MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

SuperLoRA: Parameter-Efficient Unified Adaptation of Large Foundation Models

Chen, Xiangyu; Liu, Jing; Wang, Ye; Wang, Pu; Brand, Matthew; Wang, Guanghui; Koike-Akino, Toshiaki

TR2024-156 November 15, 2024

Abstract

Low-rank adaptation (LoRA) and its variants are widely employed in fine-tuning large models, including large language models for natural language processing and dif- fusion models for computer vision. This paper proposes a generalized framework called SuperLoRA that unifies and extends different LoRA variants, which can be realized un- der different hyper-parameter settings. Introducing new options with grouping, folding, shuffling, projection, and tensor decomposition, SuperLoRA offers high flexibility and demonstrates superior performance, with up to a 10-fold gain in parameter efficiency for transfer learning tasks.

British Machine Vision Conference (BMVC) 2024

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

SuperLoRA: Parameter-Efficient Unified Adaptation of Large Foundation Models

Xiangyu Chen[1,2]
xiachen@merl.com
Jing Liu[2]
jiliu@merl.com
Ye Wang[2]
yewang@merl.com
Pu (Perry) Wang[2]
pwang@merl.com
Matthew Brand[2]
brand@merl.com
Guanghui Wang[3]
koike@merl.com
Toshiaki Koike-Akino[2]
wangcs@torontomu.ca

[1] University of Kansas
Lawrence, KS 66045, USA
[2] Mitsubishi Electric Research Laboratories (MERL)
Cambridge, MA 02139, USA
[3] Toronto Metropolitan University
Toronto, ON M5B 2K3, Canada

Abstract

Low-rank adaptation (LoRA) and its variants are widely employed in fine-tuning large models, including large language models for natural language processing and diffusion models for computer vision. This paper proposes a generalized framework called SuperLoRA that unifies and extends different LoRA variants, which can be realized under different hyper-parameter settings. Introducing new options with grouping, folding, shuffling, projection, and tensor decomposition, SuperLoRA offers high flexibility and demonstrates superior performance, with up to a 10-fold gain in parameter efficiency for transfer learning tasks.

1 Introduction

Large neural network models are dominating machine learning recently with the emergence of exceptional models, such as large vision models (LVMs) including Vision Transformer (ViT) [10], ConvNeXt [33] and Stable Diffusion [19] for vision tasks, and large language models (LLMs) including GPT [1], PALM2 [4], Gemini [3] and LLaMA2 [39] for natural language processing (NLP). However, the increased resource consumption and data requirement along with model size limits its generalization on downstream tasks. To solve this, Parameter-Efficient Fine-Tuning (PEFT) has been widely explored to fine-tune less parameters while retaining high performance. Among this, adapter-based techniques like LoRA (**Lo**w-**R**ank **A**daptation) [21] demonstrate advantages and flexible convenience.

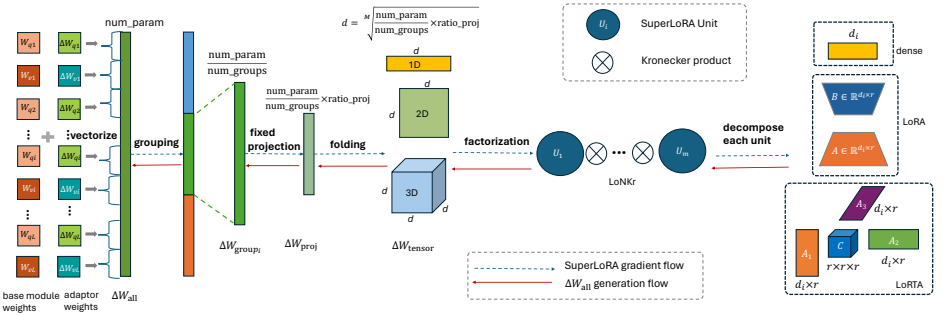


Figure 1: Schematic of SuperLoRA to fine-tune multi-layer attention modules at once with grouping, projection, folding, and factorization.

LoRA [21] approximates the weight updates of the base model by approximating the change ΔW of each weight matrix as the product of two low-rank matrices. This decreases the required parameters from d^2 to $2rd$ when $r \ll d$, where d and r are weight size and the rank, respectively. Most LoRA variants work on addressing the inherent *low-rank constraint* of matrix factorization, including LoHA (**Lo**w-rank **Ha**damard) [42], LoKr (**Lo**w-rank **Kr**onecker) [42], and LoTR (**Lo**w **T**ensor **R**ank) [5]. We discuss more related work in Section 2. However, we find these variants can be nicely unified within our framework—SuperLoRA—with different hyper-parameters as shown in Table 1. Our proposed SuperLoRA framework is depicted in Figure 1, which also yields to some new variants: LoNKr (**Lo**w-rank **N**-split **Kr**onecker) and LoRTA (**Lo**w-**R**ank **T**ensor **A**daptation). Additionally, we introduce three extended options: 1) reshaping ΔW to any arbitrary multi-dimensional tensor arrays before applying LoRA variants; 2) splitting all ΔW into an arbitrary number of groups, which breaks the boundaries for ΔW across different weights; and 3) projecting fewer number of trainable parameters into larger weights through a projection layer $\mathcal{F}(\cdot)$ with fixed parameters. Accordingly, SuperLoRA provides more flexibility and extended functionality, controlled by a set of hyper-parameters listed in Table 2. Our contributions include:

