reviews two mainstream structural constraints used by existing reparameterization approaches, *i.e.*, low-rank and sparsity. Meanwhile, we also highlight their inherent limitations that may hinder expressiveness and adaptation flexibility.

2.1 Low-rank Decomposition and its Limitation

As one of the most representative PEFT methods, LoRA [22] decomposes the update parameter as the multiplication of two low-rank matrices $\boldsymbol{B} \in \mathbb{R}^{m\times r}, \boldsymbol{A} \in \mathbb{R}^{r\times n}$:

$$\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{A}, \tag{1}$$

where $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ denotes the pretrained weights, and the rank $r \ll \min(m, n)$.



(a) Rank of the update matrix (b) Sparsity of adapter $\boldsymbol{R}$

Figure 2: Fine-tuning LLaMA-2-7B for mathematical reasoning under different parameter budgets. **(a)** Average rank of the weight update matrix $\Delta\boldsymbol{W}$. **(b)** Average density of the adapter $\boldsymbol{R}$ across different methods, measured by the non-zero ratio (NNZ).

The success of low-rank stems from the inductive bias that the over-parameterized models possess low intrinsic dimension [28, 1]. However, prior arts point out that low dimension is not equivalent to low rank. In fact, full fine-tuning generally produces update matrices with high rank. Therefore, the **rank constraint** inherently restricts the model's expressiveness, leading to a performance gap relative to full fine-tuning.

To substantiate this, we compute the rank of the updated query and value attention matrices of the Llama2-7B model fine-tuned via LoRA for math reasoning task with varying parameter ratios. As illustrated in Fig. 2(a), the rank of full fine-tuning is about 3758, significantly higher than those of LoRA (e.g., $16, 32, 64$). Hence, this low-rank structure creates a severe bottleneck that constrains the model from adequately modeling complex parameter transformations, as corroborated by prior studies [26, 25, 84].

2.2 Sparsity Structure and its Limitations

Sparsity structure approaches (*e.g.*, OFT [43] and BOFT [34], whose details are provided in App. B) reparameterize the updated weights by multiplying a series of sparse orthogonal matrices $\boldsymbol{B}_i \in \mathbb{R}^{n\times n}$:

$$\boldsymbol{W} = \boldsymbol{W}_0 \cdot \boldsymbol{R} = \boldsymbol{W}_0 \cdot (\prod_{i=1}^{m} \boldsymbol{B}_i), \tag{2}$$

Despite their effectiveness, these methods enforce a fixed, predetermined sparsity pattern where non-zero entries are specified **a priori** rather than learned. To illustrate this, we have the following reformulation:

$$(\boldsymbol{W}_0\boldsymbol{R})_{:,j} = \sum_{i:r_{i,j}\neq 0} r_{i,j}(\boldsymbol{W}_0)_{:,i} \in \operatorname{span}\big\{(\boldsymbol{W}_0)_{:,i}\big|r_{i,j}\neq 0\big\}. \tag{3}$$

First, each adapted column $(\boldsymbol{W}_0\boldsymbol{R})_{:,j}$ is restricted to a subspace $\operatorname{span}\big\{(\boldsymbol{W}_0)_{:,i}\big|r_{i,j}\neq 0\big\}$, which is manually fixed before training instead of being learned. Second, due to the masked entries in $\boldsymbol{W}_0$, the imposed sparsity pattern severs a large fraction of potential cross-dimensional interactions, greatly reducing parameter connectivity and restricting information flow.

These issues are particularly evident in practical PEFT settings where the parameter budget is tight. As shown in Fig. 2(b), when fine-tuning the query and value projections of LLaMA-2-7B using OFT and BOFT for the math reasoning task, we observe that the **NoN-Zero ratio (NNZ) of $\boldsymbol{R}$ is exceedingly low**, ranging from approximately 0.5% to 6.5%. This highly sparse $\boldsymbol{R}$ can substantially hinder adaptation flexibility and thus degrade downstream performance.

3 Methodology

Based on the analysis in Sec. 2, this section explores an alternative reparameterization beyond low-rank and sparsity.

3.1 Toeplitz Matrix

Originating from Otto Toeplitz's early 20th-century work on spectral theory [55], a Toeplitz matrix is a structured matrix in which each descending diagonal from left to right is constant:

$$\boldsymbol{T}(\boldsymbol{w}) = \begin{bmatrix} w_0 & w_1 & \cdots & w_{n-1} \\ w_{-1} & w_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & w_1 \\ w_{-n+1} & \cdots & w_{-1} & w_0 \end{bmatrix} \in \mathbb{R}^{n\times n}, \tag{4}$$

where $\boldsymbol{w} = (w_{-n+1}, \ldots, w_0, \ldots, w_{n-1})^\top$ defines the elements along the diagonals.

An appealing property of the Toeplitz matrix is that its structure is **efficient** in nature. For parameter efficiency, an $n\times n$ Toeplitz matrix is fully determined by its defining vector $\boldsymbol{w} \in \mathbb{R}^{2n-1}$, significantly lower than n^2 for general matrices. For computational efficiency, its matrix–vector multiplication can be efficiently implemented using the Fast Fourier Transform (FFT). This reduces the computational complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n\log n)$. Details on the acceleration mechanism are provided in App. D.

Another desired property is its **expressiveness**. Intuitively, this shifting structure endows each row with a distinct parameter, thereby almost surely eliminating linear dependence and resulting in a full-rank matrix. Moreover, since no deterministic zero pattern is enforced, the compactness does not rely on sparsity. Overall, this structured parameter matrix is **neither restricted by rank nor by sparsity**. Moreover, a classical result in matrix theory establishes that any matrix can be factored as a product of Toeplitz matrices:

Theorem 3.1. *[73] Every $n\times n$ matrix is generically a product of $\lfloor \frac{n}{2} \rfloor + 1$ Toeplitz matrices and always a product of at most $2n+5$ Toeplitz matrices.*

This result indicates that its expressiveness matches that of full fine-tuning under a sufficient chain length.

3.2 A Naive Implementation

Given the aforementioned properties, it is natural to introduce the Toeplitz matrix into PEFT. A naive implementation is to reparameterize the adapted weight $\boldsymbol{W}$ by multiplying the pretrained weight $\boldsymbol{W}_0$ with a chain of learnable Toeplitz matrices:

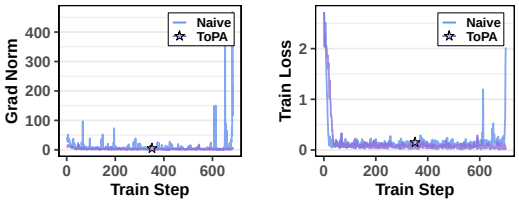
$$\boldsymbol{W} = \boldsymbol{W}_0 \cdot \boldsymbol{R} = \boldsymbol{W}_0 \cdot (\prod_{i=1}^{t} \boldsymbol{T}_i), \tag{5}$$

where each $\boldsymbol{T}_i$ is initialized as the identity matrix $\boldsymbol{I}$, ensuring $\boldsymbol{W} = \boldsymbol{W}_0$ at the start of training. However, this design suffers from a key practical limitation: repeatedly multiplying an overly long chain of Toeplitz matrices can lead to numerical instability, typically manifesting as gradient explosion. To illustrate this issue, we fine-tune LLaMA-3-8B on the commonsense reasoning task with the chain length $t = 16$ (with the matched number of parameters as LoRA$_{r=16}$). As shown in Fig. 3, we track both the Frobenius norm of gradients and the training loss during the first 700 steps. Both metrics exhibit substantial fluctuations, indicating unstable optimization and degraded learning dynamics.

To provide theoretical insights, we investigate this instability stems from the perspective of the gradient structure. In particular, the gradient with respect to $\boldsymbol{T}_i$ contains a product of all preceding and following Toeplitz factors:

Proposition 3.2 (Gradient of Toeplitz chain)**.** *Let $\mathcal{L}$ denote the scalar loss, and define the gradient with respect to $\boldsymbol{W}$ as $\boldsymbol{G} := \nabla_{\boldsymbol{W}}\mathcal{L} \in \mathbb{R}^{m\times n}$. Then, for each $i \in \{1, \dots, t\}$, the gradient with respect to $\boldsymbol{T}_i$ is given by:*

$$\nabla_{\boldsymbol{T}_i}\mathcal{L} = \Big(\prod_{k=1}^{i-1} \boldsymbol{T}_k\Big)^\top \boldsymbol{W}_0^\top \boldsymbol{G} \Big(\prod_{k=i+1}^{t} \boldsymbol{T}_k\Big)^\top. \tag{6}$$



(a) Gradient Norm (b) Train Loss

Figure 3: Training dynamics over the first 700 steps of fine-tuning LLaMA-3-8B on commonsense reasoning tasks using the naive method ($t = 16$) and ToPA$^{t=2}_{p=8}$ with matched parameters. As training proceeds, the naive method collapses, so only the first 700 steps are shown.

Eq. (6) reveals that the gradient computation involves a cumulative product of matrices across the chain. Consequently, if the spectral norms of these intermediate matrices exceed unity, which is common during training, the gradient tends to grow exponentially as the chain length t increases. This phenomenon is analogous to the well-known gradient explosion in recurrent neural networks (RNNs) [40].

3.3 ToPA: Block-wise Toeplitz Adaptation

The instability discussed in Sec. 3.2 stems from the fixed parameter budget in a single Toeplitz matrix. As a result, a long chain becomes the only choice for greater model capacity, inevitably amplifying the instability. To avoid an overly long chain, a simple yet effective method is to increase the budget in a single matrix. In view of this, we replace the plain Toeplitz matrix with the **block-wise Toeplitz matrix** to:

$$\boldsymbol{T}_{\text{blk}} = \begin{bmatrix} \boldsymbol{T}_{11} & \cdots & \boldsymbol{T}_{1p} \\ \vdots & \ddots & \vdots \\ \boldsymbol{T}_{p1} & \cdots & \boldsymbol{T}_{pp} \end{bmatrix} \in \mathbb{R}^{n\times n}. \tag{7}$$

Here, each sub-block $\boldsymbol{T}_{ij} \in \mathbb{R}^{\frac{n}{p}\times\frac{n}{p}}$ is an independent Toeplitz matrix, where p denotes the number of blocks along each dimension. On top of this partition, $\boldsymbol{T}_{\text{blk}}$ yields $p(2n-p)$ parameters, approximately p times that of the standard Toeplitz matrix. This reformulation not only allows the chain length to be reduced by a factor of p under the same parameter budget, but also maintains the same gradient *w.r.t.* $\boldsymbol{T}^i_{\text{blk}}$ as that in Eq. (6), significantly promoting training stability.

To further stabilize optimization, we introduce a residual connection to our final framework, **ToPA**:

$$\boldsymbol{W} = \boldsymbol{W}_0 \cdot \boldsymbol{R} = \boldsymbol{W}_0 \cdot (\prod_{i=1}^{t} \boldsymbol{T}^i_{\text{blk}} + \boldsymbol{I}), \tag{8}$$

where $\boldsymbol{I}$ is the identity matrix. This residual term centers the adaptation around the identity mapping, effectively stabilizing optimization [17]. Moreover, it imposes an inductive bias that keeps the adapted weights close to the pretrained initialization, which aligns with the fine-tuning settings.

As shown in Fig. 3, ToPA demonstrates markedly smoother training dynamics compared to the naive implementation under the same parameter budget, confirming the effectiveness of our approach.

3.4 Initialization Strategy of ToPA

To ensure numerical stability during fine-tuning, the model initialization should preserve the exact behavior of the pretrained network. Formally, this requires that the learnable modules contribute zero output at the start of training, *i.e.*, $\prod_{i=1}^{t} \boldsymbol{T}_{\text{blk}}^{i} = 0$. We implement this via an asymmetric initialization strategy analogous to LoRA, where $\boldsymbol{B} = 0$ and $\boldsymbol{A} \sim \mathcal{N}(0, \sigma^2)$. Formally, the first block-wise Toeplitz matrix is initialized by $\mathbf{0}$, while the others are drawn from the normal distribution:

$$\boldsymbol{w}_{ij}^{1} = 0,\ \boldsymbol{w}_{ij}^{2}, \ldots, \boldsymbol{w}_{ij}^{t} \sim \mathcal{N}(0, \sigma^2),\ i, j \in [1, p], \tag{9}$$

where $\boldsymbol{w}_{ij}^{k} \in \mathbb{R}^{2n/p-1}$ denotes the defining vector of (i, j) block of k-th block-wise Toeplitz matrix.

4 Theoretical Analysis

4.1 Achieving High-Rank and Dense Adaptation

High-rank Update. Owing to the block-wise Toeplitz structure in Eq. (7), each factor $\boldsymbol{T}_{\text{blk}}^{i}$ in ToPA is almost surely invertible. Thus, the resulting update matrix $\Delta\boldsymbol{W} := \boldsymbol{W}_0 \cdot (\prod_{i=1}^{t} \boldsymbol{T}_{\text{blk}}^{i})$ is capable of **achieving high or even full rank**. This property is formally stated in the following proposition:

Proposition 4.1. *Assume that* $\text{rank}(\boldsymbol{W}_0) = s$. *Then,* $\Delta\boldsymbol{W}$ *has rank* s *almost surely, i.e.,*

$$\mathbb{P}(\text{rank}(\Delta\boldsymbol{W}) = s) = 1. \tag{10}$$

Since the pretrained weights $\boldsymbol{W}_0$ typically exhibit high rank, ToPA naturally performs expressive, high-rank updates, effectively circumventing the representational bottlenecks of low-rank adaptation methods discussed in Sec. 2. Empirically, Fig. 2(a) confirms our analysis, showing that ToPA yields substantially higher rank updates than LoRA, under comparable parameter budgets.

Dense Transformation with Parameter Efficiency. Unlike sparsity structured methods, each block-wise Toeplitz matrix $\boldsymbol{T}_{\text{blk}}^{i}$ remains dense while requiring only $p(2n - p)$ parameters, as shown in Eq. (4). This dense parameterization yields two critical advantages. First, it restores cross-dimensional interactions severed by rigid sparsity masks, facilitating global information flow. Second, it liberates the adapted columns from being confined to a predetermined subspace, enabling the model to explore the complete column space of $\boldsymbol{W}_0$. By effectively circumventing these limitations, ToPA preserves high expressive power. As shown in Fig. 2(b), the resulting adapter $\boldsymbol{R}$ is substantially denser than those produced by OFT and BOFT, reinforcing our analysis.

4.2 Expressive Power of ToPA

❶ **Compositional Expressiveness.** Recall that Thm. 3.1 substantiates the strong expressiveness of standard Toeplitz matrices. Hence, it is natural to extend this compelling result to the block-wise Toeplitz matrix:

Theorem 4.2 (Informal)**.** *Every* $n \times n$ *matrix is generically a product of* $\lfloor \frac{n}{2p} \rfloor + 1$ *block-wise Toeplitz matrices and always a product of at most* $\frac{2n}{p} + 5$ *block-wise Toeplitz matrices (with each Toeplitz block of size* n/p*).*

This theorem suggests that a composition of block-wise Toeplitz matrices can represent any arbitrary matrix under a shorter chain length than the classical results.

❷ **Stronger Expressive Power.** Beyond compositional expressiveness, ToPA can exhibit stronger expressive power than low-rank decomposition and sparse structure methods for certain transformations. Specifically, we show that there exists a family of adaptations that can be compactly represented by ToPA but cannot be efficiently captured by either individual approach:

Table 1: Performance comparison on commonsense reasoning tasks with LLaMA-2-7B and LLaMA-3-8B backbones.

