COMCAT: Towards Efficient Compression and Customization of Attention-Based Vision Models

Jinqi Xiao [1] **Miao Yin** [1] **Yu Gong** [1] **Xiao Zang** [1] **Jian Ren** [2] **Bo Yuan** [1]

Abstract

Attention-based vision models, such as Vision Transformer (ViT) and its variants, have shown promising performance in various computer vision tasks. However, these emerging architectures suffer from large model sizes and high computational costs, calling for efficient model compression solutions. To date, pruning ViTs has been well studied, while other compression strategies that have been widely applied in CNN compression, *e.g.*, model factorization, is little explored in the context of ViT compression. This paper explores an efficient method for compressing vision transformers to enrich the toolset for obtaining compact attention-based vision models. Based on the new insight on the multi-head attention layer, we develop a highly efficient ViT compression solution, which outperforms the state-of-the-art pruning methods. For compressing DeiT-small and DeiT-base models on ImageNet, our proposed approach can achieve 0.45% and 0.76% higher top-1 accuracy even with fewer parameters. Our finding can also be applied to improve the customization efficiency of text-to-image diffusion models, with much faster training (up to $2.6\times$ speedup) and lower extra storage cost (up to $1927.5\times$ reduction) than the existing works. The code and models are publicly available at `https://github.com/jinqixiao/ComCAT`.

1. Introduction

Recently the attention-based vision models have achieved comparable or superior performance than the convolution-centered architecture in various computer vision tasks, demonstrating the promising benefit brought by using an attention mechanism (Liu et al., 2021b; Caron et al., 2021; Xie et al., 2021; Carion et al., 2020). On the downside, these emerging architectures, *e.g.*, vision transformer (ViT) and its variants (Touvron et al., 2021; Liu et al., 2021a; Li et al., 2022), suffer from even larger model sizes and higher computational costs than convolutional neural networks (CNNs), hindering their efficient deployment in many practical resource-constrained scenarios.

An attractive solution to this challenge is to perform model compression, a strategy that can reduce network size without affecting task performance. Motivated by the huge prior success of compressing CNNs (Hinton et al., 2015; Han et al., 2015), several recent studies (Yin et al., 2023; Yu et al., 2022b; Hou & Kung, 2022) have proposed to apply one (*e.g.*, pruning) or combining several (*e.g.*, pruning and knowledge distillation) compression methods for vision transformers, bringing considerable reduction in model size and/or FLOPs.

Different from the existing works, this paper aims to address the above analyzed efficiency challenge from another perspective – exploring the low-rankness of the attention-based vision models. To date, a rich set of low-rank compression techniques for CNNs have been proposed in the literature (Kim et al., 2015; Yin et al., 2021; Liebenwein et al., 2021; Yin et al., 2022b;a; Xiao et al., 2023; Xiang et al., 2023). However, consider 1) there exists a substantial difference on network architecture and operation mechanism, *e.g.*, multi-head attention in ViTs *vs.* channel-wise convolution in CNNs; and 2) as indicated in (Yu & Wu, 2023) and verified by our analysis, many weight matrices in the vision transformers do not exhibit low rankness, it is not clear that whether low-rank compression would bring the satisfied improvement on model efficiency. From the perspective of practical deployment, a question naturally arises: *For compressing attention-based vision models, can exploring model low-rankness provide comparable or even better performance than other methods such as pruning?*

To answer this question and fully unleash the potential of low-rank compression for ViTs and attention-based models, this paper first investigates the low-rankness in the multi-ahead attention layer, and proposes that the head-level low-rankness, instead of weight matrix-level, should be explored. Based on this new insight, we then develop a highly efficient

[1]Rutgers University [2]Snap Inc. Correspondence to: Jinqi Xiao <jinqi.xiao@rutgers.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

low-rank ViT compression solution with automatic rank selection. Compared with the state-of-the-art ViT pruning methods, the proposed approach can achieve 0.45% and 0.76% higher top-1 accuracy even with fewer parameters, for compressing DeiT-small and DeiT-base models on ImageNet dataset, respectively. Furthermore, our finding can also be applied to improve the efficiency for customizing text-to-image diffusion modules (Ruiz et al., 2022; Kumari et al., 2022), a recent emerging and important computer vision task, with much faster training (up to $2.6\times$ speedup) and lower extra storage cost (up to $1927.5\times$ reduction) than the state-of-the-art customization solutions.

2. Related Works

Pruning for Vision Transformers. To reduce the model size and achieve practical speedup, structured pruning on different substructures of ViT models, *e.g.*, attention heads, blocks, and rows of weight matrices, have been studied in the literature (Hou & Kung, 2022; Yu et al., 2022b; Zhu et al., 2021; Chen et al., 2021b). In addition, another research direction proposes to improve model processing speed via using dynamic or static token pruning (Bolya et al., 2022; Pan et al., 2021b; Tang et al., 2022; Goyal et al., 2020; Pan et al., 2021a). Recently, Yu *et al.* (Yu et al., 2022b) develop a unified framework to jointly perform pruning, knowledge distillation and block skipping, achieving state-of-the-art ViT compression performance.

Low-rank Compression of Weight Matrices (W) in NLP Transformers. Most of the existing studies on low-rank compressed transformers are concentrated in the NLP field. Noach *et al.* (Noach & Goldberg, 2020) decompose the weight matrices of the pre-trained language models (PLMs) using SVD and perform feature distillation to improve model performance. Ren *et al.* (Ren et al., 2022) adopt tensor decomposition to compress PLMs and achieve practical inference speedups. Hsu *et al.* (Hsu et al., 2022) introduce Fisher information to measure the importance of parameters to factorize task-specific PLMs.

Low-Rank Approximation for Attention Matrices (Q, K, V). Another line of works is to perform low-rank approximation for the attention matrices, the intermediate results from the attention mechanism. Different types of approximation schemes, including adding additional projection matrices and sparse approximation, have been investigated (Wang et al., 2020; Choromanski et al., 2020; Chen et al., 2021a). As the orthogonal effort from our approach, these methods do not reduce the model sizes of transformers.

Personalized Text-to-Image Diffusion Models. Recently released text-to-image diffusion models (Rombach et al., 2022; Ramesh et al., 2022; Saharia et al., 2022; Yu et al., 2022a) have shown impressive content-generation capability. A very emerging and practical demand is to make these pre-trained models customized for a user-provided specific concept. To that end, some efforts leverage transfer learning via fine-tuning all the parameters or introducing a word vector for the new concept (Ruiz et al., 2022; Gal et al., 2022). However, the large sizes of the diffusion models bring costly training time and high extra storage requirements during the fine-tuning. To improve the efficiency of customization, Kumari *et al.* (Kumari et al., 2022) propose to only fine-tune the key and value mapping from text to latent features in the cross-attention layers; while freezing other parts.