- We propose a new PEFT framework SuperLoRA which gracefully unifies and extends most LoRA variants.
- With projected tensor rank decomposition, SuperLoRA can adapt all weights across layers jointly with a wide range of adjustable parameter amount.
- We investigate the effect of tensor reshaping, grouping, random projection, and shuffling.
- We demonstrate high parameter efficiency for large ViT and diffusion models in two image transfer learning tasks: image classification and image generation, and GPT-2 model in NLP task.
- Significant parameter reduction by up to 10 folds can be achieved.

Table 1: Hyper-parameter settings in SuperLoRA and the resultant LoRA variant.

hyper-parameters settings	method
$\mathcal{F}=I$, weight-wise, $K=1, C_{g1}=I, M=1$	dense FT
$\mathcal{F}=I$, weight-wise, $K=1, C_{g1}=I, M=2$	LoRA [21]
$\mathcal{F}=I$, weight-wise, $K=2, C_{gk}=I, M=2$	LoKr [42]
$\mathcal{F}=I$, group-wise, $G=1, M>2$	LoTR [5]
$\mathcal{F}=I$, group-wise, $K>2, C_{gk}=I, M=2$	LoNKr
$\mathcal{F}=I$, group-wise, $K=1, M>2$	LoRTA

Table 2: Hyperparameters and notation.

notation	description
r	rank of factorization
$\mathcal{F}$	mapping function
ρ	compression ratio
G	number of groups
M	order of tensor modes
K	number of splits

2 Related Work

PEFT algorithms have been widely explored for transfer learning tasks in both computer vision [16, 22, 23] and NLP fields [12, 15, 24, 30, 31] as they not only save memory and time, but also require much less data for fine-tuning. Thus, PEFT enables the efficient utilization of capabilities from large pretrained models for tasks with limited data. Adapter-based methods [7, 13, 20, 35], that freeze the base model weights and fine-tune only the additional adapter parameters, stand out since their plug-and-play nature enables many downstream tasks to share the same large model, while the adapter holds only the task-specific information. The widely used method LoRA [21] and its extensions [14, 43] assume that the weight correction term can be estimated by low-rank decomposition under the low-dimensional manifold hypothesis.

Addressing the inherent *low-rank constraint* of matrix factorization in LoRA, LoHA [42] divides ΔW into two splits and combines them with Hadamard product, and KronA [11] combines the two splits with a Kronecker product to enlarge the overall rank. LoKr [42] further extended KronA to convolutional layers. LoDA (Low-Dimensional Adaptation) [32] extended LoRA by introducing nonlinearity. Our SuperLoRA can nicely generalize and extend such variants.

Instead of approximating weight-wise updates, LoTR [5] jointly approximates all ΔW across the model with careful handling to preserve the geometric meaning of each weight. Differently, SuperLoRA relaxes the geometrically meaningful boundaries by caring the total number of parameters and splitting it to any number of groups. For high-order tensor decomposition, LoTR employs more stringent Tensor Train Decomposition to deal with the core tensor explosion, while SuperLoRA coupled Tucker Decomposition with a fixed projection layer. Besides, their proposed methods are restricted to context when ΔW is the same high-order tensor, while with reshaping, LoRTA can be applied to any weight shape.

Most recent work [8] decomposes each convolution kernel into a learnable filter atom and its non-learnable counterparts. The concept of filter atom is similar to the projection layer of SuperLoRA. However, it works on each convolutional kernels separately, resulting in a waste of parameters, while SuperLoRA considers the entire model jointly. Besides, the atom coefficients are obtained from matrix factorization, while SuperLoRA uses a Fastfood projection [28], which is faster, simpler and more theoretically justifiable to exploit intrinsic dimensionality [2]. In addition, SuperLoRA can control the size of atoms directly while atoms in their method are restricted in factorization.

Local LoRA [25] aims to reduce memory consumption at fine-tuning by splitting large model into groups and then fine-tuning group-by-group sequentially, but no adjustment on the LoRA structure was proposed. Instead, SuperLoRA focuses on how to split and assign

LoRA for each group, which is a viable extension of Local LoRA.