Model	Method	# Param	BoolQ	PIQA	SIQA	ARC-c	ARC-e	OBQA	HellaS	WinoG	Avg
LLaMA2-7B	LoRA$_{r=32}$	16.8M	70.5	81.1	80.1	71.5	85.4	78.6	84.6	78.5	78.8
	DoRA$_{r=32}$	17.0M	70.7	81.3	79.5	71.2	85.6	78.2	84.8	78.4	78.7
	HiRA$_{r=32}$	16.8M	70.6	82.2	79.3	69.8	85.9	79.4	86.5	79.8	79.2
	RidgeLoRA$_{r=32}$	17.0M	71.2	83.5	**81.1**	71.5	87.0	79.8	87.3	82.6	80.5
	OFT$_{b=64}$	16.8M	70.5	83.1	80.2	69.9	86.6	80.4	86.5	82.0	79.9
	BOFT$^{m=2}_{b=32}$	17.0M	70.6	83.2	79.2	70.1	**87.4**	77.6	82.4	78.6	78.6
	HRA$_{r=64}$	16.8M	69.9	81.8	78.1	68.8	85.5	76.4	80.6	82.0	77.9
	ToPA$^{t=2}_{p=16}$	16.7M	**71.5**	**84.5**	80.9	**72.5**	85.6	**83.6**	**87.7**	**84.5**	**81.4**
LLaMA3-8B	LoRA$_{r=32}$	13.6M	73.0	88.8	80.2	80.4	92.8	84.8	93.5	86.0	84.9
	DoRA$_{r=32}$	13.8M	72.9	88.7	80.1	80.9	93.0	85.6	93.8	86.2	85.2
	HiRA$_{r=32}$	13.6M	74.0	89.0	81.8	**83.0**	92.7	86.4	**95.1**	86.3	86.0
	RidgeLoRA$_{r=32}$	17.0M	73.7	88.3	81.7	81.5	92.9	88.2	91.7	85.2	85.4
	OFT$_{b=64}$	16.9M	74.6	88.5	81.6	82.1	**93.2**	87.4	94.3	85.6	85.9
	BOFT$^{m=3}_{b=16}$	12.8M	72.9	88.6	81.2	80.5	91.8	84.6	93.5	84.8	84.7
	HRA$_{r=64}$	13.6M	71.1	88.6	79.2	79.3	92.6	84.8	92.6	83.8	84.0
	ToPA$^{t=3}_{p=8}$	12.6M	**75.1**	**89.9**	**82.1**	81.5	92.2	**88.4**	94.4	**86.5**	**86.3**

Table 2: Performance comparison on math reasoning and code generation tasks with the LLaMA-3-8B backbone.

Method	# Param	GSM8K	MATH	HumanEval	MBPP
LoRA$_{r=32}$	13.6M	70.7	22.5	50.0	**67.5**
DoRA$_{r=32}$	13.8M	71.1	22.7	51.2	66.9
PiSSA$_{r=32}$	13.6M	72.1	23.0	50.0	65.1
HiRA$_{r=32}$	13.6M	68.9	22.1	50.6	67.2
RidgeLoRA$_{r=32}$	16.9M	69.8	21.7	48.2	65.9
OFT$_{b=64}$	17.0M	73.8	22.9	45.1	66.1
BOFT$^{m=3}_{b=16}$	12.8M	69.4	22.4	45.1	**67.5**
ToPA$^{t=3}_{p=8}$	12.6M	**74.7**	**23.5**	**51.8**	67.2

Table 3: Performance comparison on math reasoning and code generation tasks with the Qwen3-4B backbone.

Method	# Param	GSM8K	MATH	HumanEval	MBPP
LoRA$_{r=16}$	5.9M	82.6	47.0	69.5	71.2
DoRA$_{r=16}$	5.9M	83.2	46.6	68.9	72.2
PiSSA$_{r=16}$	5.9M	83.5	47.4	70.1	73.0
HiRA$_{r=16}$	5.9M	83.6	46.0	67.7	71.2
RidgeLoRA$_{r=16}$	5.9M	83.7	46.5	66.5	71.2
OFT$_{b=32}$	6.1M	83.8	47.6	69.5	73.8
BOFT$^{m=2}_{b=16}$	6.1M	83.1	47.8	67.7	**74.6**
ToPA$^{t=3}_{p=5}$	5.5M	**84.3**	**48.2**	**73.2**	73.5

Theorem 4.3. *There exists a class of adapted weight matrices $\boldsymbol{W}$, that can be realized by ToPA with $\mathcal{O}(n)$ trainable parameters, while achieving the same $\boldsymbol{W}$ with LoRA at least requires $\mathcal{O}(n^2)$ parameters and with OFT or BOFT requires at least $\mathcal{O}(n \log n)$ parameters.*

This theorem indicates that ToPA can realize certain transformations with significantly fewer parameters than low-rank or sparsity methods, which highlights ToPA's superior expressive capacity.

5 Experiments

5.1 Experiments Protocols

Datasets. We evaluate ToPA on a broad range of NLP and CV tasks. **(1) Commonsense Reasoning**: We train on Commonsense170K [23] dataset and evaluate on eight commonsense benchmarks. **(2) Math Reasoning & Code Generation**: For math reasoning, we train on a 100k subset of MetaMathQA [75] dataset and evaluate on GSM8K [12] and MATH [75] datasets. For code generation, we train on CodeFeedback [82] dataset, and evaluate on HumanEval [8] and MBPP [2]. **(3) Subject-Driven Generation**: We adopt the DreamBooth [48] dataset. **(4) Natural Language Understanding**: We use the GLUE [57] benchmark. For details of these datasets, please refer to App. E.1.

Competitors. We compare ToPA with representative methods, including low-rank techniques such as LoRA [22], DoRA [33], and PiSSA [38]. We also consider high-rank variants of LoRA, such as HiRA [25] and RidgeLoRA [84]. In addition, we include sparsity approaches, including OFT [43] and BOFT [34], and their variant HRA [76]. Details of these baselines can be found in App. E.2.

Implementation Details. We use task-specific configurations across different evaluation settings. For commonsense reasoning, we fine-tune LLaMA-2-7B [56] and LLaMA-3-8B [52]. For mathematical reasoning and code generation, we evaluate on LLaMA-3-8B and Qwen3-4B [67]. For subject-driven generation, we adopt Stable Diffusion v1.5 [46]. Finally, for NLU tasks, we utilize DeBERTaV3-base [18]. Additional details are provided in App. E.3.



Figure 4: Qualitative comparison of results on subject-driven generation tasks.

5.2 Commonsense Reasoning

Tab. 1 reports results on commonsense reasoning tasks. Across both LLaMA-2-7B and LLaMA-3-8B backbones, ToPA achieves the best average performance while maintaining a comparable or smaller parameter count. Specifically, ToPA attains a leading average score of 81.4% on LLaMA-2-7B and 86.3% on LLaMA-3-8B, consistently surpassing all baseline methods. These findings demonstrate that ToPA enables highly effective and efficient adaptation for commonsense reasoning tasks.

5.3 Math Reasoning & Code Generation

Tab. 2 and Tab. 3 report ToPA's performance on math and code generation tasks with LLaMA-3-8B and Qwen3-4B. Overall, ToPA consistently surpasses all baseline methods on GSM8K, MATH, and HumanEval benchmarks with comparable or fewer parameters. Specifically, with LLaMA-3-8B, ToPA achieves 74.7% on GSM8K and 51.8% on HumanEval, exceeding the strongest baseline by 0.9% and 0.6%, respectively. On Qwen3-4B, ToPA achieves the best performance, with 84.3% on GSM8K and 73.2% on HumanEval. Although ToPA is slightly below the leading method on MBPP, the gap is marginal. These results confirm ToPA's effectiveness in both math and code generation.

5.4 Subject-Driven Generation

Tab. 4 presents the quantitative evaluation of subject-driven generation on Stable Diffusion v1.5. Under a comparable or reduced parameter budget, ToPA achieves superior performance on DINO [5], CLIP-I [44], and CLIP-T [44] metrics. Given that DINO and CLIP-I quantify subject fidelity while CLIP-T measures prompt alignment, these results indicate that ToPA excels at preserving subject identity and adhering to textual instructions. Although there is a marginal trade-off in sample diversity as indicated by the LPIPS [79] metric, the overall performance remains robust. These findings are visually corroborated in Fig. 4, where ToPA consistently outperforms baselines like LoRA and OFT, yielding images with higher fidelity and superior prompt alignment across diverse subjects.

Table 4: Performance comparison on subject-driven generation.

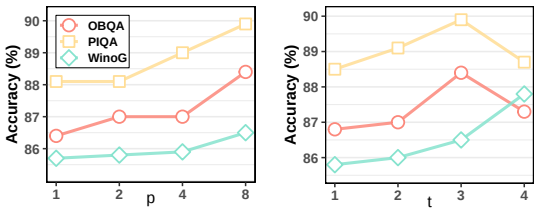
Method	# Param	DINO↑	CLIP-I↑	CLIP-T↑	LPIPS↑
LoRA$_{r=64}$	12.8M	0.654	0.793	0.230	**0.743**
DoRA$_{r=64}$	12.8M	0.654	0.791	0.231	0.740
PiSSA$_{r=64}$	12.8M	0.634	0.782	0.228	0.739
HiRA$_{r=64}$	12.8M	0.668	0.794	0.226	0.736
RidgeLoRA$_{r=64}$	12.8M	0.657	0.791	0.222	0.737
OFT$_{b=64}$	23.3M	0.653	0.793	0.225	0.735
BOFT$^{m=2}_{b=32}$	23.4M	0.664	**0.796**	0.225	0.734
ToPA$^{t=2}_{p=32}$	11.5M	**0.676**	**0.796**	**0.234**	0.740

5.5 Natural Language Understanding

Tab. 5 demonstrates the performance of different methods for fine-tuning the DeBERTaV3-base model on the GLUE benchmark, with all methods limited to about 0.44M parameters. The results indicate that ToPA achieves the best average score of 88.75, outperforming several baselines. In

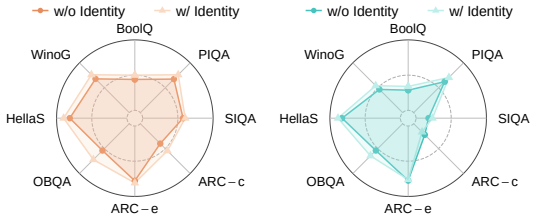
Table 5: Performance comparison on the GLUE benchmark using the DeBERTaV3-base model. We report Matthew's correlation for CoLA, Pearson correlation for STS-B, matched accuracy for MNLI, and accuracy for the other datasets.

Method	# Param	MNLI	SST-2	CoLA	QQP	QNLI	RTE	MRPC	STS-B	Avg
LoRA$_{r=8}$	0.44M	89.92	94.72	69.31	91.68	93.99	84.48	89.46	91.12	88.09
DoRA$_{r=8}$	0.47M	89.67	94.50	68.86	91.55	**94.23**	85.92	89.46	91.56	88.22
PiSSA$_{r=8}$	0.44M	89.90	95.07	68.52	91.78	93.98	86.64	89.22	91.11	88.28
HiRA$_{r=8}$	0.44M	90.20	**95.64**	69.80	91.08	93.26	**87.00**	89.22	91.45	88.46
RidgeLoRA$_{r=8}$	0.47M	89.74	95.18	68.82	91.57	93.98	85.20	88.97	91.12	88.07
OFT$_{b=16}$	0.44M	89.55	94.38	68.97	91.25	93.56	84.48	88.73	91.24	87.77
BOFT$^{m=2}_{b=8}$	0.47M	89.46	94.27	68.12	91.32	93.63	85.20	88.97	90.98	87.74
HRA$_{r=16}$	0.44M	**90.24**	95.41	70.80	91.24	93.70	85.20	89.71	91.11	88.43
ToPA$^{t=2}_{p=4}$	0.44M	90.15	95.18	**70.88**	**91.85**	94.05	86.28	**89.95**	**91.68**	**88.75**



(a) Impact of p (b) Impact of t

Figure 5: Ablation study on the effects of p and t in ToPA. We vary p or t and fine-tune LLaMA-3-8B using ToPA on commonsense reasoning tasks.



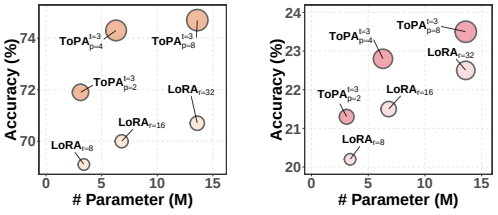
(a) LLaMA-2-7b (b) LLaMA-3-8b

Figure 6: Ablation of the identity residual in ToPA on commonsense reasoning tasks when fine-tuning LLaMA-2-7B and LLaMA-3-8B.

particular, ToPA yields a 0.29% gain in the average score and consistently ranks first or second across eight tasks. These results show that ToPA has a stronger capability under the same parameter budget.

5.6 Performance Comparison across Parameter Scales

To evaluate ToPA across different parameter scales, we compare it with LoRA on mathematical reasoning tasks using LLaMA-3-8B. LoRA uses ranks $r \in \{8, 16, 32\}$, while ToPA fixes $t = 3$ and varies $p \in \{2, 4, 8\}$ for comparable parameter budgets. As shown in Fig. 7, ToPA outperforms LoRA at each matched scale on both GSM8K and MATH, demonstrating its effectiveness.



(a) GSM8K (b) MATH

Figure 7: Performance across parameter scales. We compare ToPA and LoRA on math reasoning tasks using LLaMA-3-8B.

5.7 Influence of p and t

Impact of p: Fig. 5(a) examines the impact of the block number p. Concretely, we fix the chain length t to 3 and vary p over $\{1, 2, 4, 8\}$. As p increases, the number of learnable parameters grows accordingly, leading to a stronger expressive capacity for ToPA. Consequently, we observe a consistent performance gain on all three datasets, where larger p generally yields higher accuracy.

Impact of t: Fig. 5(b) investigates the impact of chain length t. We fix the block number p to 8 and vary t in $\{1, 2, 3, 4\}$. As t increases from 1 to 3, performance consistently improves on all three benchmarks, indicating stronger expressiveness with longer chains. However, when t reaches 4, accuracy slightly drops on OBQA and PIQA, which may be attributed to the optimization instability of longer chains, leading to suboptimal training, as discussed in Sec. 3.

5.8 Ablation Study of Identity Residual

To evaluate the contribution of the identity residual in ToPA, we conducted an ablation study by fine-tuning LLaMA-2-7B and LLaMA-3-8B on commonsense reasoning benchmarks. As illustrated in Fig. 6, incorporating the identity matrix consistently enhances performance across both backbones and all evaluated datasets, confirming its effectiveness.

6 Related Work

6.1 Reparameterization-Based Methods

Among PEFT approaches, reparameterization-based methods are widely adopted. These methods reparameterize large weight matrices into compact auxiliary components, such that only the lightweight parameters are updated during fine-tuning. After training, the auxiliary parameters can be merged back into the original weights without increasing inference latency, enabling efficient adaptation with minimal computational overhead.

Most existing methods typically follow two main directions. The first is low-rank decomposition, such as LoRA [22], which factorizes weight updates into the products of two low-rank matrices to achieve parameter efficiency. To further enhance performance, numerous variants have been proposed. For instance, DoRA [33] decomposes the weight into magnitude and direction components, directly tuning the magnitude while applying LoRA to the direction, thereby improving learning capacity. PiSSA [38] applies SVD to decompose each pretrained weight matrix into principal and residual components, updating the principal part while freezing the residual part to enable faster convergence. HiRA [25] expresses the update matrix as the Hadamard product between the frozen pretrained weight and a low-rank adapter, thereby achieving a higher-rank and more expressive adaptation. RidgeLoRA [84] replaces LoRA's parallel structure with a series connection and augments the low-rank adapter with a diagonal ridge term, enabling high-rank weight updates with improved representation capacity. The second category introduces sparsity-based reparameterization. OFT [43] reparameterizes the pretrained matrix using a sparse orthogonal matrix to preserve pretrained knowledge. BOFT [34] extends OFT by replacing the single sparse orthogonal matrix with a product of multiple sparse butterfly matrices, thereby further enhancing model expressiveness. PSOFT [65] confines orthogonal transformations to a low-rank principal subspace, effectively preserving pretrained geometry while improving efficiency and expressiveness.