3. Method

3.1. Preliminaries

The attention operation can be viewed as the mapping from a query and a set of key-value pairs to an output. To better extract and learn the information from different representation subspace and spatial regions, the state-of-the-art attention-based vision models adopt multi-head attention (MHA) as:

$$\mathrm{MHA}(X_Q, X_K, X_V) = \mathrm{Concat}(head_1, \ldots, head_h)W^O, \quad (1)$$

where $X_Q, X_K, X_V \in \mathbb{R}^{n\times d_m}$ are the input embedding matrices, n is the sequence length, d_m is the embedding dimension, and h is the number of heads. For each $head_i$, it performs attention operation as follows:

$$\begin{aligned} head_i &= \mathrm{Attention}(X_Q W_i^Q,\ X_K W_i^K,\ X_V W_i^V) \\ &= \mathrm{Softmax}(\frac{X_Q W_i^Q (X_K W_i^K)^T}{\sqrt{d_k}}) X_V W_i^V, \end{aligned} \quad (2)$$

where W_i^Q, $W_i^K \in \mathbb{R}^{d_m\times d_k}$, $W_i^V \in \mathbb{R}^{d_m\times d_v}$, $W^O \in \mathbb{R}^{hd_v\times d_m}$ are the weight matrices, and d_k and d_v are the dimension of X_Q and X_K, respectively. Since $d_k = d_v = \frac{d_m}{h}$, for simplicity we use d instead of d_k and d_v.

3.2. Exploring Low-Rankness in MHA Layer

In this subsection, we describe our proposed low-rank MHA for efficient vision models. We first demonstrate the low-rankness of the weight matrices in each attention head, and analyze the limitation when only leveraging the matrix-level low-rankness. Built on these observation and analysis, we then propose to explore the head-level low-rankness and formulate the mechanism. We further detail the procedure of using this finding to improve model efficiency in two important scenarios: compressing vision transformers and customization of diffusion models.

Low-Rankness of W_i^Q, W_i^K, W_i^V, W^O. Low-rankness of the weight matrices has been widely observed in many types of deep learning architectures including CNN and NLP transformers (Kim et al., 2015; Ren et al., 2022), inspiring us to explore its potential existence in the attention-based

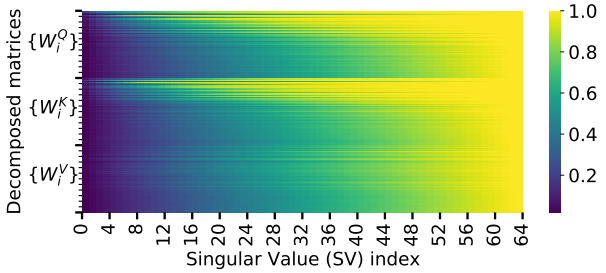

Figure 1. The heatmap of the cumulative singular values after performing SVD for each weight matrices in the MHA layers of the pre-trained DeiT-base model. It is seen that some matrices in some layers exhibit weaker low-rankness than others, implying that directly factorizing individual weight matrix may not be efficient.

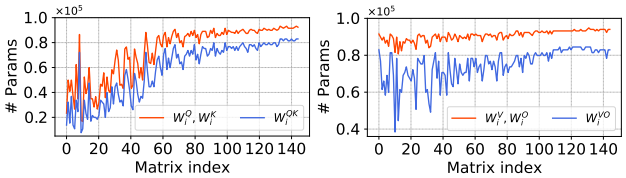

Figure 2. The number of parameters when the ratio of the cumulative singular values reaches 90% with factorizing W^Q, W^K, W^V, W^O, and the corresponding W^{QK}, W^{VO} of the pre-trained DeiT-base. It is seen that exploring head-level low-rankness is more parameter efficient than matrix-level.

vision models. To that end, we analyze the distributions of the singular values of the weight matrices (W_i^Q, W_i^K, W_i^V,W^O) in the pre-trained DeiT-base model (Touvron et al., 2021). Figure 1 shows the heatmap of the cumulative singular values after applying Singular Value Decomposition (SVD) into each weight matrices. It is seen that the phenomenon that most information is concentrated in part of singular values (the largest ones) indeed exist in the weight matrices across different heads and layers, indicating the potential of exploring low rankness of attention-based models.

Limitation of Matrix-Level Low-Rankness. Based on the above observation, a natural idea is to construct each weight matrix (W_i^Q, W_i^K, W_i^V,W_i^O) in its own low-rank format. However, we argue that the benefit brought this straightforward strategy would not be significant. As shown in Figure 1, some types of weight matrices, *e.g.*, W_i^V, do not exhibit sufficient low-rankness, a phenomenon that is also observed in the matrices of the higher layers, thereby limiting the overall potential performance improvement brought by low-rank factorization.

Exploring Head-Level Low-Rankness. To better leverage the low rankness in the MHA layer and fully unleash the potential benefits, we propose to explore the low-rank property at the head level for efficient multi-head attention. Our idea is motivated by the observation that there exists consecutive linear transformations in the attention head, *e.g.*, $X_Q W_i^Q (X_K W_i^K)^T = X_Q (W_i^Q W_i^{K^T}) X_K^T$ in Eq. 2, opening up the opportunities of constructing the

Table 1. Top-1 accuracy (**without fine-tuning**) with factorizing the individual weight matrices ("Matrix-Level") and the combined matrices ("Head-Level") in the attention layer of the pre-trained DeiT-small distilled model (original accuracy is 80.90%).

# of Params. in MHA ↓		**20%**	**40%**	**60%**	**80%**
Top-1(%)	**Head-Level**	**79.81**	**76.11**	**63.56**	**11.5**
	Matrix-Level	73.01	56.15	22.95	0.73

combinations of weight matrices, *e.g.*, $W_i^Q W_i^{K^T}$, instead of the individual matrix, in the low-rank format. According to linear algebra, such reformulation brings two benefits. First, it provides more parameter-efficient low-rank solution. More specifically, for two full-rank matrices $A \in \mathbb{R}^{n\times d}$ and $B \in \mathbb{R}^{d\times n}$, the total number of parameters of their rank-r approximations $A' \in \mathbb{R}^{n\times r}$ and $B' \in \mathbb{R}^{r\times n}$ is $2nr$; while the same rank-r approximation $C' \in \mathbb{R}^{n\times r}$ for $C = AB$ only contains nr parameters. Second, it relaxes the constraints of applying low-rankness approximation. Since the low-rankness of A or B, instead of both, is sufficient to lead to the low-rank C, it indicates that the head-level low-rankness is a more common and feasible opportunity when aiming to leverage low-rankness in the MHA layer.

Motivated by these benefits, now we formulate our idea in the context of multi-head attention mechanism. First Eq. 1 and Eq. 2 can be reformulated as:

$$\begin{aligned}\mathrm{MHA}(X_Q, X_K, X_V) &= \sum_{i=1}^{h} head_i W_i^O \\ &= \sum_{i=1}^{h} \mathrm{Softmax}(\frac{X_Q (W_i^Q W_i^{K^T}) X_K^T}{\sqrt{d_k}}) X_V (W_i^V W_i^O),\end{aligned} \tag{3}$$

where $W_i^O \in \mathbb{R}^{d\times d_m}$ and $W^O = \mathrm{Concat}(W_1^O, \ldots, W_h^O)$. Recall that a matrix $W \in \mathbb{R}^{in\times out}$ can be low-rank approximated by performing SVD as $W \approx W' = U\Sigma S' = US$, where $U \in \mathbb{R}^{in\times r}$, $S' \in \mathbb{R}^{r\times out}$, diagonal matrix $\Sigma \in \mathbb{R}^{r\times r}$, $S = \Sigma S' \in \mathbb{R}^{r\times out}$, and r is the rank value. Then the entire multi-head attention (Eq. 3) can be constructed in the low-rank format as:

$$\begin{aligned}&\mathrm{MHA}(X_Q, X_K, X_V) \\ &\quad\approx \sum_{i=1}^{h} \mathrm{Softmax}(\frac{X_Q (U_i^Q S_i^{K^T}) X_K^T}{\sqrt{d_k}}) X_V (U_i^V S_i^O) \\ &\quad= \mathrm{Concat}(head_1', \ldots, head_h') W'^O, \quad \text{where} \\ &head_i' = \mathrm{Attention}(X_Q U_i^Q,\ X_K S_i^K,\ X_V U_i^V) \\ &\qquad= \mathrm{Softmax}(\frac{X_Q U_i^Q (X_K S_i^K)^T}{\sqrt{d_k}}) X_V U_i^V, \\ &S^O = \mathrm{Concat}(S_1^O, \ldots, S_h^O).\end{aligned} \tag{4}$$