3 SuperLoRA

3.1 Low-Rank Adaptation (LoRA)

LoRA [21] assumes the update ΔW of each weight matrix W for fine-tuning can be approximated by a low-rank mapping as $\Delta W = AB^\top$ ($[\cdot]^\top$ denotes matrix transpose), which is added to the frozen weight matrix as shown in Figure 4(a):

$$W' = W + \Delta W = W + AB^\top, \tag{1}$$

where $A \in \mathbb{R}^{d_1 \times r}$, $B \in \mathbb{R}^{d_2 \times r}$, and the rank is r. With a smaller r compared with the matrix dimensions, it only requires $(d_1 + d_2)r$ parameters for each weight matrix, while full fine-tuning (FT) for dense $\Delta W \in \mathbb{R}^{d_1 \times d_2}$ results in $d_1 d_2$ parameters. LoRA has been widely used in fine-tuning large models as much less trainable parameters save memory usage at training while retaining performance, making it easily adapted to downstream tasks.

3.2 SuperLoRA

Figure 1 shows the overview of SuperLoRA, which is a generalization of LoRA variants to allow high flexibility in the weight update ΔW. SuperLoRA can be formulated as:

$$\Delta W_{\text{group}_g} = \mathcal{F}\left(\bigotimes_{k=1}^{K} \left(C_{gk} \times_1 A_{gk1} \times_2 \cdots \times_M A_{gkM} \right) \right), \tag{2}$$

where $\mathcal{F}(\cdot)$ is a simple projection function applied on the results of SuperLoRA modules. We denote $\times_m$ as mode-m tensor product, and $\otimes$ as Kronecker product[1]. Here, M represents the order of the reshaped weight tensor modes, and high-order Tucker decomposition [41] is employed to formulate this high-order tensor, where $C_{gk} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_M}$ is the M-D core tensor and $A_{gkm} \in \mathbb{R}^{d_m \times r_m}$ are 2D plane factors. SuperLoRA units in Figure 1 are combined with Kronecker product across K splits in a proper shape. Depending on reshaping, each split has multiple choices including a combination of dense fine-tuning (FT: 1D), LoRA (2D), and high-order Tucker decomposition (3D, 4D, etc.).

For SuperLoRA as depicted in Figure 1, we first concatenate all $\Delta W \in \mathbb{R}^{d_i \times d_i}$ across multiple layers to get the total correction of $\Delta W_{\text{all}} \in \mathbb{R}^{\Sigma_i d_i^2}$. Then, ΔW_{all} is divided into g groups: $\{\Delta W_{\text{group}_g}\}$ for $g \in \{1, 2, \ldots, G\}$. Each LoRA module will then produce $\Delta W_{\text{group}_g}$. Finally, stretch $\Delta W_{\text{group}_g}$ to one dimension, fetch corresponding size of ΔW from those $\Delta W_{\text{group}_g}$ and add it to candidate weight matrix, *e.g.*, query and value projection weights for attention modules across layers. Figure 2 shows the grouping mechanism which provides various options, including weight-wise, layer-wise, and general grouping. Reshaping in Figure 2(c) can solve the unbalanced fan-in/fan-out issue in Figure 2(b) when stacking multiple weights.

SuperLoRA can further modify the tensor arrays through a simple mapping $\mathcal{F}(\cdot)$: *e.g.*, we can project much smaller ΔW_{lora_g} into larger final $\Delta W_{\text{group}_g}$ to improve the parameter efficiency. We use the Fastfood projection [2, 28] as shown in Figure 3, which is given by

$$\Delta W_{\text{group}_g} = \mathcal{F}(\Delta W_{\text{lora}_g}) = \text{vec}[\Delta W_{\text{lora}_g}]\, \mathcal{H}' \,\text{diag}[\mathcal{G}]\, \Pi\, \mathcal{H}\, \text{diag}[\mathcal{B}], \tag{3}$$

[1] Kronecker product: $C = A \otimes B$, where $A \in \mathbb{R}^{m \times n}$, $B \in \mathbb{R}^{p \times q}$, $C \in \mathbb{R}^{mp \times nq}$