6.2 Other PEFT Methods

Beyond reparameterization strategies, a number of other PEFT approaches have been proposed. The prompt-based family learns task-specific conditioning signals without updating backbone weights. Prompt Tuning [27, 83] optimizes a small set of continuous embeddings prepended to the input, whereas Prefix Tuning [30] trains key–value prefixes in attention layers to guide the model at the hidden-state level. Adapter-based methods instead insert lightweight bottleneck modules into each transformer block. The Adapter framework [21] captures task-specific transformations while keeping the pretrained weights frozen, and AdaptFormer [9] enhances this design with residual adapter branches that increase adaptation capacity at minimal parameter cost. Finally, selective-parameter tuning methods update only a very small subset of existing parameters: BitFit [77] adjusts only bias terms to obtain competitive performance with negligible overhead, while IA^3 [32] learns multiplicative gating vectors over attention and feed-forward activations to enable highly parameter-efficient task conditioning.

7 Conclusion

In this work, we identify the limitations of existing reparameterization paradigms that rely on low-rank or sparse structures. To address these challenges, we introduce ToPA. This block-wise Toeplitz framework enables high-rank, dense parameter updates while maintaining parameter efficiency and optimization stability. Theoretically, we prove that ToPA possesses compositional expressiveness and can offer superior expressive capacity compared to low-rank and sparse methods. Extensive experiments across diverse models and tasks validate the effectiveness of our approach.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China: 62525212, U23B2051, 62236008, 62441232, 62521007, U2541229, U21B2038, 62576332 and 62502500, in part by Youth Innovation Promotion Association CAS, in part by the Strategic Priority Research Program of the Chinese Academy of Sciences, Grant No. XDB0680201, in part by the project ZR2025ZD01 supported by Shandong Provincial Natural Science Foundation, in part by Ningbo Science and Technology Innovation 2025 Major Project 2025Z027, in part by the China National Postdoctoral Program for Innovative Talents under Grant BX20240384, in part by Beijing Natural Science Foundation–Xiaomi Innovation Joint Fund (Grant No. 26L030072), in part by General Program of the Chinese Postdoctoral Science Foundation under Grant No. 2025M771558, and in part by Beijing Natural Science Foundation under Grant No. L252144.

References

[1] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Annual Meeting of the Association for Computational Linguistics, pages 7319–7328, 2021.

[2] Jacob Austin, Augustus Odena, Maxwell I. Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. Program synthesis with large language models. CoRR, 2021.

[3] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: reasoning about physical commonsense in natural language. In The AAAI Conference on Artificial Intelligence, pages 7432–7439, 2020.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Annual Conference on Neural Information Processing Systems, 2020.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In IEEE/CVF International Conference on Computer Vision, pages 9630–9640, 2021.

[6] Richard Caron and Tim Traynor. The zero set of a polynomial. WSMR Report, pages 05–02, 2005.

[7] Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In International Workshop on Semantic Evaluation, pages 1–14, 2017.

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. CoRR, abs/2107.03374, 2021.

[9] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In Annual Conference on Neural Information Processing Systems, 2022.

[10] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,, pages 2924–2936, 2019.

[11] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. CoRR, 2018.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. CoRR, 2021.

[13] David A. Cox, John Little, and Donal O'Shea. Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.). Springer, 1997.

[14] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In International Workshop on Paraphrasing, 2005.

[15] Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. The third PASCAL recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL@ACL 2007 Workshop on Textual Entailment and Paraphrasing, pages 1–9, 2007.

[16] Robert M Gray et al. Toeplitz and circulant matrices: A review. Foundations and Trends® in Communications and Information Theory, 2(3):155–239, 2006.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, 2016.

[18] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. In International Conference on Learning Representations, 2023.

[19] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Annual Conference on Neural Information Processing Systems, 2021.

[20] Feng Hong, Jiangchao Yao, Yifei Shen, Dongsheng Li, Ya Zhang, and Yanfeng Wang. Improving diffusion models for class-imbalanced training data via capacity manipulation. In International Conference on Learning Representations, 2026.

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 2790–2799, 2019.

[22] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022.

[23] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Ka-Wei Lee. Llm-adapters: An adapter family for parameter-efficient fine-tuning of large language models. In Conference on Empirical Methods in Natural Language Processing, pages 5254–5276, 2023.

[24] Cong Hua, Qianqian Xu, Zhiyong Yang, Zitai Wang, Shilong Bao, and Qingming Huang. Openworldauc: Towards unified evaluation and optimization for open-world prompt tuning. In International Conference on Machine Learning, 2025.

[25] Qiushi Huang, Tom Ko, Zhan Zhuang, Lilian Tang, and Yu Zhang. Hira: Parameter-efficient hadamard high-rank adaptation for large language models. In International Conference on Learning Representations, 2025.

[26] Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. Mora: High-rank updating for parameter-efficient fine-tuning. CoRR, 2024.

[27] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Conference on Empirical Methods in Natural Language Processing, pages 3045–3059, 2021.

[28] Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In International Conference on Learning Representations, 2018.

[29] Sicong Li, Qianqian Xu, Zhiyong Yang, Zitai Wang, Linchao Zhang, Xiaochun Cao, and Qingming Huang. Focal-sam: Focal sharpness-aware minimization for long-tailed classification. In International Conference on Machine Learning 2025, 2025.

[30] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In Annual Meeting of the Association for Computational Linguistics, pages 4582–4597, 2021.

[31] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with alphacode. Science, 378(6624):1092–1097, 2022.

[32] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In Annual Conference on Neural Information Processing Systems, 2022.

[33] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. In International Conference on Machine Learning, 2024.

[34] Weiyang Liu, Zeju Qiu, Yao Feng, Yuliang Xiu, Yuxuan Xue, Longhui Yu, Haiwen Feng, Zhen Liu, Juyeon Heo, Songyou Peng, Yandong Wen, Michael J. Black, Adrian Weller, and Bernhard Schölkopf. Parameter-efficient orthogonal finetuning via butterfly factorization. In International Conference on Learning Representations, 2024.

[35] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2019.

[36] Jiaan Luo, Feng Hong, Qiang Hu, Xiaofeng Cao, Feng Liu, and Jiangchao Yao. Long-tailed recognition with model rebalancing. In Advances in Neural Information Processing Systems, 2025.

[37] Jingbiao Mei, Mingsheng Sun, Jinghong Chen, Pengda Qin, Yuhong Li, Da Chen, and Bill Byrne. Expo-hm: Learning to explain-then-detect for hateful meme detection. In International Conference on Learning Representations, 2026.

[38] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. Pissa: Principal singular values and singular vectors adaptation of large language models. In Annual Conference on Neural Information Processing Systems, 2024.

[39] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Conference on Empirical Methods in Natural Language Processing, pages 2381–2391, 2018.

[40] Razvan Pascanu, Tomás Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In International Conference on Machine Learning, 2013.

[41] Yuxin Peng, Zishuo Wang, Geng Li, Xiangtian Zheng, Sibo Yin, and Hulingxiao He. A survey on fine-grained multimodal large language models. Chinese Journal of Electronics, 35(2):771–803, 2026.

[42] Zhen Qin, Xiaodong Han, Weixuan Sun, Bowen He, Dong Li, Dongxu Li, Yuchao Dai, Lingpeng Kong, and Yiran Zhong. Toeplitz neural network for sequence modeling. In International Conference on Learning Representations, 2023.

[43] Zeju Qiu, Weiyang Liu, Haiwen Feng, Yuxuan Xue, Yao Feng, Zhen Liu, Dan Zhang, Adrian Weller, and Bernhard Schölkopf. Controlling text-to-image diffusion by orthogonal finetuning. In Annual Conference on Neural Information Processing Systems, 2023.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In International Conference on Machine Learning, pages 8748–8763, 2021.

[45] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In Conference on Empirical Methods in Natural Language Processing, pages 2383–2392, 2016.

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10674–10685, 2022.

[47] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention, volume 9351, pages 234–241, 2015.

[48] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22500–22510, 2023.

[49] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: an adversarial winograd schema challenge at scale. Commun. ACM, 64(9):99–106, 2021.

[50] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social iqa: Commonsense reasoning about social interactions. In Conference on Empirical Methods in Natural Language Processing, pages 4462–4472, 2019.

[51] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, 2013.

[52] Llama Team. The llama 3 herd of models. CoRR, abs/2407.21783, 2024.

[53] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Kathleen S. Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Agüera y Arcas, Claire Cui, Marian Croak, Ed H. Chi, and Quoc Le. Lamda: Language models for dialog applications. CoRR, abs/2201.08239, 2022.

[54] Ye Tian, Ying Fu, and Jun Zhang. Transformer-based under-sampled single-pixel imaging. Chinese Journal of Electronics, 32(5):1151–1159, 2023.

[55] Otto Toeplitz. Zur theorie der quadratischen und bilinearen formen von unendlichvielen veränderlichen: I. teil: Theorie der l-formen. Mathematische Annalen, 70(3):351–376, 1911.

[56] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. CoRR, 2023.

[57] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In International Conference on Learning Representations, 2019.

[58] Ding Wang, Jin Ren, Haiming Huang, and Junfei Qiao. Particle swarm optimization for adaptive-critic feedback control with power system applications. Chinese Journal of Electronics, 34(4):1265–1274, 2025.

[59] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In International Joint Conference on Artificial Intelligence, pages 4144–4150, 2017.

[60] Zitai Wang, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. Openauc: Towards auc-oriented open-set recognition. In Advances in Neural Information Processing Systems, 2022.

[61] Zitai Wang, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. A unified generalization analysis of re-weighting and logit-adjustment for imbalanced learning. In Advances in Neural Information Processing Systems, 2023.

[62] Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. Trans. Assoc. Comput. Linguistics, 7:625–641, 2019.

[63] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Annual Conference on Neural Information Processing Systems, 2022.

[64] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1112–1122, 2018.

[65] Fei Wu, Jia Hu, Geyong Min, and Shiqiang Wang. Efficient orthogonal fine-tuning with principal subspace adaptation. In International Conference on Learning Representations, 2026.

[66] Zhikang Xu, Qianqian Xu, Zitai Wang, Cong Hua, Sicong Li, Zhiyong Yang, and Qingming Huang. Mind the way you select negative texts: Pursuing the distance consistency in OOD detection with vlms. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026.

[67] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. CoRR, 2025.

[68] Haoming Yang, Ke Ma, Xiaojun Jia, Yingfei Sun, Qianqian Xu, and Qingming Huang. Cannot see the forest for the trees: Invoking heuristics and biases to elicit irrational choices of llms. In International Conference on Machine Learning, 2025.

[69] Haoming Yang, Ke Ma, Yangbangyan Jiang, Ligong Zhang, Xiaojun Jia, Longtao Huang, Yingfei Sun, Qianqian Xu, and Qingming Huang. Narrative paradoxes in llm inference: Shaping reasoning trajectories toward misalignment. In Advances in Neural Information Processing Systems, 2026.

[70] Haoming Yang, Ke Ma, Ligong Zhang, Xiaojun Jia, Yingfei Sun, Qianqian Xu, and Qingming Huang. Hidden dangers of compositional generation: Diagnosing semantic safety failures in text-to-image models. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2026.

[71] Zhiyong Yang, Qianqian Xu, Sicong Li, Zitai Wang, Xiaochun Cao, and Qingming Huang. Dirmixe: Harnessing test agnostic long-tail recognition with hierarchical label vartiations. IEEE Trans. Pattern Anal. Mach. Intell., 2026.

[72] Zhiyong Yang, Qianqian Xu, Zitai Wang, Sicong Li, Boyu Han, Shilong Bao, Xiaochun Cao, and Qingming Huang. Harnessing hierarchical label distribution variations in test agnostic long-tail recognition. In International Conference on Machine Learning, 2024.

[73] Ke Ye and Lek-Heng Lim. Every matrix is a product of toeplitz matrices. Found. Comput. Math., 16(3):577–598, 2016.

[74] Zhaoda Ye, Xiangteng He, and Yuxin Peng. Unsupervised cross-media hashing learning via knowledge graph. Chinese Journal of Electronics, 31(6):1081–1091, 2022.

[75] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In International Conference on Learning Representations, 2024.

[76] Shen Yuan, Haotian Liu, and Hongteng Xu. Bridging the gap between low-rank and orthogonal adaptation via householder reflection adaptation. In Annual Conference on Neural Information Processing Systems, 2024.

[77] Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In Annual Meeting of the Association for Computational Linguistics, pages 1–9, 2022.

[78] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Annual Meeting of the Association for Computational Linguistics, pages 4791–4800, 2019.

[79] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In IEEE Conference on Computer Vision and Pattern Recognition, pages 586–595, 2018.

[80] Tao Zhang, Ying Fu, and Jun Zhang. Deep guided attention network for joint denoising and demosaicing in real image. Chinese Journal of Electronics, 33(1):303–312, 2024.

[81] Tianjiao Zhang, Huangjie Zheng, Jiangchao Yao, Xiangfeng Wang, Mingyuan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed diffusion models with oriented calibration. In International Conference on Learning Representations, 2024.

[82] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. In Findings of the Association for Computational Linguistics, pages 12834–12859, 2024.

[83] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. Int. J. Comput. Vis., 130(9):2337–2348, 2022.

[84] Junda Zhu, Jun Ai, Yujun Li, Yichun Yin, Yasheng Wang, Lifeng Shang, and Qun Liu. Ridgelora: Matrix ridge enhanced low-rank adaptation of large language models. In Annual Conference on Neural Information Processing Systems, 2025.

Appendix

Contents

A Proof **18**

A.1 Proof of Prop. 3.2 18

A.2 Proof of Prop. 4.1 18

A.3 Proof of Thm. 4.2 19

A.4 Proof of Thm. 4.3 21

B Detailed Introduction of OFT and BOFT **23**

C Analysis of Parameter and Computation Cost **24**

C.1 Parameter Cost 24

C.2 Computation Cost 24

D Fast Matrix–Vector Multiplication for Toeplitz Matrices **25**

E More Experiment Protocols **26**

E.1 Datasets 26

E.2 Competitors 29

E.3 Implementation Details 29

F More Experiment Results **30**

F.1 Additional Ablation Study of the Identity Residual 30

F.2 Additional Results on Training Instability in the Naive Method 30

F.3 Multi-Seed Evaluation on Math Reasoning and Code Generation 31

F.4 Additional Qualitative Results for Subject-Driven Generation 31

F.5 Qualitative Results for Math Reasoning 31

G Limitations **36**

H Broader Impacts **37**

A Proof

A.1 Proof of Prop. 3.2

Proposition A.1 (Restate of Prop. 3.2)**.** *Let $\mathcal{L}$ denote the scalar loss, and define the gradient with respect to $\boldsymbol{W}$ as $\boldsymbol{G} := \nabla_{\boldsymbol{W}}\mathcal{L} \in \mathbb{R}^{m\times n}$. Then, for each $i \in \{1,\dots,t\}$, the gradient with respect to $\boldsymbol{T}_i$ is given by:*

$$\nabla_{\boldsymbol{T}_i}\mathcal{L} = \big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\big)^\top \boldsymbol{W}_0^\top \boldsymbol{G}\big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\big)^\top. \tag{11}$$

Proof. Fix an index $i \in \{1,\dots,t\}$. Let $\mathrm{d}\boldsymbol{T}_i$ denote the differential of $\boldsymbol{T}_i$. Then the induced differential of $\boldsymbol{W}$ can be expressed as:

$$\mathrm{d}\boldsymbol{W} = \boldsymbol{W}_0\big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\big)(\mathrm{d}\boldsymbol{T}_i)\big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\big). \tag{12}$$

Accordingly, the differential of the loss function $\mathcal{L}$ is given by:

$$\mathrm{d}\mathcal{L} = \mathrm{tr}\left(\boldsymbol{G}^\top \mathrm{d}\boldsymbol{W}\right) = \mathrm{tr}\left(\boldsymbol{G}^\top \boldsymbol{W}_0\big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\big)(\mathrm{d}\boldsymbol{T}_i)\big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\big)\right). \tag{13}$$