Here $W_i^Q W_i^{K^T} \approx U_i^Q S_i^{K^T} (\mathrm{rank} = r_1)$, $W_i^V W_i^O \approx U_i^V S_i^O (\mathrm{rank} = r_2)$, $U_i^Q, S_i^K \in \mathbb{R}^{d_m\times r_1}$, $U_i^V, S_i^O \in \mathbb{R}^{d_m\times r_2}$, and $S^O \in \mathbb{R}^{hr_2\times d_m}$.

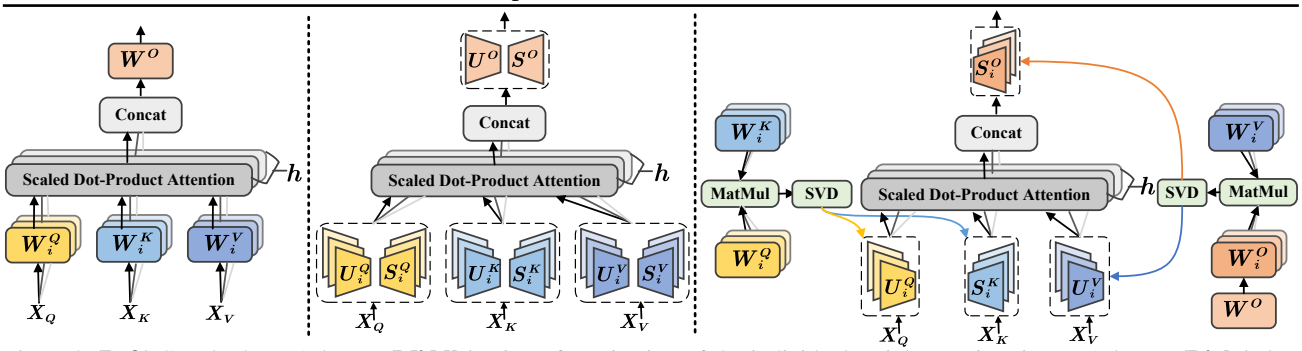

Figure 3. **(Left)** Standard MHA layer. **(Middle)** Direct factorization of the individual weight matrices in MHA layer. **(Right)** Our proposed exploring head-level low-rankness of MHA layer.

Notice that as shown in Eq. 4, in addition to exploring the low-rankness of $W_i^{QK} = W_i^Q W_i^{K^T}$, the inter-matrix correlation between W_i^V and W_i^O is also considered, bringing the low-rank construction for $W_i^{VO} = W_i^V W_i^O$. In Figure 2, we compare the direct low-rank decomposition of W_i^Q and W_i^K with the decomposition of $W_i^Q W_i^{K^T}$ and report the required number of parameters after performing low-rank factorization in the case that the ratio of the cumulative singular values reaches 90%. By comparing the required number of parameters to reach this threshold, we can evaluate and compare the low-rankness of the original matrices, *i.e.*, fewer parameters indicate better low-rankness. This is because to preserve the same amount of information, *e.g.*, 90% cumulative singular values, the matrix with better low-rankness requires fewer parameters. As shown in this figure, the number of parameters required for factorizing $W_i^Q W_i^{K^T}$ (blue line) to reach the 90% threshold is always smaller than that for W_i^Q and W_i^K (red line), indicating that the combination matrix shows better low-rankness. Table 1 illustrates the benefit of such head-level low-rank MHA mechanism, with its application for fine-tuning-free ViT compression as example. Compared to directly applying SVD to the individual weight matrices, our approach brings much higher model accuracy with the same compression ratio, verifying the two benefits (parameter efficiency and relaxed low-rank constraint) indicated in our prior analysis.

Low-Rank MHA for Vision Transformer Compression. A direct application of our proposed low-rank MHA is to compress vision transformers. In general, for a b-block ViT with one MHA layer and one 2-layer feedforward network (FFN) per block, the corresponding compression task using low-rank MHA can be formulated as follows:

$$\min_{\{W_{i,j}^{QK}, W_{i,j}^{VO}, W_{k,j}^{FFN}\}_{i=1,j=1,k=1}^{h,b,2}} \mathcal{L}(\{W_{i,j}^{QK}, W_{i,j}^{VO}, W_{k,j}^{FFN}\})$$
$$\text{s.t.} \sum_{j=1}^{b}(\sum_{i=1}^{h}\mathcal{C}(\mathcal{R}(W_{i,j}^{QK})) + \mathcal{C}(\mathcal{R}(W_{i,j}^{VO})) + \sum_{k=1}^{2}\mathcal{C}(\mathcal{R}(W_{k,j}^{FFN}))) \leq \varepsilon, \tag{5}$$

where $\mathcal{L}(\cdot)$, $\mathcal{C}(\cdot)$ and $\mathcal{R}(\cdot)$ are the functions that return the loss, cost (*e.g.*, model size or FLOPs) and rank, respectively. $W_{i,j}^{QK} = W_{i,j}^Q W_{i,j}^{K^T}$, $W_{i,j}^{VO} = W_{i,j}^V W_{i,j}^O$ and $W_{k,j}^{FFN}$ are the combination matrices of the i-th attention head and weight matrix in the FFN in the j-th block. It is seen that given the target cost budget (ε) of the compressed ViTs, rank is an important type of hyperparameter that directly determines the accuracy and complexity. In practice, because the huge range of the possible rank values, the proper rank selection for all the blocks and layers of vision transformers is challenging. For instance, there exist 4.53×10^{188} rank combinations when performing low-rank compression for DeiT-small model, making manual selection impracticable.

To address this challenge, we propose an automatic rank selection approach to efficiently incorporate low-rank MHA into ViT compression. Our key idea is to interpret the automatic rank selection of low-rank compression as a specialized neural architecture search (NAS), considering the fact that the choice of the rank essentially decides the final structure of the compressed ViT. From this insight, the proper rank value can be identified via differentiable sampling-based search, a strategy has been well studied in NAS literature (Liu et al., 2018; Wu et al., 2019; Tan & Le, 2019).