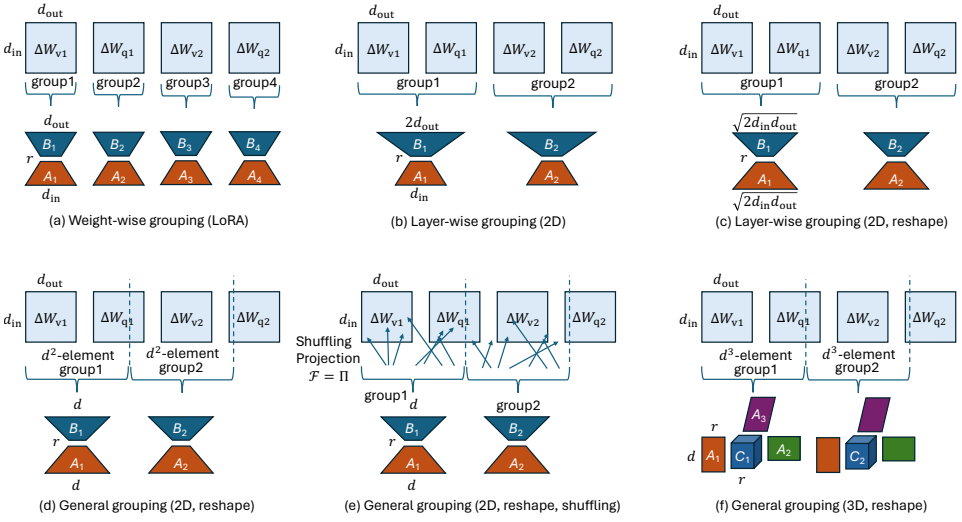


Figure 2: Examples of grouping mechanism.

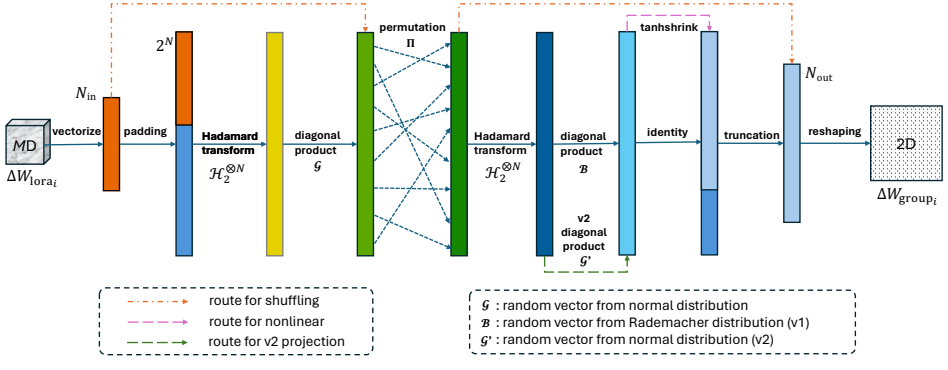
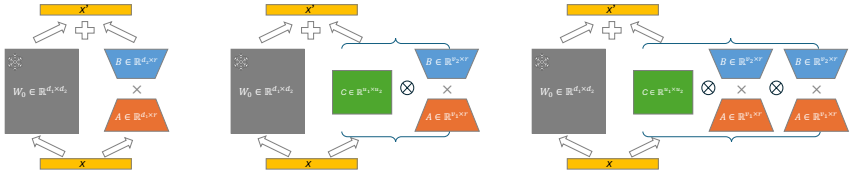


Figure 3: Illustration of Fastfood projection and its variants.

where vec[·] is a vectorization operator, diag[·] denotes a diagonalization operator, $\mathcal{H}$ is left-truncated Walsh–Hadamard matrix, $\mathcal{H}'$ is its right-truncated version, $\mathcal{G}$ is a random vector drawn from normal distribution, Π is a random permutation matrix, and $\mathcal{B}$ is a random vector drawn from Rademacher distribution. The compression ratio for the projection $\mathcal{F}(\cdot)$ is $\rho = |\Delta W_{\text{lora}_g}|/|\Delta W_{\text{group}_g}|$, where $|\cdot|$ denotes the total number of elements of the tensor. It is a fast Johnson–Lindenstrauss transform with log-linear complexity due to the fast Walsh–Hadamard transform, and no additional parameters are required when the random seed is predetermined. The projection also includes a shuffling variant as in Figure 2(e).

SuperLoRA and LoKr/LoNKr: LoKr is depicted in Figure 4(b), which can be extended as shown in Figure 4(c). We call it LoNKr, which combines K splits composed of sub LoRA units through Kronecker products: *i.e.*, $K > 2$. When $K = 2$, it reduces to LoKr but with an additional flexibility. For example, LoNKr can still adapt multiple attention modules at once with an adjustable group size G, unlike weight-wise adaptation of LoKr.

SuperLoRA and LoTR: While LoRA estimates ΔW in a weight-wise independent way,



(a) LoRA (b) LoKr (c) LoNKr (weight-wise)

Figure 4: Overview of (a) LoRA; (b) LoKr; (c) LoNKr (weight-wise).

SuperLoRA considers the whole weights ΔW_{all} jointly. It can relax the restriction of the weight shape and geometric meaning of weight axis unlike LoTR. Here, the number of groups can be adjusted to balance between parameter amount and fine-tuning performance. When the number of groups is the number of weights and the group boundary matches the weight boundary, it corresponds to weight-wise LoRA. When the number of groups is $G = 1$, SuperLoRA corresponds to LoTR [5], but with an additional projection mapping $\mathcal{F}$.