By the cyclic property of the trace, we can rearrange the terms as:

$$\mathrm{d}\mathcal{L} = \mathrm{tr}\left(\Big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\Big)\boldsymbol{G}^\top \boldsymbol{W}_0\Big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\Big)\mathrm{d}\boldsymbol{T}_i\right). \tag{14}$$

Recalling that $\mathrm{d}\mathcal{L} = \mathrm{tr}\left((\nabla\boldsymbol{T}_i\mathcal{L})^\top \mathrm{d}\boldsymbol{T}_i\right)$, we identify the gradient with respect to $\boldsymbol{T}_i$ as

$$\nabla_{\boldsymbol{T}_i}\mathcal{L} = \left(\big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\big)\boldsymbol{G}^\top \boldsymbol{W}_0\big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\big)\mathrm{d}\boldsymbol{T}_i\right)^\top = \big(\prod_{k=1}^{i-1}\boldsymbol{T}_k\big)^\top \boldsymbol{W}_0^\top \boldsymbol{G}\big(\prod_{k=i+1}^{t}\boldsymbol{T}_k\big)^\top. \tag{15}$$

□

A.2 Proof of Prop. 4.1

Proposition A.2 (Restate of Prop. 4.1)**.** *Let $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ satisfy $\mathrm{rank}(\boldsymbol{W}_0) = s$. Let $\{\boldsymbol{T}^i_{\mathrm{blk}}\}_{i=1}^t$ be a sequence of $n \times n$ block-wise Toeplitz matrices and define the update*

$$\Delta\boldsymbol{W} \;=\; \boldsymbol{W}_0\Big(\prod_{i=1}^{t}\boldsymbol{T}^i_{\mathrm{blk}}\Big). \tag{16}$$

Then $\mathrm{rank}(\Delta\boldsymbol{W}) = s$ almost surely, i.e.,

$$\mathbb{P}(\mathrm{rank}(\Delta\boldsymbol{W}) = s) = 1. \tag{17}$$

Proof. We first derive an upper bound on $\mathrm{rank}(\Delta\boldsymbol{W})$ that holds for all admissible choices of $\{\boldsymbol{T}^i_{\mathrm{blk}}\}_{i=1}^t$. We then show that this bound is achieved with probability 1 by noting that the set of singular block-wise Toeplitz matrices has Lebesgue measure zero, and hence the product $\prod_{i=1}^t \boldsymbol{T}^i_{\mathrm{blk}}$ is invertible with probability 1.

Upper bound. For any block-wise Toeplitz matrices $\boldsymbol{T}^i_{\mathrm{blk}}$, we have

$$\Delta\boldsymbol{W} = \boldsymbol{W}_0\Big(\prod_{i=1}^{t}\boldsymbol{T}^i_{\mathrm{blk}}\Big). \tag{18}$$

Since right multiplication cannot increase matrix rank, it follows that

$$\mathrm{rank}(\Delta\boldsymbol{W}) = \mathrm{rank}\left(\boldsymbol{W}_0\prod_{i=1}^{t}\boldsymbol{T}^i_{\mathrm{blk}}\right) \le \min\left(\mathrm{rank}(\boldsymbol{W}_0),\ \mathrm{rank}\Big(\prod_{i=1}^{t}\boldsymbol{T}^i_{\mathrm{blk}}\Big)\right) \le s. \tag{19}$$

Almost-sure equality. Let $\boldsymbol{P} \coloneqq \prod_{i=1}^{t} \boldsymbol{T}_{\mathrm{blk}}^{i} \in \mathbb{R}^{n\times n}$. If $\boldsymbol{P}$ is invertible, then right multiplication by $\boldsymbol{P}$ preserves rank, and hence

$$\operatorname{rank}(\Delta \boldsymbol{W}) = \operatorname{rank}(\boldsymbol{W}_0 \boldsymbol{P}) = \operatorname{rank}(\boldsymbol{W}_0) = s. \tag{20}$$

Therefore, it suffices to show that $\boldsymbol{P}$ is invertible with probability 1.

Each block-wise Toeplitz matrix $\boldsymbol{T}_{\mathrm{blk}}^{i}$ is determined by finitely many parameters. Denote its parameter vector by $\boldsymbol{\theta}_i$, and write $\boldsymbol{T}_{\mathrm{blk}}(\boldsymbol{\theta}_i)$ for the corresponding block-wise Toeplitz matrix. Define

$$f(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t) \;\coloneqq\; \det\Big(\prod_{i=1}^{t} \boldsymbol{T}_{\mathrm{blk}}(\boldsymbol{\theta}_i) \Big). \tag{21}$$

Then f is a polynomial in $(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t)$. Moreover, f is not identically zero: since $\boldsymbol{I}_n$ is itself a block-wise Toeplitz matrix, choosing $\boldsymbol{T}_{\mathrm{blk}}^{i} = \boldsymbol{I}_n$ for all i yields $\boldsymbol{P} = \boldsymbol{I}_n$ and hence $f = 1 \neq 0$. Consequently, the degenerate set

$$\mathcal{D} \;\coloneqq\; \{(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t) : f(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_t) = 0\} \tag{22}$$

is a proper algebraic subset of the block-wise Toeplitz parameter space [13], and in particular has Lebesgue measure 0 [6]. Equivalently, the set of singular block-wise Toeplitz products $\boldsymbol{P}$ has measure 0, and thus $\boldsymbol{P}$ is invertible with probability 1.

Hence, with probability 1, $\boldsymbol{P}$ is invertible, and Eq. (20) implies $\operatorname{rank}(\Delta \boldsymbol{W}) = s$. Combining this with the upper bound in Eq. (19) completes the proof.

□

A.3 Proof of Thm. 4.2

Theorem A.3 (Restatement of Thm. 4.2)**.** *Let $m, p \in \mathbb{N}$ and denote $n := mp$. Consider the $p \times p$ block partition of $\mathbb{C}^{n\times n}$ in which each block has size $m \times m$. Let $\mathrm{Toep}_m(\mathbb{C}) \subset \mathbb{C}^{m\times m}$ denote the linear space of $m \times m$ Toeplitz matrices, and define the set of block-wise Toeplitz matrices as:*

$$\mathcal{TB}_{m,p} := \Big\{ T_{\mathrm{blk}} = [(T_{\mathrm{blk}})_{ij}]_{i,j=1}^{p} \in \mathbb{C}^{n\times n} \;\Big|\; (T_{\mathrm{blk}})_{ij} \in \mathrm{Toep}_m(\mathbb{C}) \text{ for all } i, j \Big\}. \tag{23}$$

For any integer $r \geq 1$, define the polynomial multiplication map:

$$\rho_r^{\mathrm{TB}} : \mathcal{TB}_{m,p}^{r} \to \mathbb{C}^{n\times n}, \qquad \rho_r^{\mathrm{TB}}(T_{\mathrm{blk}}^{(1)}, \ldots, T_{\mathrm{blk}}^{(r)}) = T_{\mathrm{blk}}^{(1)} \cdots T_{\mathrm{blk}}^{(r)}. \tag{24}$$

Let

$$r_0 := \left\lfloor \frac{m}{2} \right\rfloor + 1. \tag{25}$$

Then, the following statements hold.

- *(i) (Generic factorization.) The map $\rho_{r_0}^{\mathrm{TB}}$ is dominant. Equivalently, there exists a Zariski-open dense set $\mathcal{U} \subset \mathbb{C}^{n\times n}$ such that every matrix $A \in \mathcal{U}$ admits a factorization*

$$A = T_{\mathrm{blk}}^{(1)} \cdots T_{\mathrm{blk}}^{(r_0)}, \qquad T_{\mathrm{blk}}^{(i)} \in \mathcal{TB}_{m,p}. \tag{26}$$

- *(ii) (Generic minimality.) If $s < r_0$, then ρ_s^{TB} is not dominant. Consequently, a generic matrix $A \in \mathbb{C}^{n\times n}$ cannot be expressed as a product of s block-wise Toeplitz matrices.*

- *(iii) (Worst-case invertible matrices.) Every invertible matrix $A \in \mathrm{GL}_n(\mathbb{C})$ admits a factorization into a product of at most $2r_0$ block-wise Toeplitz matrices.*

- *(iv) (Worst-case arbitrary matrices.) Every matrix $A \in \mathbb{C}^{n\times n}$ admits a factorization into a product of at most $4r_0 + 1$ block-wise Toeplitz matrices.*

Proof. We structure the proof into multiple parts, corresponding to claims (i) through (iv).

A linear-algebraic model for $\mathcal{TB}_{m,p}$. We begin by establishing a tensor product representation of the space $\mathcal{TB}_{m,p}$. Let $U := \mathbb{C}^{p\times p}$ and $V := \mathrm{Toep}_m(\mathbb{C})$. Consider the map

$$\Phi : U \otimes V \to \mathbb{C}^{n\times n}, \qquad \Phi(A \otimes T) := A \otimes T, \tag{27}$$

where $A \otimes T$ is viewed as a $p \times p$ block matrix whose (i,j)-block is given by $A_{ij}T$. We claim that Φ restricts to a linear isomorphism from $U \otimes V$ onto $\mathcal{TB}_{m,p}$.

To establish surjectivity, consider an arbitrary element $T_{\text{blk}} = [(T_{\text{blk}})_{ij}] \in \mathcal{TB}_{m,p}$. For each (i,j), since block $(T_{\text{blk}})_{ij}$ lies in V, we may expand it relative to a fixed basis $\{B_k\}_{k=1}^{2m-1}$ of V:

$$(T_{\text{blk}})_{ij} = \sum_{k=1}^{2m-1} c_{ij,k} B_k. \tag{28}$$

Letting $\{E_{ij}\}$ denote the standard basis of U, we can rewrite the global matrix as:

$$T_{\text{blk}} = \sum_{i,j=1}^{p} \sum_{k=1}^{2m-1} c_{ij,k} \, (E_{ij} \otimes B_k) \in \text{Im}(\Phi), \tag{29}$$

Thus, $\mathcal{TB}_{m,p} \subseteq \text{Im}(\Phi)$. The reverse inclusion is immediate from the definition of $\mathcal{TB}_{m,p}$, implying $\text{Im}(\Phi) = \mathcal{TB}_{m,p}$. Injectivity follows from the fact that the basis elements $\{E_{ij} \otimes B_k\}_{i,j,k}$ map to block matrices with disjoint supports and linearly independent within each block. Therefore, Φ is an isomorphism, and the dimension of the space is given by:

$$\dim \mathcal{TB}_{m,p} = \dim(U \otimes V) = \dim(U)\dim(V) = p^2(2m-1). \tag{30}$$

Proof of (i): dominance via a full-rank differential. Consider the Toeplitz multiplication map

$$\rho_r^{\text{toep}} : V^r \to \mathbb{C}^{m \times m}, \qquad \rho_r^{\text{toep}}(T_1, \ldots, T_r) = T_1 \cdots T_r. \tag{31}$$

By the results from [73], for $r = r_0$ there exists $\sigma = (T_1, \ldots, T_{r_0}) \in V^{r_0}$ such that the differential $(d\rho_{r_0}^{\text{toep}})_\sigma : V^{r_0} \to \mathbb{C}^{m \times m}$ is surjective.

We lift this configuration to the block setting. Define the factors

$$T_{\text{blk}}^{(i)} := I_p \otimes T_i \in \mathcal{TB}_{m,p}, \qquad \tau := (T_{\text{blk}}^{(1)}, \ldots, T_{\text{blk}}^{(r_0)}) \in \mathcal{TB}_{m,p}^{r_0}. \tag{32}$$

For any perturbation tuple $(\Delta T_{\text{blk}}^{(1)}, \ldots, \Delta T_{\text{blk}}^{(r_0)}) \in \mathcal{TB}_{m,p}^{r_0}$, the product rule yields

$$(d\rho_{r_0}^{\text{TB}})_\tau(\Delta T_{\text{blk}}^{(1)}, \ldots, \Delta T_{\text{blk}}^{(r_0)}) = \sum_{i=1}^{r_0} T_{\text{blk}}^{(1)} \cdots T_{\text{blk}}^{(i-1)} \, \Delta T_{\text{blk}}^{(i)} \, T_{\text{blk}}^{(i+1)} \cdots T_{\text{blk}}^{(r_0)}. \tag{33}$$

Using the isomorphism $\mathcal{TB}_{m,p} \cong U \otimes V$, we decompose the perturbations as:

$$\Delta T_{\text{blk}}^{(i)} = \sum_{\alpha=1}^{p^2} E_\alpha \otimes \Delta T_i^{(\alpha)}, \qquad \Delta T_i^{(\alpha)} \in V, \tag{34}$$

where $\{E_\alpha\}_{\alpha=1}^{p^2}$ is a basis of U. Observing the identity $(I_p \otimes A)(E_\alpha \otimes B)(I_p \otimes C) = E_\alpha \otimes (ABC)$, we substitute $T_{\text{blk}}^{(j)} = I_p \otimes T_j$ into Eq. (33) to obtain the tensorized differential form:

$$(d\rho_{r_0}^{\text{TB}})_\tau(\Delta T_{\text{blk}}^{(1)}, \ldots, \Delta T_{\text{blk}}^{(r_0)}) = \sum_{\alpha=1}^{p^2} E_\alpha \otimes (d\rho_{r_0}^{\text{toep}})_\sigma\big(\Delta T_1^{(\alpha)}, \ldots, \Delta T_{r_0}^{(\alpha)}\big). \tag{35}$$

We now verify the surjectivity of $(d\rho_{r_0}^{\text{TB}})_\tau$. Let $Y \in \mathbb{C}^{n \times n}$ be an arbitrary target matrix. Under the identification $\mathbb{C}^{n \times n} \cong U \otimes \mathbb{C}^{m \times m}$, we may uniquely write $Y = \sum_{\alpha=1}^{p^2} E_\alpha \otimes X_\alpha$, with $X_\alpha \in \mathbb{C}^{m \times m}$. By the surjectivity of $(d\rho_{r_0}^{\text{toep}})_\sigma$, for each index α, there exists a tuple $(\Delta T_1^{(\alpha)}, \ldots, \Delta T_{r_0}^{(\alpha)}) \in V^{r_0}$ satisfying $(d\rho_{r_0}^{\text{toep}})_\sigma(\Delta T_1^{(\alpha)}, \ldots, \Delta T_{r_0}^{(\alpha)}) = X_\alpha$. Constructing the global perturbation $\Delta T_{\text{blk}}^{(i)} := \sum_{\alpha=1}^{p^2} E_\alpha \otimes \Delta T_i^{(\alpha)}$ and substituting it into Eq. (35) yields $(d\rho_{r_0}^{\text{TB}})_\tau(\Delta T_{\text{blk}}^{(1)}, \ldots, \Delta T_{\text{blk}}^{(r_0)}) = Y$. Hence $(d\rho_{r_0}^{\text{TB}})_\tau$ is surjective onto $\mathbb{C}^{n \times n}$.

Since $\mathcal{TB}_{m,p}^{r_0}$ is irreducible and $\rho_{r_0}^{\text{TB}}$ is a polynomial map, the existence of a point τ with surjective differential implies that $\rho_{r_0}^{\text{TB}}$ is dominant [73]. This proves (i).

Proof of (ii): generic minimality by a dimension obstruction. If ρ_s^{TB} were dominant for some s. This requires the dimension of the domain to be at least the dimension of the codomain:

$$\dim(\mathcal{TB}_{m,p}^s) \geq \dim(\mathbb{C}^{n\times n}) = n^2. \tag{36}$$

Substituting Eq. (30) and recalling $n = mp$, we obtain the inequality:

$$s\,p^2(2m-1) \geq m^2p^2 \quad \Longleftrightarrow \quad s \geq \frac{m^2}{2m-1}. \tag{37}$$

Direct calculation shows that $\frac{m^2}{2m-1} > \frac{m}{2}$. Since s must be an integer, the minimal possible value is $s = \left\lceil \frac{m^2}{2m-1} \right\rceil = \lfloor \frac{m}{2} \rfloor + 1 = r_0$. Therefore, no integer $s < r_0$ can yield dominance, proving (ii).