Figure 4 illustrates the overall framework for automatic rank search for low-rank ViT. Here for simple notation, we use $W_j^* \in \mathbb{R}^{in_j \times out_j}$ to denote the matrices that need to be decomposed in the j-th layer, i.e., $W_{i,j}^{QK}$, $W_{i,j}^{VO}$ and $W_{k,j}^{FFN}$, with the candidate rank set as $R_j = \{r_j^1, r_j^2, r_j^a, \dots r_j^{max}\}$. Assume that $r_j^* \in R_j$ is the currently selected rank for $W_j \approx W_j^* = U_j^* S_j^*$ ($U_j^* \in \mathbb{R}^{in_j \times r_j^*}$ and $S_j^* \in \mathbb{R}^{r_j^* \times out_j}$). Then we alternately update the selection probability $P_j = \{p_j^1, p_j^2, p_j^a, \dots p_j^{max}\}$ for rank candidates and the parameters of W_j^*. To be specific, because P_j is calculated via GumbelSoftmax (Jang et al., 2016), *i.e.*, $P_j = \text{GumbelSoftmax}(\alpha_\mathbf{j})$ with learnable vector $\alpha_\mathbf{j}$, P_j can be updated via minimizing the following loss (with the frozen weight parameters):

$$\mathcal{L}_{Prob} = \mathcal{L}_{CE}(\overline{Y}, Y) \cdot (\frac{\sum_{j=1}^{b}\sum p_j^a \mathcal{C}(r_j^a)}{\varepsilon})^{\beta}, \tag{6}$$

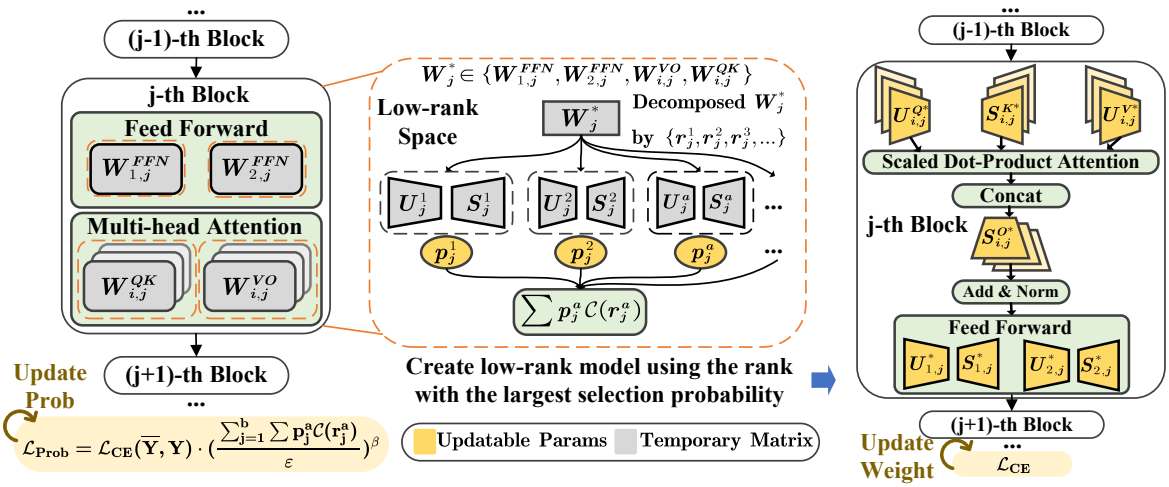

Figure 4. Compressing vision transformer using low-rank MHA layers and automatic rank selection.

where $\mathcal{L}_{CE}(\cdot)$ is the cross-entropy loss, $\overline{Y}$ is the final output of the entire model, Y is the ground truth, and β is the hyper-parameter controlling overall search process. Here as shown in Figure 4, the calculation of $\overline{Y} = Y_b$ is based on considering all the decomposition candidates (U_j^a, S_j^a) with different rank settings and their selection probabilities. After finishing the probability update, W_j^* is first factorized via using the rank that corresponds to the largest selection probability, and then updated via minimizing the cross-entropy loss with the frozen rank selection probabilities. The rank settings can be then finally determined after multiple rounds of such alternated update of probabilities and weights.

Low-Rank MHA for Personalized Text-to-Image Diffusion. Our proposed low-rank MHA can also be used for efficiently customizing text-to-image diffusion, an emerging computer vision task that the pre-trained diffusion model can quickly synthesize high-quality visual instantiations of user-defined concepts with few examples of images and guided text prompt. More specifically, given a pre-trained diffusion model $\{W_{diff}\}$ that has been well trained on image set $\{\mathbf{x}\}$ and the condition vector set $\{\mathbf{c}\}$ obtained from text prompt, we aim to minimize the following loss:

$$\mathbb{E}_{\epsilon,\mathbf{x_{new}},\mathbf{c_{new}},t}[w_t \|\{W_{new}\}(\alpha_t \mathbf{x_{new}} + \sigma_t \epsilon, \mathbf{c_{new}}) - \mathbf{x_{new}}\|_2^2], \tag{7}$$

where α_t, σ_t and w_t are the function of timestep t controlling diffusion process, and $\mathbf{x_{new}}$ and $\mathbf{c_{new}}$ denote the user-provided images and text prompts, respectively, with $|\{\mathbf{x_{new}}\}| \ll |\{\mathbf{x}\}|$ and $|\{\mathbf{c_{new}}\}| \ll |\{\mathbf{c}\}|$. Notice that here $\{W_{new}\}$ is initialized as $f(\{W_{diff}\})$, where $f(\cdot)$ can be either an identity function, meaning that the personalized model $\{W_{new}\}$ is directly initialized as the pre-trained $\{W_{diff}\}$, or a transformation function, indicating that the initialization for $\{W_{new}\}$ is the modification of $\{W_{diff}\}$.

As indicated in (Kumari et al., 2022), updating the entire $\{W_{new}\}$ is very computationally inefficient and easily causes overfitting, due to the large size of $\{W_{diff}\}$ and small size of $\{\mathbf{x_{new}}\}$. Inspired by the insights that 1) only changing a few parameters is sufficiently to make the diffusion models learn the user-defined concept (Kumari et al., 2022); and 2) adding the low-rank component is an efficient fine-tuning strategy for large-size language models (LLMs) in NLP tasks (Aghajanyan et al., 2020), we propose to use the low-rank MHA to improve the deployment efficiency of personalized text-to-image diffusion. Figure 5 illustrates the overall framework. More specifically, the MHA layer of the personalized model is initialized as:

$$\begin{aligned}\mathrm{MHA}(X_Q, X_K, X_V) = \\ \sum_{i=1}^{h} \mathrm{Softmax}\Big(\frac{X_Q(W_i^Q W_i^{K^T} + U_i^Q S_i^{K^T})X_K^T}{\sqrt{d_k}}\Big) \\ X_V(W_i^V W_i^O + (U_i^V S_i^O))\end{aligned} \tag{8}$$

where $\{W_i^Q, W_i^K, W_i^V, W_i^O\}$ are obtained from the pre-trained diffusion model $\{W_{diff}\}$, and $\{U_i^Q, S_i^K, U_i^V, S_i^O\}$ are the randomly initialized low-rank components. As shown in Figure 5, in the model customization process all the parameters of the pre-trained model $\{W_{diff}\}$ are frozen; while their computation follows the mechanism described in our proposed low-rank MHA. Meanwhile, the added low-rank component U_i^Q, S_i^K, U_i^V and S_i^O are updated to make W_{new} adapt for the user-provided new concepts.