Reshaping to regular tensor: Grouping multiple layers together by concatenating ΔW along one axis results in skew $\Delta W_{\text{group}_g}$, limiting the choice of ranks in LoRA modules and leading to worse approximation. For example, stacking query and value weight updates as $[\Delta W_{\text{q}}, \Delta W_{\text{v}}]$ will be of size $d_1 \times 2d_2$, which is less efficient for LoRA as A and B matrices have unbalanced sizes. To solve this, we propose to reshape $\Delta W_{\text{group}_g}$ to a regular tensor: *i.e.*, square-like 2D matrix, cubic-like 3D tensor, or high-order hyper-cubic tensors having same dimension size across all axes. This reshaping can reduce the dimension per axis in the order of $\mathcal{O}[N^{1/M}]$ for N being the number of stacking weights, that in return can allow higher rank size per plane factors. Several examples of grouping and reshaping are discussed in Appendix A.3, and its geometric analysis in Appendix A.4.

LoRTA: Folding a matrix $\Delta W_{\text{group}_g}$ into high-order tensor (*e.g.*, 3D, 4D, 5D) can decrease parameters with tensor rank decomposition, like Tucker decomposition, where $\Delta W_{\text{group}_g}$ is represented by M 2D plane factors and one MD core tensor. We refer to this variant of SuperLoRA using Tucker decomposition as LoRTA. For example, when $M = 3$ and $K = 1$, we have 3D tensor rank decomposition for $\Delta W_{\text{group}_g} \in \mathbb{R}^{d1 \times d2 \times d3}$ as follows:

$$\Delta W_{\text{group}_g} = C_{gK} \times_1 A_{gK1} \times_2 A_{gK2} \times_3 A_{gK3}, \tag{4}$$

where $C_{gK} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ is a reshaped 3D core tensor, $A_{gKm} \in \mathbb{R}^{d_m \times r}$ is a mode-m 2D plane factor, and $\times_m$ denotes mode-m tensor product. For simplicity, we set a rank $r = r_m$ for any mode $m \in \{1, 2, \ldots, M\}$.

The core tensor may cause the explosion of parameters with larger rank as the number of parameters is exponential as r^M. This may be resolved by restricting the core tensor to be strongly diagonal or identity. For instance, $M = 2$ with identity core tensor $C_{gK} = I$ corresponds to the original LoRA, and $M = r = 1$ identity core tensor corresponds to the dense FT. When using a diagonal core tensor, it reduces to Candecomp-Parafac (CP) decomposition. Figure 7 shows the number of required parameters with CP decomposition. One can see that higher-order tensor decomposition can significantly reduce the total number of trainable parameters at a certain rank. We provide another solution without limiting the core tensor by coupling with the projection layer $\mathcal{F}$ below.

Shuffling: Another simple projection is to use a shuffling function without compression. It can be achieved by simplifying the fastfood projection without $\mathcal{H}$, $\mathcal{H}'$, $\mathcal{G}$, and $\mathcal{B}$ but with

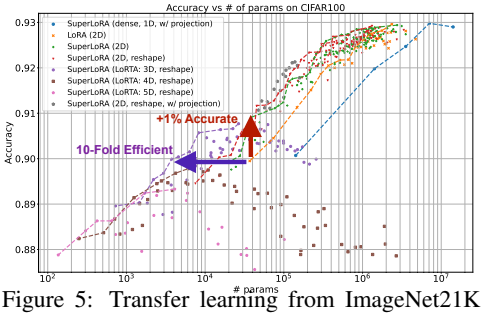


Figure 5: Transfer learning from ImageNet21K to CIFAR100, parameters in classifier head excluded.

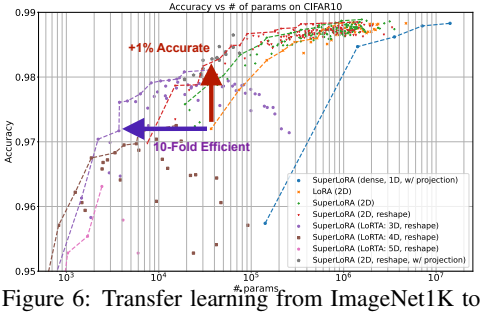


Figure 6: Transfer learning from ImageNet1K to CIFAR10, freeze classifier head after manual label matching.

the random permutation Π and projection ratio $\rho = 1$. As SuperLoRA updates all weights at once, we have a flexibility in a way to distribute $\Delta W_{\text{group}_g}$ towards which element of W. To understand how the weight assignment method impacts, we consider a random shuffling case for the projection function $\mathcal{F}$. Several projection variants including shuffling are discussed in Appendix A.6.

4 Transfer Learning Experiments

4.1 Classification transfer task

Transfer learning for image classification is conducted between ImageNet21k [9] and CIFAR100 [26] based on a ViT-base [10] model. More details of the ViT model are described in Appendix A.1. The query and value projection layers in the attention modules are fine-tuned with SuperLoRA. The model is trained for 5,000 steps with the stochastic gradient descent (SGD) optimizer, with a batch size of 128 and a learning rate of 0.05. The OneCycleLR [38] scheduler is used.