Auxiliary lemma. Let $G := \mathrm{GL}_n(\mathbb{C})$. We recall a standard result for affine algebraic groups.

Lemma A.4. *[73] Let G be an algebraic group and $U, V \subseteq G$ be two Zariski-open and Zariski-dense subsets. Then, $UV = G$.*

Proof of (iii): worst-case factorization over $\mathrm{GL}_n(\mathbb{C})$. From (i), the image of $\rho_{r_0}^{\mathrm{TB}}$ contains a Zariski-open dense subset $\mathcal{U} \subset \mathbb{C}^{n\times n}$. Let $\mathcal{U}_G := \mathcal{U} \cap G$. Since G is open and dense in $\mathbb{C}^{n\times n}$, their intersection $\mathcal{U}_G$ remains Zariski-open dense in G. Applying Lemma A.4 with $U = V = \mathcal{U}_G$ yields $\mathcal{U}_G\mathcal{U}_G = G$. Thus, for any $A \in G$, there exist $X, Y \in \mathcal{U}_G$ such that $A = XY$.

By the definition of $\mathcal{U}$, each of X and Y can be written as a product of r_0 block-wise Toeplitz matrices: $X = \prod_{i=1}^{r_0} X_i$ and $Y = \prod_{i=1}^{r_0} Y_i$ with $X_i, Y_i \in \mathcal{TB}_{m,p}$. Consequently,

$$A = XY = \Big(\prod_{i=1}^{r_0} X_i\Big)\Big(\prod_{i=1}^{r_0} Y_i\Big) \tag{38}$$

is a product of $2r_0$ block-wise Toeplitz matrices, proving (iii).

Proof of (iv): reduction to a rank-normal form. Let $A \in \mathbb{C}^{n\times n}$ be an arbitrary matrix with $\mathrm{rank}(A) = s$. By Gaussian elimination, there exist invertible matrices $P_0, Q_0 \in G$ such that

$$P_0\,A\,Q_0 = \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} =: J_s. \tag{39}$$

We observe that J_s can be realized as a product of a permutation a specific Toeplitz shift. Let $B_k := (\delta_{i+k,j})_{i,j=1}^n$ be the shift matrix with ones on the k-th diagonal. Note that B_k is a Toeplitz matrix. For $k = n - s$, there exists a permutation matrix Π such that:

$$J_s = B_{n-s}\Pi^{-1}. \tag{40}$$

Substituting this back into Eq. (39), we have

$$A = P_0^{-1}J_sQ_0^{-1} = P_0^{-1}B_{n-s}(\Pi^{-1}Q_0^{-1}) =: P\,B\,Q, \tag{41}$$

where $P = P_0^{-1}$ and $Q = \Pi^{-1}Q_0^{-1}$ belong to G, and $B := B_{n-s}$.

A crucial observation is that $B \in \mathcal{TB}_{m,p}$. This holds because B is globally Toeplitz, and the restriction of any Toeplitz matrix to sub-blocks formed by consecutive indices preserves the Toeplitz structure.

Finally, applying result (iii), the invertible matrices P and Q each factor into $2r_0$ block-wise Toeplitz matrices. Thus, the factorization

$$A = P\,B\,Q \tag{42}$$

is a product of $(2r_0) + 1 + (2r_0) = 4r_0 + 1$ block-wise Toeplitz matrices, proving (iv). □

A.4 Proof of Thm. 4.3

Theorem A.5 (Restate of Thm. 4.3)**.** *Let $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$ be a pretrained weight matrix with full column rank, i.e., $\mathrm{rank}(\boldsymbol{W}_0) = n$. Then there exists a class of adapted weight matrices $\boldsymbol{W}$, that can be realized by ToPA with $\mathcal{O}(n)$ trainable parameters, while achieving the same $\boldsymbol{W}$ with LoRA requires at least $\mathcal{O}(n^2)$ parameters and with OFT or BOFT requires at least $\mathcal{O}(n\log n)$ parameters.*

Proof.

1. Construction of the Adaptation via ToPA.

Choose constants $t = \mathcal{O}(1)$ and $p = \mathcal{O}(1)$, both independent of n. For each $k = 1, \ldots, t$, define a block-wise Toeplitz matrix as:

$$\boldsymbol{T}_{\text{blk}}^{(k)} = \begin{bmatrix} \boldsymbol{T}_{11}^{(k)} & \cdots & \boldsymbol{T}_{1p}^{(k)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{T}_{p1}^{(k)} & \cdots & \boldsymbol{T}_{pp}^{(k)} \end{bmatrix}, \tag{43}$$

where each sub-block $\boldsymbol{T}_{ij}^{(k)} \in \mathbb{R}^{\frac{n}{p} \times \frac{n}{p}}$ is a Toeplitz matrix parameterized by a vector $\boldsymbol{w}_{ij}^{(k)} \in \mathbb{R}^{2n/p-1}$. We assume the entries of all parameter vectors are drawn independently from a continuous distribution.

Define the adapter $\boldsymbol{R}$ as:

$$\boldsymbol{R} = \prod_{k=1}^{t} \boldsymbol{T}_{\text{blk}}^{(k)} + \boldsymbol{I}_n, \tag{44}$$

and let the adapted weight:

$$\boldsymbol{W} = \boldsymbol{W}_0 \boldsymbol{R}. \tag{45}$$

The total number of trainable parameters in ToPA for this adaptation is

$$\textbf{\#params}_{\text{(ToPA)}} = t \cdot p^2 \cdot \left(\frac{2n}{p} - 1 \right) = t(2pn - p^2) = \mathcal{O}(n), \tag{46}$$

given that t and p are constants.

2. Properties of the Adaptation.

Rank of the Update Matrix. Let $\boldsymbol{w}^{(k)} \in \mathbb{R}^N$ denote the concatenated vector of all parameters in $\boldsymbol{T}_{\text{blk}}^{(k)}$, where $N = p^2(2n/p - 1)$. The determinant $\det(\boldsymbol{T}_{\text{blk}}^{(k)})$ constitutes a multivariate polynomial $f_k : \mathbb{R}^N \to \mathbb{R}$ with respect to $\boldsymbol{w}^{(k)}$. Since the Toeplitz structure subsumes the identity matrix, there exists a parameter configuration $\boldsymbol{w}_0^{(k)}$ yielding $\boldsymbol{T}_{\text{blk}}^{(k)} = \boldsymbol{I}_n$, which implies $f_k(\boldsymbol{w}_0^{(k)}) = 1$. Consequently, f_k is not identically zero.

Since the zero set of a nonzero polynomial has Lebesgue measure zero [6], it follows that under a continuous parameter distribution,

$$\mathbb{P}\big(\boldsymbol{T}_{\text{blk}}^{(k)} \text{ is singular}\big) = 0. \tag{47}$$

Since t is constant, all $\boldsymbol{T}_{\text{blk}}^{(k)}$ are invertible almost surely, and so is their product $\boldsymbol{T} := \prod_{k=1}^{t} \boldsymbol{T}_{\text{blk}}^{(k)}$. Consider the update matrix $\Delta\boldsymbol{W} = \boldsymbol{W} - \boldsymbol{W}_0 = \boldsymbol{W}_0\boldsymbol{T}$, we have the following equation holds almost surely:

$$\operatorname{rank}(\Delta\boldsymbol{W}) = \operatorname{rank}(\boldsymbol{W}_0\boldsymbol{T}) = \operatorname{rank}(\boldsymbol{W}_0) = s, \tag{48}$$

where the last equality uses the fact that right multiplication by an invertible matrix preserves rank.

Density of $\boldsymbol{R}$. Each entry of $\boldsymbol{R}$ is a polynomial in the parameters. For any fixed (i, j), if this polynomial is not identically zero, its zero set has Lebesgue measure zero, and thus $\mathbb{P}((\boldsymbol{R})_{ij} = 0) = 0$. Because $\boldsymbol{T}$ is invertible and generic, one can verify that for each (i, j) the polynomial is indeed nonzero with probability 1. A union bound over the finitely many entries yields

$$\mathbb{P}\big(\boldsymbol{R} \text{ is dense}\big) = 1. \tag{49}$$

Consequently, the set of parameters yielding a dense and full-rank $\boldsymbol{R}$ almost surely.

Uniqueness of $\boldsymbol{R}$. Because $\boldsymbol{W}_0$ has full column rank, the linear map $\boldsymbol{X} \mapsto \boldsymbol{W}_0\boldsymbol{X}$ is injective on $\mathbb{R}^{n \times n}$. Therefore, for a given adapted weight $\boldsymbol{W}$, there is a unique $\boldsymbol{R}$ satisfying $\boldsymbol{W} = \boldsymbol{W}_0\boldsymbol{R}$. Consequently, any method that produces the same $\boldsymbol{W}$ must effectively represent this specific $\boldsymbol{R}$.

3. Parameter Lower Bounds for LoRA and OFT/BOFT.

LoRA Requires $\mathcal{O}(n^2)$ Parameters. Suppose LoRA realizes the same adaptation, i.e., $\boldsymbol{W} = \boldsymbol{W}_0 + \boldsymbol{B}\boldsymbol{A}$ with $\boldsymbol{B} \in \mathbb{R}^{m \times r}$, $\boldsymbol{A} \in \mathbb{R}^{r \times n}$. Then $\Delta\boldsymbol{W} = \boldsymbol{B}\boldsymbol{A}$, and because $\operatorname{rank}(\Delta\boldsymbol{W}) = s$, we

must have $r \geq s$. The number of trainable parameters is $r(m+n) \geq s(m+n)$. With $s = \Theta(n)$ and $m = \mathcal{O}(n)$, we have:

$$\textbf{\#params}_{(\text{LoRA})} \gtrsim n(m+n) = \mathcal{O}(n^2). \tag{50}$$

OFT/BOFT Require $\Omega(n \log n)$ Parameters. Both OFT and BOFT construct the adapter as a product of sparse matrices $\boldsymbol{B}_i$ with butterfly structures. Since the adapter $\boldsymbol{R}$ is unique, realizing the same adaptation with OFT or BOFT requires

$$\prod_{i=1}^{m} \boldsymbol{B}_i = \boldsymbol{R}. \tag{51}$$

Note that $\boldsymbol{R}$ is a dense matrix.

For OFT, the multiplication chain consists of only one factor, i.e., $m = 1$. Therefore, $\boldsymbol{B}$ itself must be dense, which leads to

$$\textbf{\#params}_{(\text{OFT})} \gtrsim \mathcal{O}(n^2). \tag{52}$$

For BOFT, a fundamental result in butterfly factorization theory states that representing a dense matrix with such a factorization requires at least $\Omega(n \log n)$ parameters [34]. Consequently, to represent the same adaptation with BOFT, the required number of parameters satisfies

$$\textbf{\#params}_{(\text{BOFT})} \gtrsim \mathcal{O}(n \log n). \tag{53}$$

□

B Detailed Introduction of OFT and BOFT

Both OFT and BOFT adapt a pretrained weight matrix through a multiplicative orthogonal update:

$$\boldsymbol{W} = \boldsymbol{W}_0 \boldsymbol{R} = \boldsymbol{W}_0 \Big(\prod_{i=1}^{m} \boldsymbol{B}_i \Big), \qquad \boldsymbol{B}_i \in \mathbb{R}^{n \times n}, \;\; \boldsymbol{B}_i^\top \boldsymbol{B}_i = \boldsymbol{I}. \tag{54}$$

OFT. OFT uses a single ($m = 1$) structured sparse orthogonal matrix with a block-diagonal form:

$$\boldsymbol{R} = \boldsymbol{B}_1 = \operatorname{diag}(\boldsymbol{R}_1, \boldsymbol{R}_2, \ldots, \boldsymbol{R}_r), \qquad \boldsymbol{R}_j \in \mathbb{R}^{b \times b}, \tag{55}$$

where $rb = n$ and each block is constrained to be orthogonal:

$$\boldsymbol{R}_j^\top \boldsymbol{R}_j = \boldsymbol{I}_b. \tag{56}$$

The resulting transformation matrix explicitly manifests as:

$$\boldsymbol{B}_1 = \begin{bmatrix} \boldsymbol{R}_1 & 0 & \cdots & 0 \\ 0 & \boldsymbol{R}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{R}_r \end{bmatrix}. \tag{57}$$

BOFT. BOFT generalizes OFT by parameterizing the orthogonal matrix $\boldsymbol{R}$ as a product of m sparse orthogonal factors:

$$\boldsymbol{R} = \prod_{i=1}^{m} \boldsymbol{B}_i, \tag{58}$$

Here, each $\boldsymbol{B}_i$ is structured as a block-diagonal matrix, exclusively constituted by multiple butterfly factors $\boldsymbol{F}(k_i) \in \mathbb{R}^{k_i \times k_i}$:

$$\boldsymbol{B}_i = \operatorname{diag}\big(\boldsymbol{F}_1(k_i), \ldots, \boldsymbol{F}_{n/k_i}(k_i)\big), \quad k_i = b \cdot 2^i. \tag{59}$$

Specifically, each $\boldsymbol{F}(k_i)$ is defined as a structured sparse matrix populated by learnable $b \times b$ sub-blocks:

$$\boldsymbol{F}(k_i) = \begin{bmatrix} \operatorname{diag}(\boldsymbol{E}_1^{(1)}, \ldots, \boldsymbol{E}_p^{(1)}) & \operatorname{diag}(\boldsymbol{E}_1^{(2)}, \ldots, \boldsymbol{E}_p^{(2)}) \\ \operatorname{diag}(\boldsymbol{E}_1^{(3)}, \ldots, \boldsymbol{E}_p^{(3)}) & \operatorname{diag}(\boldsymbol{E}_1^{(4)}, \ldots, \boldsymbol{E}_p^{(4)}) \end{bmatrix}, \tag{60}$$

where $p = k_i/(2b)$, and $\boldsymbol{E}^{(j)} \in \mathbb{R}^{b \times b}$ represents the learnable block entry.

C Analysis of Parameter and Computation Cost

C.1 Parameter Cost

For ToPA, the fine-tuning process is formulated as

$$\boldsymbol{W} = \boldsymbol{W}_0 \cdot \left(\prod_{i=1}^{t} \boldsymbol{T}_{\text{blk}}^{i} + \boldsymbol{I} \right), \tag{61}$$

where each $\boldsymbol{T}_{\text{blk}}^{i}$ denotes a block-wise Toeplitz matrix in the transformation chain. For a block-wise Toeplitz matrix containing p^2 sub-blocks, the total number of learnable parameters is $p(2n-p)$. Consequently, a chain of length t introduces $tp(2n-p)$ parameters in total.

For comparison, LoRA employs a pair of rank-r matrices, resulting in $r(m+n)$ parameters. In many practical settings, the input and output dimensions are of similar scale, i.e., $m \approx n$. Under this common assumption, when $tp \approx r$, the parameter count of ToPA becomes broadly comparable to that of LoRA, while still enabling richer and more expressive updates to the underlying weight matrix.