4. Experiments

4.1. Image Classification on ImageNet-1K

Setting. We first validate our approach on the ImageNet-1K dataset (Deng et al., 2009) for the image classification task. The dataset includes 1.2M training images and 50K validation samples. We adopt the baseline models, *i.e.,* dense networks without compression, and training recipe from DeiT (Touvron et al., 2021) since they show promising results on training the transformer models by only using

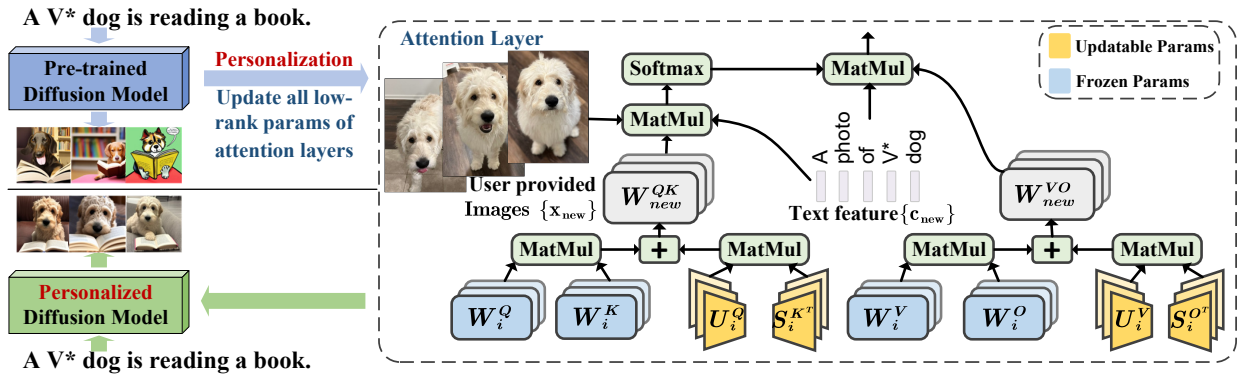

Figure 5. Customizing text-to-image diffusion model using low-rank MHA mechanism. Latent image feature $\mathbf{x_{new}}$ corresponds to X_Q and text feature $\mathbf{c_{new}}$ corresponds to X_K and X_V.

Table 2. Comparison between our method and various approaches, including model pruning, sparse training and token reduction, for compressing DeiT-small and DeiT-base on ImageNet.

Method	Compression	Top-1	FLOPs (↓%)	Params (↓%)
DeiT-small	Baseline	79.8	-	-
COMCAT (Ours)	Low-rank	79.27	**51.21**	**49.98**
COMCAT (Ours)	Low-rank	79.58	44.93	43.82
COMCAT (Ours)	Low-rank	**79.92**	41.15	40.11
UPop (Shi et al., 2023)	Pruning	79.6	39	39
UVC (Yu et al., 2022b)	Pruning	78.82	49.59	-
SCOP (Tang et al., 2020)	Pruning	77.5	43.6	-
S^2ViTE (Chen et al., 2021b)	Sparse	79.22	31.63	33.94
ToMe (Bolya et al., 2022)	Token	79.4	41.30	0
PS-ViT (Tang et al., 2022)	Token	79.4	43.5	0
HVT (Pan et al., 2021b)	Token	78.0	47.8	0
PoWER (Goyal et al., 2020)	Token	78.3	41.3	0
DeiT-base	Baseline	81.8	-	-
COMCAT (Ours)	Low-rank	**82.26**	**61.68**	**61.06**
CT-GFM (Yu & Wu, 2023)	Low-rank	81.28	-	40
MD-ViT (Hou & Kung, 2022)	Pruning	81.5	60	-
UVC (Yu et al., 2022b)	Pruning	80.57	54.5	-
VTP (Zhu et al., 2021)	Pruning	80.7	43.2	44.44
S^2ViTE (Chen et al., 2021b)	Sparse	82.22	33.13	34.41
PS-ViT (Tang et al., 2022)	Token	81.5	44.3	0
IA-RED^2 (Pan et al., 2021a)	Token	80.3	32.96	0

the ImageNet-1K without other large-scale datasets for pre-training and thus are widely adopted. We compare our approach with previous state-of-the-art ViT compression methods, including low-rank (Yu & Wu, 2023), model pruning (Hou & Kung, 2022; Yu et al., 2022b; Zhu et al., 2021; Tang et al., 2020), sparse training (Chen et al., 2021b) and token pruning (Bolya et al., 2022; Pan et al., 2021b; Tang et al., 2022; Goyal et al., 2020; Pan et al., 2021a).

Implementation Details. The whole process of our method consists of two steps: searching ranks and fine-tuning. We conduct the automatic rank selection algorithm to the pre-trained DeiT model to produce the low-rank model under the given constraint. In the fine-tuning process, the initial learning rate is set as 0.0001 and decreases to the minimum learning rate of 0.000001 with the Cosine scheduler. The weight decay for training the compressed DeiT-small is set as 0.005. The rest of the training hyper-parameters are consistent with DeiT (Touvron et al., 2021).

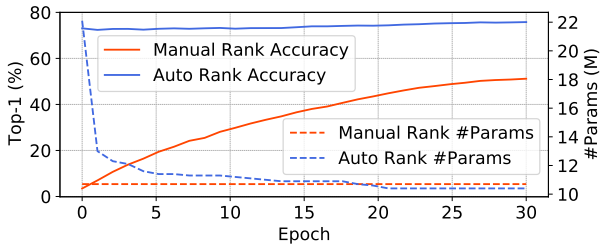

Figure 6. Comparison of the manual rank selection (red lines) and our proposed automatic rank selection method (blue lines) for DeiT-small on the ImageNet-1K dataset. We show the changes of the top-1 accuracy (solid lines) and the number of parameters (#Params, dashed lines) during training.

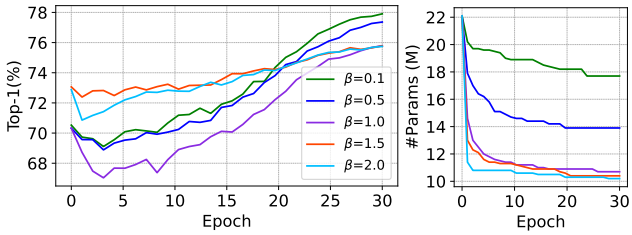

Figure 7. The effect of different beta values on the searching rank of the DeiT-small model.

Comparison Results. Table 2 shows the performance of different compression methods on the ImageNet dataset. Compared with the previous state-of-the-art automatic pruning method UVC (Yu et al., 2022b), our compressed models enjoy 0.45% and 1.69% higher top-1 accuracy with much larger FLOPs reduction on DeiT-small and DeiT-base, respectively. Additionally, our approach can significantly reduce the model parameters, *i.e.,* 49.98% reduction for DeiT-small and 61.06% for DeiT-base, while the compressed model from UVC (Yu et al., 2022b) can not. Compared with low-rank work of CT-GFM (Yu & Wu, 2023), our method achieves 0.98% accuracy increase with much fewer number of parameters. Compared with sparse training work of S^2ViTE (Chen et al., 2021b), with similar top-1 accuracy, we achieve much larger FLOPs reduction as 19.58% and 28.55% and parameters reduction as 16.04% and 26.65% on DeiT-small and DeiT-base, respectively. Compared with

Table 3. Measured speedup for the low-rank compressed DeiT-small and DeiT-base models on different computing platforms.