Classification results versus the number of parameters are shown in Figure 5 with Pareto frontier lines. Comparing group-wise SuperLoRA (2D with/without reshape) with weight-wise LoRA, we can find that SuperLoRA show better performance in terms of the trade-off between classification accuracy and the number of parameters. Noticeably, we observe three to four times advantage in terms of parameter efficiency for the same accuracy. As the largest number of groups is set to 24 (*i.e.* LoRA), it indicates smaller number of groups are superior. This may be because ViT model is excessively large for the CIFAR100 dataset, with much more redundant weights. Grouping weights and layers together can reduce noise brought by the redundancy. With reshaping $\Delta W_{\text{group}_g}$ to a square matrix, classification accuracy further increases in the lower parameter regime and the range of parameters the model can cover becomes wider as higher rank can be used while maintaining a smaller number of parameters.

To examine the effect of higher-order tensor folding, the order M is set to be 3, 4 and 5 for SuperLoRA (*i.e.* LoRTA) as well as 2. For $M = 2$ cases with 2D tensor, we use identity core tensor like typical LoRA. With the increase of order from 2 to 5, higher order takes place lower-order at fewer-parameter regimes. Moreover, data points for high-order LoRTA show a hill-like trend with the increase of parameters. This may be caused by the inefficient core tensor, which increases parameters rapidly without benefiting the accuracy. When comparing the lowest rank LoRA (which achieves around 0.9 accuracy with about 4×10^4 parameters),

our LoRTA (3D) significantly improves the accuracy by about 1% at the comparable number of parameters, and more significantly reduces the number of parameters by 10 folds to keep the comparable accuracy of 0.9.

Finally, we address the impact of the projection layer $\mathcal{F}(\cdot)$. Fixed fastfoood projection as in Figure 3 is applied on SuperLoRA. For 1D dense, the plot for a projection ratio of $\{1, 0.5, 0.25, 0.1, 0.01\}$ is placed from right to left in Figure 5. The classification accuracy dropped less than 1% from projection ratio 1 to 0.1 (*i.e.* 90% less parameters), but it is worse than LoRA. To get some results of projection for the number of parameters around 10^4 and 10^5, we select a few settings for SuperLoRA (2D, reshape) with $G = 1$ as shown in the figure with a marker of dark stars. Most projection results demonstrate better accuracy compared with other SuperLoRA settings without projection in the same number of parameters level. This result shows a smaller adapter with fixed projection layer is a strong functionality to improve the parameter efficiency of SuperLoRA.

We confirmed the remarkable gain of our SuperLoRA on a transfer learning task for image classification with ViT models. In Appendix A.4, we further discussed the geometric analysis of SuperLoRA, and grouping impacts in Appendix A.5. In addition, We evaluated the advantage in another transfer learning task for image generation with diffusion models in Section 4.2 and appendices A.9, A.11 and A.12.

4.2 Image generation transfer task

4.2.1 Settings

For the image generation task, SuperLoRA is evaluated by transfer learning between SVHN [34] and MNIST datasets [29]. Both datasets have 10 classes corresponding to images of the digits 0 to 9, where the SVHN images have a more complicated color background, while the MNIST images are nearly black-and-white with black background. The generative model is a classifier-free diffusion model [18] and more details can be found in Appendices A.2 and A.8. We focus on the transfer learning from SVHN to MNIST. The reverse transfer learning from MNIST to SVHN is discussed in Appendix A.12. As we found ℓ_1-distance based IS is more consistent to the perceptual visual quality, we focus on IS metric results in the main content, while the results for other metrics can be found in Appendix A.11. For following figures, Pareto frontier lines/dots are mainly shown to provide the limit of each method, while Appendix provides more complete figures with all data points.

4.2.2 Grouping effect

First, we evaluated how splitting all ΔW_{all} into multiple groups affects the performance. Figure 9 shows the results of dense, original weight-wise LoRA and group-wise SuperLoRA with different number of groups. Sweeping the rank and the number of groups, we plot the image quality metrics in y-axis and the required number of trainable parameters in x-axis. Pareto frontier lines/data points are also shown in the figure.

Figure 9 shows that the dense FT for ΔW presents the best IS, while requiring most parameters. Original weight-wise LoRA is closest to dense, in terms of both IS and parameter amount. However, in low-parameter regimes, SuperLoRA (2D, group1), *i.e.* LoTR, shows the best results compared with other grouping. While in the middle of parameter amount axis, other splittings including groups $G = 8$ and 12 show slightly better IS compared with LoRA. Besides, splitting ΔW_{all} yields much more tradeoff points compared with both LoRA

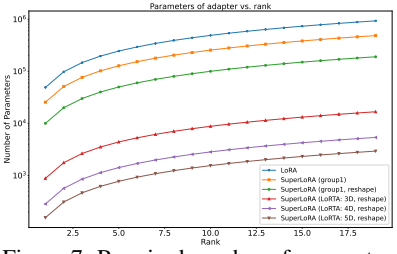


Figure 7: Required number of parameters.