C.2 Computation Cost

In the forward computation, we consider the form $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, where $\boldsymbol{x} \in \mathbb{R}^n$ is an input vector and $\boldsymbol{W}$ denotes the adapted weight matrix after fine-tuning. For ease of exposition, we assume that the pretrained weight matrix is square with size $n \times n$. Under ToPA, the adaptation is applied multiplicatively to the pretrained weight matrix $\boldsymbol{W}_0$, and the resulting parameter is expressed as

$$\boldsymbol{W} = \boldsymbol{W}_0 \left(\prod_{k=1}^{t} \boldsymbol{T}_{\text{blk}}^{(k)} + \boldsymbol{I} \right), \tag{62}$$

where each $\boldsymbol{T}_{\text{blk}}^{(k)}$ is a block-wise Toeplitz matrix and $\boldsymbol{I}$ denotes the identity matrix. To analyze the computational cost, we consider the intermediate computation

$$\boldsymbol{z} = \prod_{i=1}^{t} \boldsymbol{T}_{\text{blk}}^{i} \boldsymbol{x}. \tag{63}$$

We partition the input vector into p equal-length sub-vectors, i.e.,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1^\top & \cdots & \boldsymbol{x}_p^\top \end{bmatrix}^\top, \quad \boldsymbol{x}_j \in \mathbb{R}^q, \quad n = qp, \tag{64}$$

and each block-wise Toeplitz matrix is a $p \times p$ block matrix

$$\boldsymbol{T}_{\text{blk}} = \begin{bmatrix} \boldsymbol{T}_{11} & \cdots & \boldsymbol{T}_{1p} \\ \vdots & \ddots & \vdots \\ \boldsymbol{T}_{p1} & \cdots & \boldsymbol{T}_{pp} \end{bmatrix}, \quad \boldsymbol{T}_{ij} \in \mathbb{R}^{q \times q}. \tag{65}$$

where every block $\boldsymbol{T}_{ij}$ is a Toeplitz matrix. The matrix–vector multiplication $\boldsymbol{T}_{\text{blk}}\boldsymbol{x}$ is then carried out in a block-wise way:

$$(\boldsymbol{T}_{\text{blk}}\boldsymbol{x})_i = \sum_{j=1}^{p} \boldsymbol{T}_{ij}\boldsymbol{x}_j, \qquad i = 1, \ldots, p, \tag{66}$$

so that the output vector is likewise obtained as

$$\boldsymbol{T}_{\text{blk}}\boldsymbol{x} = \begin{bmatrix} \sum_{j=1}^{p} \boldsymbol{T}_{1j}\boldsymbol{x}_j \\ \vdots \\ \sum_{j=1}^{p} \boldsymbol{T}_{pj}\boldsymbol{x}_j \end{bmatrix}. \tag{67}$$

Each block–vector product $\boldsymbol{T}_{ij}\boldsymbol{x}_j$ can be computed using FFT-based Toeplitz multiplication with computational cost of $\mathcal{O}(q \log q)$. Since there are p^2 Toeplitz blocks in total, a single block-wise Toeplitz matrix–vector multiplication incurs a cost of

$$\mathcal{O}\left(p^2 q \log q\right) = \mathcal{O}(pn \log q). \tag{68}$$

Table 6: Efficiency comparison on Qwen3-4B for math reasoning. Time denotes the average wall-clock time per training step, and memory denotes the peak GPU memory consumption.

Method	#Param (M)	Time (s)	Memory (MB)
LoRA$_{r=16}$	5.9	1.1	13178
DoRA$_{r=16}$	5.9	1.4	13558
HiRA$_{r=16}$	5.9	1.1	13188
OFT$_{b=32}$	6.1	2.8	16415
BOFT$^{m=2}_{b=16}$	6.1	3.5	19786
ToPA$^{t=3}_{p=5}$	5.5	1.8	15861

Table 7: Efficiency comparison on LLaMA-2-7B for commonsense reasoning. Time denotes the average wall-clock time per training step, and memory denotes the peak GPU memory consumption. BOFT runs out of memory under the same settings without gradient checkpointing.

Method	#Param (M)	Time (s)	Memory (MB)
LoRA$_{r=32}$	16.8	0.9	16494
DoRA$_{r=32}$	17.0	1.3	17070
HiRA$_{r=32}$	16.8	0.9	16083
OFT$_{b=32}$	16.8	3.3	21814
BOFT$^{m=2}_{b=16}$	17.0	–	OOM
ToPA$^{t=2}_{p=16}$	16.7	1.6	18040

In ToPA, this operation is applied sequentially along a chain of length t, yielding a total cost of

$$\mathcal{O}\left(tpn \log q\right) \tag{69}$$

To evaluate the practical computational efficiency of ToPA, we measure both the peak GPU memory consumption and the average wall-clock time per training step. All methods are evaluated under aligned settings on eight RTX 4090 GPUs, with `per_device_batch_size=1` and gradient accumulation enabled. We report results on Qwen3-4B for math reasoning and LLaMA-2-7B for commonsense reasoning. **Notably, on LLaMA-2-7B, BOFT runs out of memory without gradient checkpointing and thus cannot be executed under the same setting.**

As shown in Tab. 6 and Tab. 7, ToPA introduces moderate overhead compared with low-rank methods such as LoRA, DoRA, and HiRA, while remaining substantially more efficient than sparsity structured methods such as OFT and BOFT. Considering the consistent performance gains achieved by ToPA, this overhead is practically acceptable.

D Fast Matrix–Vector Multiplication for Toeplitz Matrices

In this section, we provide the details of the fast matrix–vector multiplication for Toeplitz matrices. The method relies on embedding the Toeplitz matrix into a larger circulant matrix and computing the resulting product via the fast Fourier transform (FFT), see, e.g., [16, 42]. We first recall the classical circulant case, which serves as the algorithmic backbone of our construction.

Circulant matrices and Fourier diagonalization. Let $C \in \mathbb{R}^{n\times n}$ be a circulant matrix whose structure is fully determined by its first column $c = (c_0, \ldots, c_{n-1})^\top$. Each row is obtained by a cyclic shift of c,

$$C = \begin{pmatrix} c_0 & c_{n-1} & \cdots & c_1 \\ c_1 & c_0 & \cdots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ c_{n-1} & c_{n-2} & \cdots & c_0 \end{pmatrix}, \qquad C_{ij} = c_{(i-j) \bmod n}. \tag{70}$$

Let F denote the unitary discrete Fourier transform (DFT) matrix. A standard structural fact is that every circulant matrix is diagonalized by F:

$$C = F^* \Lambda F, \qquad \Lambda = \mathrm{diag}(Fc). \tag{71}$$

Consequently, the matrix–vector product Cx may be written as

$$Cx = F^*(\Lambda(Fx)) = F^*((Fc) \odot (Fx)), \tag{72}$$

where $\odot$ denotes elementwise multiplication. Since F and F^* can be computed using FFT and inverse FFT, the resulting complexity is $\mathcal{O}(n \log n)$, in contrast to the naive $\mathcal{O}(n^2)$ cost.

Embedding a Toeplitz matrix into a circulant matrix. Consider now a Toeplitz matrix $T \in \mathbb{R}^{n \times n}$ whose entries depend only on the diagonal offset:

$$T_{ij} = t_{i-j}, \qquad i, j = 0, \ldots, n-1. \tag{73}$$

Although T is not circulant, it can be embedded into a $(2n) \times (2n)$ circulant matrix $\widetilde{C}$ whose top-left $n \times n$ block coincides with T. We construct the generating column $\tilde{c} \in \mathbb{R}^{2n}$ as

$$\tilde{c}_k = \begin{cases} t_k, & 0 \le k \le n-1, \\ 0, & k = n, \\ t_{k-2n}, & n+1 \le k \le 2n-1, \end{cases} \tag{74}$$

and define $\widetilde{C}$ to be the circulant matrix generated by $\tilde{c}$. Partitioning

$$\widetilde{C} = \begin{pmatrix} T & C_{12} \\ C_{21} & C_{22} \end{pmatrix}, \qquad C_{st} \in \mathbb{R}^{n \times n}, \tag{75}$$

identifies T exactly as the upper-left block of $\widetilde{C}$.

Recovering the Toeplitz product via zero padding. Given $x \in \mathbb{R}^n$, we form the zero-padded vector

$$\tilde{x} = \begin{pmatrix} x \\ 0_n \end{pmatrix} \in \mathbb{R}^{2n}. \tag{76}$$

Using the block structure above, we obtain

$$\widetilde{C}\,\tilde{x} = \begin{pmatrix} Tx \\ C_{21}x \end{pmatrix} \in \mathbb{R}^{2n}, \tag{77}$$

so the desired Toeplitz product is recovered simply by selecting the first n coordinates:

$$Tx = P\,(\widetilde{C}\,\tilde{x}), \qquad P = \begin{bmatrix} I_n & 0_{n \times n} \end{bmatrix}. \tag{78}$$

FFT-based implementation. Since $\widetilde{C}$ is circulant, it admits the $(2n)$-point Fourier diagonalization

$$\widetilde{C} = F_{2n}^*\,\widetilde{\Lambda}\,F_{2n}, \qquad \widetilde{\Lambda} = \mathrm{diag}(F_{2n}\tilde{c}), \tag{79}$$

which yields the practical computation rule

$$Tx = P\,F_{2n}^*\Big((F_{2n}\tilde{c}) \odot (F_{2n}\tilde{x})\Big). \tag{80}$$

Thus, the Toeplitz matrix–vector product can be implemented via fast Fourier transform (FFT), achieving an overall computational complexity of $\mathcal{O}(n \log n)$.

E More Experiment Protocols

E.1 Datasets

Natural Language Understanding. For natural language understanding evaluation, we conduct experiments on eight benchmark datasets from the GLUE [57] benchmark: CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE. These datasets cover a diverse range of sentence-level and sentence-pair tasks, including linguistic acceptability, sentiment classification, paraphrase detection, semantic similarity, and natural language inference. Below is a brief description of these datasets:

- **CoLA** [62]. The Corpus of Linguistic Acceptability is a binary classification dataset for grammatical acceptability judgment. Each instance consists of a single sentence annotated as either acceptable or unacceptable according to linguistic criteria.

- **SST-2** [51]. The Stanford Sentiment Treebank (binary version) is a sentence-level sentiment analysis dataset. The task is to classify whether a movie review sentence expresses positive or negative sentiment.
- **MRPC** [14]. The Microsoft Research Paraphrase Corpus is a paraphrase identification dataset composed of sentence pairs. The goal is to determine whether two sentences are semantically equivalent.
- **STS-B** [7]. The Semantic Textual Similarity Benchmark contains sentence pairs annotated with real-valued similarity scores. The task is formulated as a regression problem measuring the degree of semantic similarity between the two sentences.
- **QQP** [59]. The Quora Question Pairs dataset contains pairs of user-submitted questions. The objective is to detect whether two questions are duplicates that convey the same underlying meaning.
- **MNLI** [64]. The Multi-Genre Natural Language Inference dataset is a large-scale textual entailment benchmark spanning multiple domains. Given a premise and a hypothesis, the task is to classify the relation as entailment, contradiction, or neutral.
- **QNLI** [45]. The Question-answering Natural Language Inference dataset is derived from SQuAD and reformulated as a binary classification task. Given a question and a candidate sentence, the model predicts whether the sentence contains the answer.
- **RTE** [15]. The Recognizing Textual Entailment dataset aggregates several small-scale entailment corpora. Each sample consists of a premise–hypothesis pair labeled as entailment or non-entailment.

Commonsense Reasoning. For commonsense reasoning tasks, we evaluate ToPA's effectiveness on eight widely used benchmark datasets: BoolQ, PIQA, SocialIQA (SIQA), ARC-c, ARC-e, OpenBookQA (OBQA), HellaSwag (HellaS), and WinoGrande (WinoG). These datasets cover diverse forms of commonsense inference, including physical reasoning, social interaction understanding, science question answering, narrative plausibility judgment, and pronoun resolution. Below is a brief description of these datasets:

- **BoolQ** [10]. BoolQ is a yes/no question answering dataset collected from real user queries paired with supporting passages from Wikipedia. The task requires models to determine whether the answer to the question is entailed by the given passage, often involving non-trivial reading comprehension and commonsense inference rather than direct lexical matching.
- **PIQA** [3]. The Physical Interaction Question Answering (PIQA) dataset focuses on physical commonsense reasoning about how people interact with everyday objects. Given a short description of a goal or situation, the model must choose the more plausible of two candidate solutions, which typically requires knowledge of object affordances and intuitive physical constraints.
- **SIQA** [50]. SocialIQA is a benchmark for social commonsense reasoning centered on human intentions, emotions, and social behaviors. Each example presents a short description of a social scenario followed by a multiple-choice question, and the task is to infer the most plausible motivation, reaction, or outcome based on implicit social norms and everyday experiences.
- **ARC-c** [11]. The ARC Challenge (ARC-c) subset consists of grade-school science multiple-choice questions that cannot be solved by simple information-retrieval or co-occurrence baselines. These questions require multi-step reasoning that integrates background science knowledge with commonsense inference, making ARC-c a difficult reasoning-oriented benchmark.
- **ARC-e** [11]. The ARC Easy (ARC-e) subset contains science questions from the same source as ARC-c but with relatively simpler reasoning requirements. Many questions can be answered using direct retrieval or short factual associations, and this subset is commonly used to evaluate performance on more knowledge-oriented science QA tasks.
- **OBQA** [39]. OpenBookQA is an open-book question answering dataset that provides a small collection of elementary science facts together with multiple-choice questions. Solving

each question typically requires combining one core fact from the open book with additional commonsense or world knowledge, encouraging models to perform multi-hop reasoning across explicit and implicit information.

- **HellaS** [78]. HellaSwag is a commonsense inference dataset formulated as a sentence completion task. Given a short context describing an everyday event, the model must select the most plausible continuation among four candidates. The dataset is adversarially constructed so that superficial statistical cues are ineffective, highlighting gaps in models' understanding of everyday scenarios.

- **WinoG** [49]. WinoGrande is a large-scale benchmark inspired by the Winograd Schema Challenge. Each instance consists of a short sentence with an ambiguous reference, and the task is to choose the correct entity based on commonsense reasoning. Compared to earlier Winograd datasets, WinoGrande is adversarially filtered and substantially larger, providing a more robust evaluation of commonsense reasoning abilities.

Math Reasoning & Code Generation. For mathematical reasoning and code generation tasks, we evaluate the effectiveness of our method on four widely used benchmark datasets: GSM8K, MATH, HumanEval, and MBPP. These datasets cover a broad spectrum of problem-solving skills, including grade-school arithmetic reasoning, competition-level mathematics, functional code synthesis, and programming-by-example tasks. Below is a brief description of these datasets:

- **GSM8K** [12]. The GSM8K dataset is a collection of grade-school math word problems that primarily assess multi-step arithmetic and quantitative reasoning. Each question requires reasoning through intermediate steps rather than applying a single direct formula, making it widely used to evaluate chain-of-thought and step-by-step problem-solving ability in language models.

- **MATH** [19]. The MATH dataset consists of competition-style math problems spanning subjects such as algebra, geometry, number theory, and combinatorics. Compared with GSM8K, the problems in MATH are substantially more challenging and often require formal mathematical reasoning, symbolic manipulation, and multi-stage derivations to arrive at correct solutions.

- **HumanEval** [8]. HumanEval is a benchmark for code generation and functional correctness, where each sample provides a natural-language specification and a partially implemented Python function. Models are required to synthesize executable code that satisfies hidden unit tests, and performance is measured using pass@1 metrics that evaluate correctness across multiple samples.

- **MBPP** [2]. The MBPP (Mostly Basic Python Programming) dataset contains short, self-contained Python programming tasks designed for programming-by-example evaluation. Each problem includes a textual description and a few input–output test cases, and the goal is to generate Python code that correctly implements the described functionality across unseen tests.

Subject-Driven Generation. For evaluating subject-driven image generation, we conduct experiments on the dataset introduced in DreamBooth [48]. This dataset is designed to assess a model's ability to personalize a generative model to a specific subject using only a few reference images, while preserving the subject's identity and appearance across novel contexts and backgrounds. It enables systematic evaluation of fidelity, editability, and compositional generalization in personalized image synthesis. Below is a brief description of this dataset:

- **DreamBooth** [48]. The DreamBooth dataset contains a collection of small subject-specific image sets, where each subject is represented by only a few high-quality images captured under different viewpoints and environments. During training, the model learns to bind a unique identifier token to the target subject, enabling personalized generation conditioned on text prompts. The dataset is used to evaluate whether the generated images maintain identity consistency while allowing the subject to be rendered in diverse scenes, artistic styles, and compositional settings.