Model	#Params (M)	FLOPs (G)	Top-1 (%)	Throughput (images/s)				
				Nvidia V100	Snapdragon 855	Nvidia JetsonTX2	ASIC Eyeriss	FPGA
DeiT-small	21.96	4.24	**79.8**	974.46	7.26	27.37	24.38	4.02
COMCAT (Ours)	**10.98**	**2.07**	79.27	**1512.91**	**11.00**	**40.36**	**39.34**	**6.66**
DeiT-base	86.38	16.85	81.8	301.34	1.73	9.36	6.14	0.95
COMCAT (Ours)	**33.63**	**6.46**	**82.26**	**602.51**	**4.37**	**17.80**	**14.87**	**2.09**

Table 4. Top-1 accuracy and throughput (images/s) on GPU for the compressed DeiT-Small on ImageNet when compressing FFNs using different low-rank methods **(without using Fine-Tuning)**. It is seen that using SVD to compress FFN layers brings higher accuracy and throughput. Here the throughput is measured on Nvidia V100.

# of Params. in FFN ↓		**10%**	**20%**	**30%**	**40%**	**50%**
SVD	**Top-1 (%)**	**76.75**	**73.68**	**69.76**	**54.63**	**21.31**
	Throughput	**1017.06**	**1066.81**	**1129.56**	**1191.65**	**1245.19**
Tucker	**Top-1 (%)**	73.69	70.53	63.39	42.37	10.37
	Throughput	1017.06	1058.94	1083.99	1141.29	1213.87
Tensor Train	**Top-1 (%)**	0.81	0.53	0.35	0.16	0.14
	Throughput	617.47	639.70	675.39	714.79	772.87

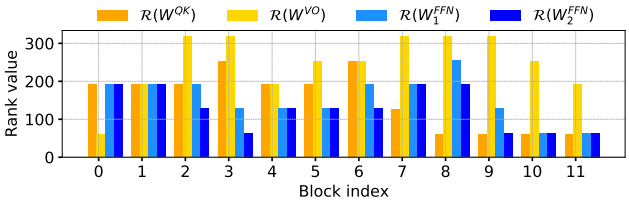

Figure 8. Rank distribution of DeiT-small model.

the work of PS-ViT (Tang et al., 2022) for token reduction, our method also achieves higher top-1 accuracy as 0.18% and 0.76% with much fewer FLOPs and model parameters on DeiT-small and DeiT-base, respectively.

Practical Speedups on Various Hardware Platforms. We further measure the practical speedups of our compressed models on various computing hardware platforms, including Nvidia Tesla V100, Nvidia Jetson TX2, Android mobile phone (Snapdragon 855, 4 Cortex-A76 + 4 Cortex-A55), ASIC accelerator Eyeriss (Chen et al., 2016), and FPGA (PYNQ Z1) in Table 3. Here the performance of Eyeriss is reported via using Timeloop (Parashar et al., 2019) with 45nm CMOS technology setting. Our compressed DeiT-small and DeiT-base models achieve significant speedups across different platforms. For example, on Snapdragon 855, our compressed DeiT-base obtains $2.52\times$ speedup than the baseline model with even higher top-1 accuracy on ImageNet. Such results demonstrate the practical effectiveness of our low-rank compression solution.

4.2. Ablation Analysis for ViT Low-Rank Compression

Automatic Rank Selection *vs.* Manual Rank Selection. To demonstrate the superiority of our automated rank selection method, we compare it with the fixed rank method. Figure 6 shows the variation curves of the top-1 accuracy on ImageNet-1K and the number of parameters for both methods during training. Our method has a smaller loss of accuracy, and the number of parameters of the model gradually converges to the target value (10.5M). In contrast, the model based on fixed-rank decomposition loses more accuracy from the beginning, resulting in poor performance. Therefore, we can conclude that our method can search for a better rank combination under the constraints.

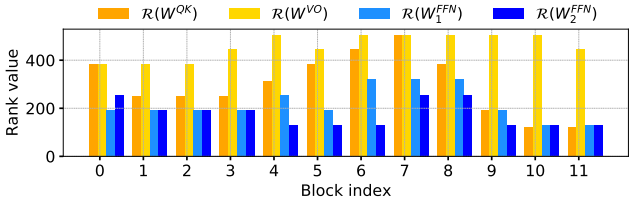

Figure 9. Rank distribution of DeiT-base model.

Hyper-Parameter for Searching Ranks. We also explore the effect of hyper-parameter β mentioned by Eq. 6 on searching rank process. Figure 7 shows the convergence of the searching process with respect to different β. It can be seen that when $\beta \geq 1$, the number of parameters of the model can converge to the target value quickly, and the final accuracy of the models is basically the same. However, when $\beta = 1.5$, the accuracy curve of the model is relatively smooth, therefore, we think 1.5 is a relatively better value for β. The final rank distribution of DeiT-small and DeiT-base are shown in Figure 8 and Figure 9.

SVD *vs.* Higher-order Tensor Decomposition. For ViT compression, in addition to the MHA, we apply the low-rank compression to the FFN (Feed-Forward Network), and the rank selection for FFN is also included in our proposed automatic rank determination mechanism. In order to find the optimal low-rank decomposition method from various low-rank decomposition methods such as SVD, Tucker decomposition, Tensor Train decomposition, we evaluate the Top-1 accuracy and throughput on GPU for the compressed DeiT-Small on ImageNet dataset when compressing FFNs using different low-rank methods. As shown in Table 4, with the same compression rate, using SVD brings higher accuracy (without fine-tuning) than using Tucker decomposition and Tensor Train decomposition with better throughput on GPU.

Table 5. Comparison of the training cost, *i.e.,* training time, GPU memory, and extra storage for each concept, and FID (Parmar et al., 2021) for various methods. Given a few images of a new concept, we generate images corresponding to the text prompt, *e.g.,* A photo of V * dog. The number in $(\cdot)$ denotes the number of images involved in training. "V*" is a unique identifier followed by the class name of the subject that is used to identify the object to be learned. Except for Goldendoodle, the rest of the images are from CustomDiffusion.

Method	Training Cost			FID							
	Training Time (s)	GPU Memory (MB)	Extra Storage (MB)*	Teddy Bear (7)	Tortoise Plushy (12)	Wooden Pot (4)	Barn (7)	Cat (5)	Dog (10)	Golden-doodle (4)	Average
COMCAT (Ours)	**193**	**11765**	**6**	76.61	**168.42**	91.13	**42.5**	139.51	**86.68**	**119.49**	**101.23**
CustomDiffusion	237	11807	75	93.92	196.15	136.76	52.23	**127.93**	146.14	150.28	128.06
DreamBooth	502	30979	11565	**62.25**	186.52	**89.31**	47.93	150.79	89.31	179.33	118.03

*The extra storage means that, given a pre-trained model, the extra storage requirement when the pre-trained model is further fine-tuned to adapt to a new customized concept. Notice that here we follow the same definition for extra storage cost used in Customize Diffusion (Kumari et al., 2022). That is, in customization scenario, because the pre-trained model needs to be always preserved for future more new concepts, any modification on the pre-trained model for the current new concept is viewed as extra storage cost. The amount of extra storage is obtained via direct measurement of file size.