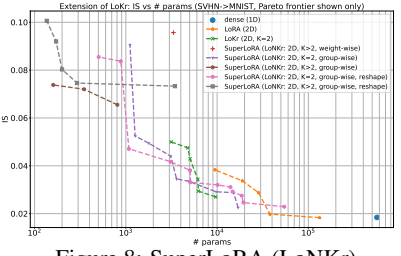


Figure 8: SuperLoRA (LoNKr)

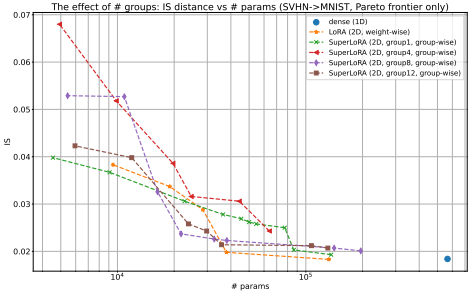


Figure 9: weight-wise *vs*. group-wise

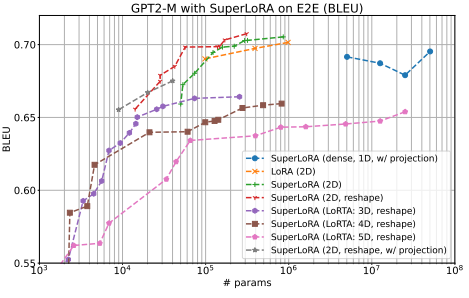


Figure 10: BLEU scores

and dense, providing us higher flexibility to adjust the trade-off between quality and parameter efficiency especially when the memory resource is limited.

4.2.3 LoKr vs. LoNKr

In 2D ΔW, we also compared LoKr with our proposed extension LoNKr, a variant of SuperLoRA. We evaluated LoNKr when the number of splits is $K \in \{2,3,4\}$, where $K = 2$ corresponds to the original LoKr. For the dense factor on the left in LoNKr/LoKr as shown in Figure 4(c), dimension is fixed to 6, 8 or 10. Figure 8 shows that more splits provide us more choices in low-parameter regimes, especially for group-wise LoNKr. LoNKr shows much more data points and better IS when the number of parameters is less than 5,000. And the least parameter for LoKr and LoNKr dropped greatly from 500 to 150 as in Figure 31(b).

4.3 Transfer learning on LLM task

We also evaluated SuperLoRA on LLM task. Specifically, following LoRA's settings, we also evaluated SuperLoRA on the GPT2-M model and with the E2E NLG challenge dataset.

4.3.1 Comparison with other adapter-based methods

For SuperLoRA, we selected the hyperparameters that required least trainable parameters while it can achieve comparable/better BLEU score. And the final metrics are obtained by averaging the scores from 3 different seeds. As shown in Table 3, with 70% less trainable parameters compared with LoRA, SuperLoRA still achieves better results in terms of all

Table 3: GPT-2 medium with different adaptation methods on E2E NLG Challenge. For all metrics, higher is better. * indicates numbers published in prior works, as compiled by Hu et al. [21].

Method	# Trainable Parameters	E2E NLG Challenge				
		BLEU	NIST	MET	ROUGE-L	CIDEr
FT*	354.92M	68.2	8.62	46.2	71.0	2.47
$\text{Adapter}^{\text{L}}$*	0.37M	66.3	8.41	45.0	69.8	2.40
$\text{Adapter}^{\text{L}}$*	11.09M	68.9	8.71	46.1	71.3	2.47
$\text{Adapter}^{\text{H}}$*	11.09M	$67.3_{\pm.6}$	$8.50_{\pm.07}$	$46.0_{\pm.2}$	$70.7_{\pm.2}$	$2.44_{\pm.01}$
FT^{Top2}*	25.19M	68.1	8.59	46.0	70.8	2.41
FT^{W_q,W_v}	48.00M	$69.4_{\pm.1}$	$8.74_{\pm.02}$	$46.0_{\pm.0}$	$71.0_{\pm.1}$	$2.48_{\pm.01}$
LoRA	0.40M	$69.28_{\pm.01}$	$8.73_{\pm.08}$	$46.51_{\pm.00}$	$71.4_{\pm.00}$	$2.49_{\pm.02}$
SuperLoRA	**0.12M**	$\mathbf{69.82}_{\pm.00}$	$\mathbf{8.76}_{\pm.02}$	$\mathbf{46.54}_{\pm.00}$	$\mathbf{71.5}_{\pm.00}$	$\mathbf{2.50}_{\pm.01}$

metrics, BLEU, NIST, MET, ROUGE-L and CIDEr, demonstrating the efficiency of SuperLoRA.