E.2 Competitors

To evaluate the effectiveness of our method, we select several representative methods as competitors. Below is a brief description of the competitors considered in our experiments:

- **LoRA** [22]. LoRA (Low-Rank Adaptation) factorizes the weight update into the product of two low-rank matrices that are learned during fine-tuning, while the original pretrained weights remain frozen. This design substantially reduces the number of trainable parameters and allows the updates to be merged back into the base model at inference time.
- **DoRA** [33]. DoRA decomposes each weight matrix into a magnitude component and a direction component. It directly tunes the magnitude component while employing LoRA to the direction component, thereby increasing the adapter's learning capacity and optimization stability.
- **PiSSA** [38]. PiSSA performs the SVD decomposition to split each pretrained weight matrix into principal and residual components, initializing LoRA with the principal component while assigning the residual component to the pretrained weight, thereby accelerating the convergence.
- **HiRA** [25]. HiRA factorizes the update matrix as the Hadamard product between the pretrained weight matrix and a low-rank matrix, thereby yielding a high-rank and more expressive adaptation.
- **RidgeLoRA** [84]. RidgeLoRA reformulates the LoRA update by replacing the parallel additive residual branch with a series, multiplicative transformation. In addition, it enriches the low-rank adapter with a learnable diagonal ridge matrix, achieving a high-rank update and further strengthening the expressiveness.
- **OFT** [43]. OFT adapts the model weights by multiplying the pretrained weights with a carefully designed sparse orthogonal matrix, which enables lightweight transformation while preserving the hyperspherical energy of the network and maintaining the pretrained semantic knowledge of the model.
- **BOFT** [34]. BOFT extends OFT by replacing the single sparse orthogonal matrix with a product of multiple sparse butterfly matrices. This structured parameterization improves the expressiveness of the transformation while preserving parameter efficiency.
- **HRA** [76]. HRA fine-tunes each pretrained matrix by right-multiplying it with an orthogonal matrix parameterized as a product of learnable Householder reflections. By regularizing the mutual orthogonality of the reflection planes, HRA balances adaptation capacity and regularity, thereby preserving pretrained knowledge while achieving strong performance.

E.3 Implementation Details

Natural Language Understanding. Following [34, 76], we adopt DeBERTaV3-Base [18] as the backbone model. We use the same batch size of 32 and maximum sequence length as in the prior work, while tuning the learning rate and the number of training epochs. We employ the AdamW [35] optimizer for training. During fine-tuning, we freeze the pretrained backbone parameters and insert adapters into the query, key, and value projection matrices in the attention modules. Detailed hyperparameter settings are reported in Tab. 8.

Table 8: The hyperparameter settings on the GLUE benchmarks

Dataset	MNLI	SST-2	CoLA	QQP	QNLI	RTE	MRPC	STS-B
# Epochs	8	15	30	15	5	40	40	25
Learning Rate	3e-4	1e-4	4e-4	1e-4	1e-4	2e-5	3e-4	8e-5
Max Seq. Len.	256	128	64	320	512	320	320	128

Subject-Driven Generation. Following [43], we adopt Stable Diffusion v1.5 [46] as the base model and inject the adapter into the query, key, value, and output projection matrices of the U-Net [47] module. We use the AdamW optimizer with a learning rate of 6×10^{-5} and a batch size of 1, training the model for approximately 1,000 steps.

Commonsense Reasoning. For the commonsense reasoning tasks, we employ the LLaMA-2-7B [56] and LLaMA-3-8B [52] as the backbone models. During fine-tuning, all pretrained weights are kept frozen, and adapters are inserted into the query and value projection matrices. Optimization is performed with AdamW optimizer using a learning rate of 3×10^{-5}, a warm up steps of 100, and a batch size of 16. The training extends for 3 epochs.

Math Reasoning & Code Generation. For math reasoning and code generation tasks, we use LLaMA-3-8B [52] and Qwen3-4B [67] as the base models. During training, adapters are applied to fine-tune the query and value projection matrices within the attention modules. Optimization is carried out using AdamW with a learning rate of 1×10^{-4} and a batch size of 32. We further employ a cosine learning rate scheduler, and the models are trained for a single epoch.

F More Experiment Results

F.1 Additional Ablation Study of the Identity Residual

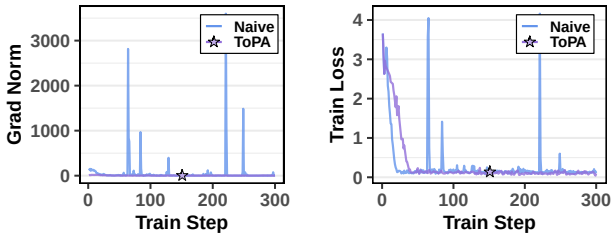
To further demonstrate the crucial role of the identity residual, we perform an additional ablation analysis. As detailed in Tab. 9, we compare the performance impact of the identity matrix when fine-tuning LLaMA-3-8B with ToPA$_{p=8}^{t=3}$ on math reasoning and code generation tasks. The result suggests that incorporating the identity residual yields consistent performance gains across all evaluated datasets. These results further confirm the effectiveness of the residual design.

Table 9: Performance comparison of LLaMA-3-8B fine-tuned with ToPA$_{p=8}^{t=3}$ on math & code generation tasks, with and without the identity residual.

Method	# Param	GSM8K	MATH	HumanEval	Mbpp
w/o Identity	12.6M	74.3	23.2	51.0	66.6
w/ Identity	12.6M	**74.7**	**23.5**	**51.8**	**67.2**

F.2 Additional Results on Training Instability in the Naive Method

Fig. 8 illustrates the training dynamics of the Frobenius norm of gradients and the training loss for the naive method ($t = 32$) and ToPA$_{p=16}^{t=2}$ when fine-tuning LLaMA-2-7B on commonsense reasoning tasks. As shown in the figures, the naive method suffers from pronounced instability: both the gradient norm and the training loss exhibit substantial fluctuations and attain relatively large values, indicating a tendency toward gradient explosion. In contrast, ToPA yields significantly more stable behavior, with both metrics remaining consistently smoother and markedly smaller than those of the naive baseline. These observations further corroborate the effectiveness of ToPA in mitigating gradient explosion, thereby leading to more stable training dynamics.



(a) Gradient Norm (b) Train Loss

Figure 8: Training dynamics over the first 300 steps of fine-tuning LLaMA-2-7B on commonsense reasoning tasks using the naive method ($t = 32$) and ToPA$_{p=16}^{t=2}$ with matched parameters. As training proceeds, the naive method collapses, so only the first 300 steps are shown.

F.3 Multi-Seed Evaluation on Math Reasoning and Code Generation

To further verify the robustness of ToPA, we repeat the math reasoning and code generation experiments on the LLaMA-3-8B backbone with three random seeds. As shown in Tab. 10, ToPA consistently achieves the best or tied-best average performance under a comparable parameter budget. These results further validate the effectiveness of ToPA.

Table 10: Multi-seed performance comparison on math reasoning and code generation tasks with the LLaMA-3-8B backbone. Results are averaged over three random seeds.

Method	# Param	GSM8K	MATH	HumanEval	MBPP
LoRA$_{r=32}$	13.6M	70.5±0.4	22.5±0.3	50.2±0.3	**67.2**±0.3
DoRA$_{r=32}$	13.8M	71.1±0.3	22.8±0.2	51.0±0.3	67.0±0.2
OFT$_{b=64}$	17.0M	73.4±0.3	22.9±0.2	45.7±0.5	66.5±0.2
BOFT$^{m=3}_{b=16}$	12.8M	69.4±0.4	22.3±0.3	45.5±0.3	67.1±0.2
ToPA$^{t=3}_{p=8}$	12.6M	**74.9**±0.3	**23.5**±0.2	**51.6**±0.3	**67.2**±0.3

F.4 Additional Qualitative Results for Subject-Driven Generation

Fig. 9, Fig. 10, and Fig. 11 demonstrate additional qualitative results for subject-driven generation tasks. From the results, we can observe that ToPA consistently produces images with higher subject fidelity and stronger instruction-following ability compared with LoRA and OFT. Overall, these qualitative comparisons further support the effectiveness of ToPA for subject-driven generation, demonstrating its ability to balance subject preservation with generation diversity.

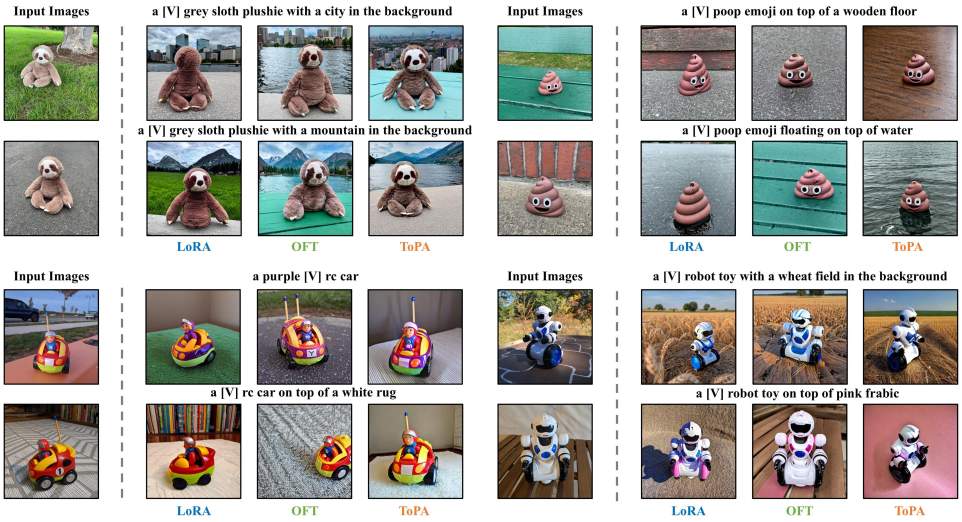


Figure 9: Additional qualitative comparison of results on subject-driven generation tasks.

F.5 Qualitative Results for Math Reasoning

In this section, we present several qualitative examples comparing the questions and corresponding responses generated by the LLaMA-3-8B model fine-tuned with LoRA and ToPA on math reasoning tasks. From these cases, we observe that ToPA consistently produces more accurate and reliable answers than LoRA. These qualitative results further support the effectiveness of ToPA in enhancing the model's math reasoning capability.

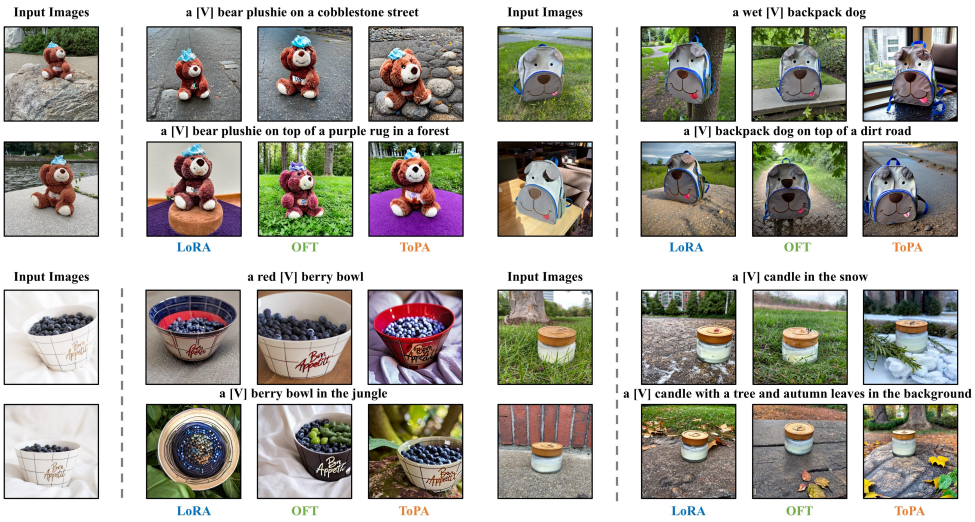


Figure 10: Additional qualitative comparison of results on subject-driven generation tasks.

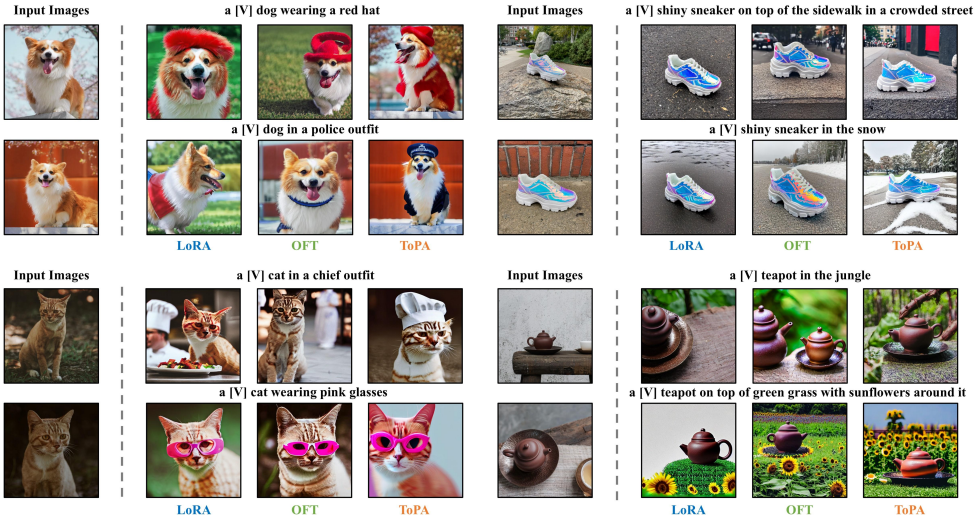


Figure 11: Additional qualitative comparison of results on subject-driven generation tasks.

Case Study

Question: I have 10 liters of orange drink that are two-thirds water and I wish to add it to 15 liters of pineapple drink that is three-fifths water. But as I pour it, I spill one liter of the orange drink. How much water is in the remaining 24 liters?

Response of LoRA$_{r=32}$: First, we need to find out how much water is in the orange drink. Two-thirds of 10 liters is 10 * 2/3 = 6.67 liters. So, there are 6.67 liters of water in the orange drink. Next, we need to find out how much water is in the pineapple drink. Three-fifths of 15 liters is 15 * 3/5 = 9 liters. So, there are 9 liters of water in the pineapple drink. Now, we need to add the water from the orange drink to the pineapple drink. The total amount of water in the pineapple drink is 9 + 6.67 = 15.67 liters. Since we spilled 1 liter of the orange drink, the remaining amount of water in the pineapple drink is 15.67 - 1 = 14.67 liters. Therefore, there are 14.67 liters of water in the remaining 24 liters. #### 14.67 The answer is: 14.67

Response of ToPA$_{p=8}^{t=3}$: We have 10 liters of orange drink that is two-thirds water, so 10 * (2/3) = 6.67 liters of water in the orange drink. We also have 15 liters of pineapple drink that is three-fifths water, so 15 * (3/5) = 9 liters of water in the pineapple drink. When we pour the orange drink, we spill 1 liter, so we have 10 - 1 = 9 liters of orange drink remaining. The remaining orange drink is two-thirds water, so 9 * (2/3) = 6 liters of water in the remaining orange drink. The total amount of water in the remaining 24 liters is 6 + 9 = 15 liters. #### 15 The answer is: 15

Correct Answer: 15

Case Study

Question: Meredith is a freelance blogger who writes about health topics and submits to clients each day as her permanent job. A blog article takes an average of 4 hours to research and write about. Last week, she wrote 5 articles on Monday and 2/5 times more articles on Tuesday than on Monday. On Wednesday, she wrote twice the number of articles she wrote on Tuesday. Calculate the total number of hours she spent writing articles in the three days.