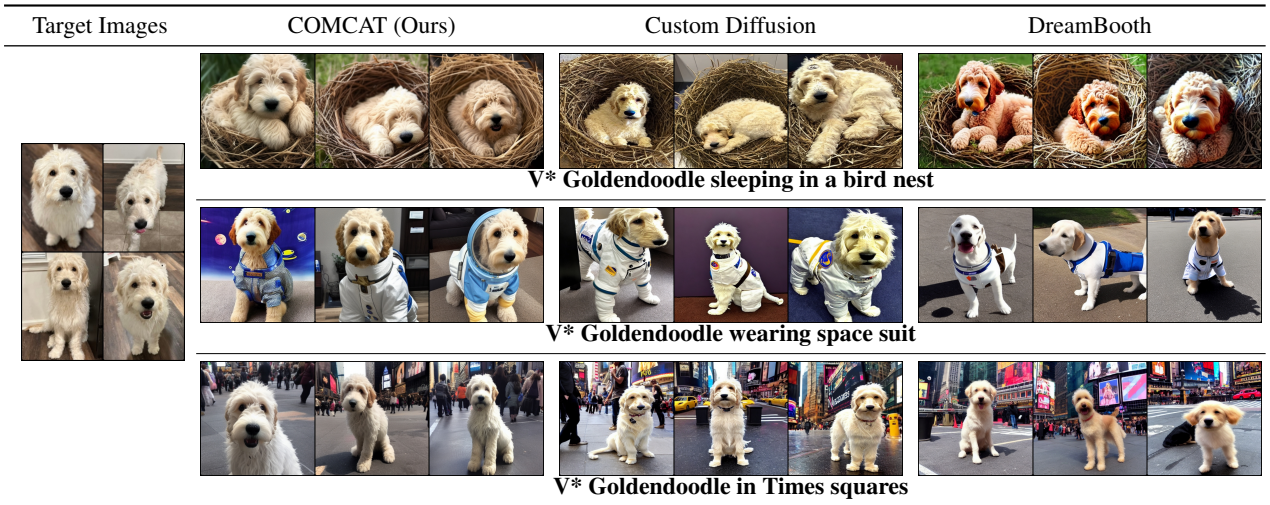

Figure 10. Fine-tuning results. All methods were trained on 1 A6000 GPU for 500 steps, and the training consumption is shown in Table 5. All images were generated with 50 steps of PNDM (Liu et al., 2022) sampler and guidance scale is 7.

Table 6. MS-COCO FID evaluation with fine-tuned models is a standard evaluation metric for text-to-image models. The pre-trained diffusion model has FID as 32.9 with the same settings of 50 PNDM (Liu et al., 2022) sampling steps and scale as 6.

	Pre-trained	Goldendoodle	Barn	Cat
MS-COCO FID	32.90	32.84	31.88	30.98
	Dog	TeddyBear	TortoisePlushy	WoodenPot
MS-COCO FID	30.85	30.89	32.30	31.29

4.3. Personalized Text-to-Image Diffusion Models

Setting. We fine-tune the *cross-attention* layer of the pre-trained Stable Diffusion (Rombach et al., 2022) (model weights obtained from HuggingFace Hub[1] (Wolf et al., 2019)) by using our proposed low-rank MHA mechanism to enable the model to learn a new concept. For quantitative evaluation, we use the six objects included in the dataset released by CustomDiffusion (Kumari et al., 2022) and one newly collected object, with the number of images contained in each one ranging from 4 to 12.

Training Cost Comparison. We first present the training cost required by all approaches in Table 5. We train all the approaches on one Nvidia RTX A6000 GPU with the batch size as 1 and the number of training steps as 500. Compared with CustomDiffusion and DreamBooth, our approach reduces the training time by 18.6% and 61.6%, respectively (see Training Time in Table 5), and decreases the extra storage space for each concept by 12.5× and 1927.5×, respectively (see Extra Storage in Table 5). The order of magnitude reduction of extra storage for each new concept is extremely important for the broad adoption of personalized text-to-image diffusion models, where users can prepare their diffusion models without the burden of model storage.

Image Quality Comparison. We then evaluate the quality of the synthesized images for all approaches. We generate 20 images for each learned target image (concept) by using the same text prompt for all approaches and calculate the FID (Heusel et al., 2017; Parmar et al., 2021) between the

[1] https://huggingface.co/CompVis/stable-diffusion-v1-4

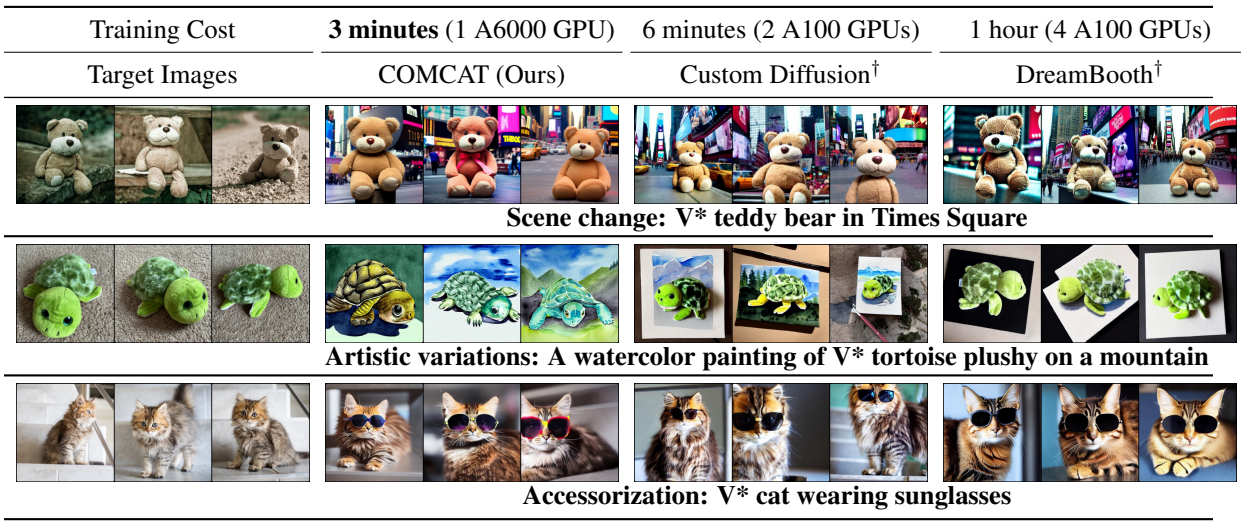

Figure 11. Fine-tuning results, where the results of CustomDiffusion† and DreamBooth† are from Custom Diffusion (Kumari et al., 2022).

synthesized and real images. The lower FID score indicates the smaller difference between the generated and the real images. As shown in Table 5, our approach achieves the lowest FID than the existing works. We further provide the qualitative comparison in Figure 10. In addition to generating the corresponding scenes accurately from text prompts, the *V* Goldendoodle* generated by our method is the closest to the real image, while the *V* Goldendoodle* synthesized by DreamBooth (Ruiz et al., 2022) has obvious differences from the real one, and images synthesized from CustomDiffusion (Kumari et al., 2022) contain less natural mouth as it has been deformed. In Figure 11, we further show that even with much fewer computation resources, *i.e.,* less training time and fewer number of GPUs for model fine-tuning, we can still generate high-quality images.