4.3.2 Exploration of SuperLoRA on GPT2-M model

As in image classification task, we evaluated SuperLoRA with many different hyperparameter settings to span across the parameter axis as shown in Figure 10. Comparing Figures 5, 6 and 10, most observations are similar, except for dense with projection, larger compression ratios can return a better BLEU score than smaller compression ratio in LLM task, while the accuracy dropped with the increase of compression ratio in image classification tasks. Results from other metrics and geometric analysis can be found in Appendix A.10.

5 Conclusion

We proposed a new unified framework called SuperLoRA, which generalizes and extends LoRA variants. SuperLoRA provides some extended variants, which we refer to as LoNKr and LoRTA. It offers a rich and flexible set of hyper-parameters, including the rank of factorization, the choice of projection function, projection ratio, the number of groups, the order of tensor, and the number of Kronecker splits. Through transfer learning experiments, we demonstrated that SuperLoRA achieves promising results in parameter efficiency for fine-tuning at low-parameter regimes. We could reduce the required number of parameters by 3 to 10 folds compared to LoRA. Future work includes studying the projection functions to further improve the efficiency in extremely-low-parameter regimes.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report, 2023.

[2] Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7319–7328, 2021.

[3] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[4] Rohan Anil, Andrew M. Dai, and Orhan Firat et al. PaLM 2 technical report, 2023.

[5] Daniel Bershatsky, Daria Cherniuk, Talgat Daulbaev, and Ivan Oseledets. LoTR: Low tensor rank weight adaptation. *arXiv preprint arXiv:2402.01376*, 2024.

[6] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International Conference on Learning Representations*, 2018.

[7] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022.

[8] Wei Chen, Zichen Miao, and Qiang Qiu. Parameter-efficient tuning of large convolutional models. *arXiv preprint arXiv:2403.00269*, 2024.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[11] Ali Edalati, Marzieh Tahaei, Ivan Kobyzev, Vahid Partovi Nia, James J Clark, and Mehdi Rezagholizadeh. Krona: Parameter efficient tuning with Kronecker adapter. *arXiv preprint arXiv:2212.10650*, 2022.

[12] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, 2021.

[13] Tianxiang Hao, Hui Chen, Yuchen Guo, and Guiguang Ding. Consolidator: Mergable adapter with group connections for visual adaptation. In *The Eleventh International Conference on Learning Representations*, 2022.

[14] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *Forty-first International Conference on Machine Learning*, 2024.

[15] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*, 2021.

[16] Xuehai He, Chunyuan Li, Pengchuan Zhang, Jianwei Yang, and Xin Eric Wang. Parameter-efficient model adaptation for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 817–825, 2023.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[18] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[19] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[20] 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*, pages 2790–2799. PMLR, 2019.

[21] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

[22] Shibo Jie and Zhi-Hong Deng. Fact: Factor-tuning for lightweight adaptation on vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1060–1068, 2023.

[23] Shibo Jie, Haoqing Wang, and Zhi-Hong Deng. Revisiting the parameter efficiency of adapters from the perspective of precision redundancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17217–17226, 2023.

[24] Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021.

[25] Oscar Key, Jean Kaddour, and Pasquale Minervini. Local LoRA: Memory-efficient fine-tuning of large language models. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@ NeurIPS 2023)*, 2023.

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[27] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[28] Quoc Le, Tamás Sarlós, Alex Smola, et al. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, page 8, 2013.

[29] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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

[31] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, 2021.

[32] Jing Liu, Toshiaki Koike-Akino, Pu Wang, Matthew Brand, Ye Wang, and Kieran Parsons. LoDA: Low-dimensional adaptation of large language models. *NeurIPS'23 Workshop on on Efficient Natural Language and Speech Processing*, 2023.

[33] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, number 5, page 7. Granada, Spain, 2011.

[35] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-destructive task composition for transfer learning. In *16th Conference of the European Chapter of the Associationfor Computational Linguistics, EACL 2021*, pages 487–503. Association for Computational Linguistics (ACL), 2021.

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[37] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. *Advances in neural information processing systems*, 29, 2016.

[38] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[39] Hugo Touvron, Louis Martin, and Kevin Stone et al. Llama 2: Open foundation and fine-tuned chat models, 2023.

[40] Anton Tsitsulin, Marina Munkhoeva, Davide Mottin, Panagiotis Karras, Alex Bronstein, Ivan Oseledets, and Emmanuel Mueller. The shape of data: Intrinsic distance for data distributions. In *International Conference on Learning Representations*, 2019.

[41] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[42] Shin-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B W Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From lyCORIS fine-tuning to model evaluation. In *The Twelfth International Conference on Learning Representations*, 2024.

[43] Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Saez De Ocariz Borde, Rickard Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *International Conference on Learning Representations*, 2024.