Response of LoRA$_{r=32}$: On Monday, Meredith wrote 5 articles, so she spent 5 * 4 = 20 hours writing articles. On Tuesday, she wrote 2/5 * 5 = 2 * 5 = 10 articles. So, she spent 10 * 4 = 40 hours writing articles on Tuesday. On Wednesday, she wrote twice the number of articles she wrote on Tuesday, so she wrote 10 * 2 = 20 articles. Therefore, she spent 20 * 4 = 80 hours writing articles on Wednesday. In total, she spent 20 + 40 + 80 = 140 hours writing articles in the three days. #### 140 The answer is: 140

Response of ToPA$_{p=8}^{t=3}$: On Monday, Meredith wrote 5 articles, so she spent 5 * 4 = 20 hours writing articles. On Tuesday, she wrote 2/5 * 5 = 2 more articles than on Monday, so she wrote 5 + 2 = 7 articles. Therefore, she spent 7 * 4 = 28 hours writing articles on Tuesday. On Wednesday, she wrote twice the number of articles she wrote on Tuesday, so she wrote 7 * 2 = 14 articles. Therefore, she spent 14 * 4 = 56 hours writing articles on Wednesday. In total, Meredith spent 20 + 28 + 56 = 104 hours writing articles in the three days. #### 104 The answer is: 104

Correct Answer: 104

Case Study

Question: John plans to sell all his toys and use the money to buy video games. He has 13 lego sets and he sells them for $15 each. He ends up buying 8 video games for $20 each and has $5 left. How many lego sets does he still have?

Response of LoRA$_{r=32}$: John sells 13 lego sets for $15 each, so he earns 13 * $15 = $195. He buys 8 video games for $20 each, so he spends 8 * $20 = $160. He has $5 left, so he has $195 - $160 = $35 from selling the lego sets. Therefore, John still has 13 - 8 = 5 lego sets. #### 5 The answer is: 5

Response of ToPA$_{p=8}^{t=3}$: John sells 13 lego sets for $15 each, so he earns 13 * $15 = $195. He then buys 8 video games for $20 each, so he spends 8 * $20 = $160. He has $5 left, so he has $195 - $160 - $5 = $30 from selling the lego sets. Since each lego set is worth $15, he still has $30 / $15 = 2 lego sets left. #### 2 The answer is: 2

Correct Answer: 2

Case Study

Question: Gloria is shoe shopping when she comes across a pair of boots that fit her shoe budget. However, she has to choose between the boots and two pairs of high heels that together cost five dollars less than the boots. If one pair of heels costs $33 and the other costs twice as much, how many dollars are the boots?

Response of LoRA$_{r=32}$: One pair of heels costs $33. The other pair of heels costs twice as much, so it costs 2 * $33 = $66. The total cost of the two pairs of heels is $33 + $66 = $99. The boots cost five dollars less than the two pairs of heels, so they cost $99 - $5 = $94. Therefore, the boots cost $94. #### 94 The answer is: 94

Response of ToPA$_{p=8}^{t=3}$: One pair of heels costs $33, and the other costs twice as much, so the second pair of heels costs 2 * $33 = $66. The total cost of the two pairs of heels is $33 + $66 = $99. The boots cost five dollars more than the two pairs of heels, so the boots cost $99 + $5 = $104. Therefore, the boots cost $104. #### 104 The answer is: 104

Correct Answer: 104

Case Study

Question: A translation of the plane takes $5 + 5i$ to $6 + 8i$. Find the complex number that the translation takes -6 to.

Response of LoRA$_{r=32}$: We are given that the translation takes $5 + 5i$ to $6 + 8i$. To find the complex number that the translation takes -6 to, we need to find the complex number that is -6 units away from $5 + 5i$ in the direction of $6 + 8i$. We can use the formula for the distance between two complex numbers to find the distance between $5 + 5i$ and -6. The distance between $5 + 5i$ and -6 is $\sqrt{(-6-5)^2 + (-6-5i)^2} = \sqrt{121 + 121} = \sqrt{242} = 14$. Now, we need to find the complex number that is 14 units away from $5 + 5i$ in the direction of $6 + 8i$. We can use the formula for the direction of a complex number to find the direction of $6 + 8i$. The direction of $6 + 8i$ is $\tan^{-1}\left(\frac{8}{6}\right) = \tan^{-1}\left(\frac{4}{3}\right)$. Now, we can use the formula for the complex number that is 14 units away from $5 + 5i$ in the direction of $6 + 8i$. The complex number is $5 + 5i + 14\left(\cos\left(\tan^{-1}\left(\frac{4}{3}\right)\right) + i\sin\left(\tan^{-1}\left(\frac{4}{3}\right)\right)\right)$. Simplifying, we get $5 + 5i + 14\left(\frac{4}{5} + i\frac{3}{5}\right) = 5 + 5i + 56\left(\frac{4}{5} + i\frac{3}{5}\right)$. Further simplifying, we get $5+5i+224\left(\frac{4}{5} + i\frac{3}{5}\right) = 5+5i+896\left(\frac{4}{5} + i\frac{3}{5}\right)$. Finally, we get $5+5i+896\left(\frac{4}{5} + i\frac{3}{5}\right) = 5 + 5i + 3584\left(\frac{4}{5} + i\frac{3}{5}\right)$. Therefore, the complex number that the translation takes -6 to is $\boxed{3584\left(\frac{4}{5} + i\frac{3}{5}\right)}$.The answer is: $3584\left(\frac{4}{5} + i\frac{3}{5}\right)$

Response of ToPA$_{p=8}^{t=3}$: The translation takes $5+5i$ to $6+8i$. This means that the translation adds 1 to the real part and 3 to the imaginary part. So, if we want to find the complex number that the translation takes -6 to, we need to add 1 to the real part and 3 to the imaginary part. Therefore, the complex number that the translation takes -6 to is $-6+1+3i = \boxed{-5+3i}$. The answer is: $-5 + 3i$

Correct Answer: $-5 + 3i$

Case Study

Question: Compute

$$\begin{pmatrix} 3 & 1 & 0 \\ -7 & 4 & 2 \\ 0 & 5 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \end{pmatrix}.$$

Response of LoRA$_{r=32}$: We want to multiply the matrix by the vector. First, we multiply the first row of the matrix by the first element of the vector, the second row by the second element, and the third row by the third element. So, we have $3 \cdot 4 + 1 \cdot (-1) + 0 \cdot (-2) = 12 - 1 + 0 = 11$. Next, we multiply the first row of the matrix by the second element of the vector, the second row by the third element, and the third row by the first element. So, we have $-7 \cdot 4 + 4 \cdot (-1) + 0 \cdot (-2) = -28 + 4 + 0 = -24$. Finally, we multiply the first row of the matrix by the third element of the vector, the second row by the first element, and the third row by the second element. So, we have $0 \cdot 4 + 5 \cdot (-1) + (-1) \cdot (-2) = 0 - 5 + 2 = -3$. Therefore, the product of the matrix and the vector is $\begin{pmatrix} 11 \\ -24 \\ -3 \end{pmatrix}$. The answer is: $\begin{pmatrix} 11 \\ -24 \\ -3 \end{pmatrix}$

Response of ToPA$_{p=8}^{t=3}$: We have a 3×3 matrix and a 3×1 matrix. To multiply these matrices, we need to multiply each element of the first matrix by the corresponding element of the second matrix and then sum the products. So, we have

$$\begin{pmatrix} 3 & 1 & 0 \\ -7 & 4 & 2 \\ 0 & 5 & -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \end{pmatrix} = \begin{pmatrix} 3 \cdot 4 + 1 \cdot (-1) + 0 \cdot (-2) \\ -7 \cdot 4 + 4 \cdot (-1) + 2 \cdot (-2) \\ 0 \cdot 4 + 5 \cdot (-1) + (-1) \cdot (-2) \end{pmatrix} = \begin{pmatrix} 12 - 1 - 0 \\ -28 - 4 - 4 \\ 0 - 5 + 2 \end{pmatrix} = \begin{pmatrix} 11 \\ -36 \\ -3 \end{pmatrix}.$$

So, the product is $\boxed{\begin{pmatrix} 11 \\ -36 \\ -3 \end{pmatrix}}$.The answer is: $\begin{pmatrix} 11 \\ -36 \\ -3 \end{pmatrix}$

Correct Answer: $\begin{pmatrix} 11 \\ -36 \\ -3 \end{pmatrix}$

Case Study

Question: Compute $\arccos\left(-\frac{1}{2}\right)$. Express your answer in radians.

Response of LoRA$_{r=32}$: First, we need to find the angle whose cosine is $-\frac{1}{2}$. We know that the cosine of an angle is negative when the angle is in the second or fourth quadrant. In the second quadrant, the cosine of an angle is negative when the angle is between $\frac{2\pi}{3}$ and π. So, we need to find the angle between $\frac{2\pi}{3}$ and π whose cosine is $-\frac{1}{2}$. We can use a calculator to find that the angle is $\boxed{\frac{5\pi}{6}}$. The answer is: $\frac{5\pi}{6}$

Response of ToPA$_{p=8}^{t=3}$: The cosine function is positive in the first and fourth quadrants, so the angle θ that satisfies $\cos\theta = -\frac{1}{2}$ must be in the second or third quadrant. In the second quadrant, the reference angle is $\theta = \frac{2\pi}{3}$, and in the third quadrant, the reference angle is $\theta = \frac{4\pi}{3}$. Since $\arccos$ gives the angle in the principal branch, we take the angle in the second quadrant, so $\arccos\left(-\frac{1}{2}\right) = \boxed{\frac{2\pi}{3}}$. The answer is: $\frac{2\pi}{3}$

Correct Answer: $\frac{2\pi}{3}$

Case Study

Question: Given $\|\mathbf{v}\| = 5$ and $\|\mathbf{w}\| = 8$, find the largest possible value of

$$\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|.$$

Response of LoRA$_{r=32}$: First, we need to find the projection of $\mathbf{v}$ onto $\mathbf{w}$. The projection of $\mathbf{v}$ onto $\mathbf{w}$ is given by

$$\operatorname{proj}_{\mathbf{w}} \mathbf{v} = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|^2} \mathbf{w}.$$

We know that $\|\mathbf{v}\| = 5$ and $\|\mathbf{w}\| = 8$, so we can substitute these values into the formula:

$$\operatorname{proj}_{\mathbf{w}} \mathbf{v} = \frac{\mathbf{v} \cdot \mathbf{w}}{8^2} \mathbf{w}.$$

Now, we need to find the largest possible value of $\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|$. To do this, we need to maximize the magnitude of the projection, which is given by

$$\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{8^2} \mathbf{w} \right\|.$$

Since $\mathbf{w}$ is a unit vector, we can simplify this expression to

$$\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{8^2} \mathbf{w} \right\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{64} \mathbf{w} \right\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{64} \mathbf{w} \right\|.$$

To maximize this expression, we need to maximize the magnitude of $\mathbf{v} \cdot \mathbf{w}$. Since $\mathbf{v}$ and $\mathbf{w}$ are vectors, we can think of $\mathbf{v} \cdot \mathbf{w}$ as the dot product of $\mathbf{v}$ and $\mathbf{w}$. The dot product of two vectors is equal to the product of their magnitudes times the cosine of the angle between them. So, we want to maximize the magnitude of $\mathbf{v} \cdot \mathbf{w}$, which is equal to the product of the magnitudes of $\mathbf{v}$ and $\mathbf{w}$ times the cosine of the angle between them. Since $\|\mathbf{v}\| = 5$ and $\|\mathbf{w}\| = 8$, the magnitude of $\mathbf{v} \cdot \mathbf{w}$ is equal to $5 \cdot 8 \cdot \cos\theta$, where θ is the angle between $\mathbf{v}$ and $\mathbf{w}$. To maximize this expression, we want to maximize the value of $\cos\theta$. The maximum value of $\cos\theta$ is 1, which occurs when $\theta = 0$. So, the largest possible value of $\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|$ is $\|\mathbf{v} \cdot \mathbf{w}\| = 5 \cdot 8 \cdot \cos 0 = 40$. Therefore, the largest possible value of $\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|$ is $\boxed{40}$.The answer is: 40

Response of ToPA$_{p=8}^{t=3}$: The projection of $\mathbf{v}$ onto $\mathbf{w}$ is given by

$$\operatorname{proj}_{\mathbf{w}} \mathbf{v} = \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|^2} \mathbf{w}.$$

We want to find the largest possible value of $\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|$. Since $\|\mathbf{v}\| = 5$ and $\|\mathbf{w}\| = 8$, we have

$$\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{\|\mathbf{w}\|^2} \mathbf{w} \right\| = \left\| \frac{\mathbf{v} \cdot \mathbf{w}}{64} \mathbf{w} \right\| = \frac{|\mathbf{v} \cdot \mathbf{w}|}{64} \|\mathbf{w}\| = \frac{|\mathbf{v} \cdot \mathbf{w}|}{64} \cdot 8 = \frac{|\mathbf{v} \cdot \mathbf{w}|}{8}.$$

To maximize this expression, we want to maximize $|\mathbf{v} \cdot \mathbf{w}|$. Since $\|\mathbf{v}\| = 5$ and $\|\mathbf{w}\| = 8$, the largest possible value of $|\mathbf{v} \cdot \mathbf{w}|$ occurs when $\mathbf{v}$ and $\mathbf{w}$ are parallel. In this case, $\mathbf{v} \cdot \mathbf{w} = \|\mathbf{v}\| \cdot \|\mathbf{w}\| = 5 \cdot 8 = 40$. Therefore, the largest possible value of $\|\operatorname{proj}_{\mathbf{w}} \mathbf{v}\|$ is $\frac{40}{8} = \boxed{5}$. The answer is: 5

Correct Answer: 5

G Limitations

Although ToPA shows consistent effectiveness across diverse NLP and CV tasks, our current evaluation is still mainly focused on single-modal backbone models. Extending ToPA to multimodal large models, such as vision-language models, remains an important direction for future work, where cross-modal interactions may introduce new adaptation requirements.

In addition, due to computational resource constraints, our experiments are conducted on models up to the 8B scale. While the results demonstrate the effectiveness of ToPA on multiple architectures, further investigation on larger-scale models would be valuable to better understand its scalability and practical behavior in more demanding settings.

H Broader Impacts

This work aims to improve the efficiency and effectiveness of parameter-efficient fine-tuning. Its positive impact lies in reducing the computational and memory cost of adapting large pretrained models, which may make fine-tuning more accessible to researchers and practitioners with limited resources. It may also help reduce the environmental cost associated with repeatedly adapting large models.

This work is primarily methodological and does not target a specific high-risk application. We do not identify direct negative societal impacts beyond those generally associated with improving the adaptability of large models. In practice, downstream deployment should still consider task-specific risks, such as misuse, bias, or unreliable generations, depending on the application domain.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction accurately reflect the proposed method, theoretical analysis, and empirical results presented in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated Limitations section discussing the current evaluation scope and future extensions to multimodal and larger-scale models.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The theoretical results are stated with the necessary assumptions and are supported by formal proofs in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of the model architectures, datasets, baselines, hyperparameter settings, and evaluation protocols used in the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: If the paper is accepted, we will release the complete codebase along with detailed reproduction instructions.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies the main datasets, model backbones, baselines, evaluation protocols, and key training details. Additional hyperparameters and implementation settings are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The appendix includes a subsection reporting results over multiple random seeds. The paper reports the mean and standard deviation to reflect the stability and statistical variation of the main experimental results.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper reports the compute resources used for the experiments, including the GPU type, memory usage, and wall-clock training time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a Broader Impacts section discussing the potential benefits of more efficient model adaptation and the general downstream risks associated with adapting large models.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new high-risk pretrained models, image generators, or scraped datasets. Therefore, this item is not applicable to the proposed methodological work.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly cites the existing datasets, models, and baselines used in the experiments. We use publicly available assets and follow their corresponding licenses and terms of use.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release new datasets, model, or other standalone assets, so this item is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, human-subject studies, or user experiments, so this item is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve human-subject studies, crowdsourcing, or user experiments, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology does not use LLMs as an important, original, or non-standard research component.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.