MS-COCO Evaluation. Lastly, we perform the experiments to understand if the fine-tuned models can generate images that are unrelated to the learned target subject (*V**). We use the prompted text of $5,000$ images from the MS-COCO 2017 (Lin et al., 2014) validation set to generate images and calculate the FID. As shown in Table 6, the FID from the personalized models are similar to the pre-trained text-to-image model, indicating that they can synthesize high-quality images for unrelated concepts. Thus, the model fine-tuned by our method still holds the distribution of the synthesized images as the pre-trained model.

5. Conclusion

This paper fundamentally investigates the low-rankness in the multi-ahead attention layer of the emerging vision models and proposes that the head-level low-rankness should be explored for efficient model design, bringing highly efficient low-rank ViT compression solution. Our method not only outperforms existing compression approaches by providing higher performance but also brings faster practical speedup. Particularly, our finding is further applied for efficient customization of text-to-image diffusion models, outperforming the state-of-the-art solutions.

6. Acknowledgements

This work was partially supported by National Science Foundation under Grant CCF-1937403 and CCF-1955909.

References

Aghajanyan, A., Zettlemoyer, L., and Gupta, S. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*, 2020.

Bolya, D., Fu, C.-Y., Dai, X., Zhang, P., Feichtenhofer, C., and Hoffman, J. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *European conference on computer vision*, pp. 213–229. Springer, 2020.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Chen, B., Dao, T., Winsor, E., Song, Z., Rudra, A., and Ré, C. Scatterbrain: Unifying sparse and low-rank attention. *Advances in Neural Information Processing Systems*, 34: 17413–17426, 2021a.

Chen, T., Cheng, Y., Gan, Z., Yuan, L., Zhang, L., and Wang, Z. Chasing sparsity in vision transformers: An end-to-end exploration. *Advances in Neural Information Processing Systems*, 34:19974–19988, 2021b.

Chen, Y.-H., Emer, J., and Sze, V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. *ACM SIGARCH Computer Architecture News*, 44(3):367–379, 2016.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-Or, D. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

Goyal, S., Choudhury, A. R., Raje, S., Chakaravarthy, V., Sabharwal, Y., and Verma, A. Power-bert: Accelerating bert inference via progressive word-vector elimination. In *International Conference on Machine Learning*, pp. 3690–3699. PMLR, 2020.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hou, Z. and Kung, S.-Y. Multi-dimensional vision transformer compression via dependency guided gaussian process search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3669–3678, 2022.

Hsu, Y.-C., Hua, T., Chang, S., Lou, Q., Shen, Y., and Jin, H. Language model compression with weighted low-rank factorization. *arXiv preprint arXiv:2207.00112*, 2022.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

Kim, Y.-D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530*, 2015.

Kumari, N., Zhang, B., Zhang, R., Shechtman, E., and Zhu, J.-Y. Multi-concept customization of text-to-image diffusion. *arXiv preprint arXiv:2212.04488*, 2022.

Li, Y., Yuan, G., Wen, Y., Hu, J., Evangelidis, G., Tulyakov, S., Wang, Y., and Ren, J. Efficientformer: Vision transformers at mobilenet speed. *arXiv preprint arXiv:2206.01191*, 2022.

Liebenwein, L., Maalouf, A., Feldman, D., and Rus, D. Compressing neural networks: Towards determining the optimal layer-wise decomposition. *Advances in Neural Information Processing Systems*, 34:5328–5344, 2021.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Liu, H., Simonyan, K., and Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

Liu, L., Ren, Y., Lin, Z., and Zhao, Z. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021a.

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021b.

Noach, M. B. and Goldberg, Y. Compressing pre-trained language models by matrix decomposition. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 884–889, 2020.

Pan, B., Panda, R., Jiang, Y., Wang, Z., Feris, R., and Oliva, A. Ia-red ^{2}: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021a.

Pan, Z., Zhuang, B., Liu, J., He, H., and Cai, J. Scalable vision transformers with hierarchical pooling. In *Proceedings of the IEEE/cvf international conference on computer vision*, pp. 377–386, 2021b.

Parashar, A., Raina, P., Shao, Y. S., Chen, Y.-H., Ying, V. A., Mukkara, A., Venkatesan, R., Khailany, B., Keckler, S. W., and Emer, J. Timeloop: A systematic approach to dnn accelerator evaluation. In *2019 IEEE international symposium on performance analysis of systems and software (ISPASS)*, pp. 304–315. IEEE, 2019.

Parmar, G., Zhang, R., and Zhu, J.-Y. On buggy resizing libraries and surprising subtleties in fid calculation. *arXiv preprint arXiv:2104.11222*, 2021.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Ren, Y., Wang, B., Shang, L., Jiang, X., and Liu, Q. Exploring extreme parameter compression for pre-trained language models. *arXiv preprint arXiv:2205.10036*, 2022.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. *arXiv preprint arXiv:2208.12242*, 2022.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E., Ghasemipour, S. K. S., Ayan, B. K., Mahdavi, S. S., Lopes, R. G., et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Shi, D., Tao, C., Jin, Y., Yang, Z., Yuan, C., and Wang, J. Upop: Unified and progressive pruning for compressing vision-language transformers. *arXiv preprint arXiv:2301.13741*, 2023.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pp. 6105–6114. PMLR, 2019.

Tang, Y., Wang, Y., Xu, Y., Tao, D., Xu, C., Xu, C., and Xu, C. Scop: Scientific control for reliable neural network pruning. *Advances in Neural Information Processing Systems*, 33:10936–10947, 2020.

Tang, Y., Han, K., Wang, Y., Xu, C., Guo, J., Xu, C., and Tao, D. Patch slimming for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12165–12174, 2022.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., and Keutzer, K. Fbnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10734–10742, 2019.

Xiang, L., Yin, M., Zhang, C., Sukumaran-Rajam, A., Sadayappan, P., Yuan, B., and Tao, D. Tdc: Towards extremely efficient cnns on gpus via hardware-aware tucker decomposition. In *Proceedings of the 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, pp. 260–273, 2023.

Xiao, J., Zhang, C., Gong, Y., Yin, M., Sui, Y., Xiang, L., Tao, D., and Yuan, B. Haloc: Hardware-aware automatic low-rank compression for compact neural networks. 2023.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021.

Yin, M., Sui, Y., Liao, S., and Yuan, B. Towards efficient tensor decomposition-based dnn model compression with optimization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10683, June 2021.

Yin, M., Phan, H., Zang, X., Liao, S., and Yuan, B. Batude: Budget-aware neural network compression based on tucker decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8874–8882, 2022a.

Yin, M., Sui, Y., Yang, W., Zang, X., Gong, Y., and Yuan, B. Hodec: Towards efficient high-order decomposed convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12308, 2022b.

Yin, M., Uzkent, B., Shen, Y., Jin, H., and Yuan, B. Gohsp: A unified framework of graph and optimization-based heterogeneous structured pruning for vision transformer. *arXiv preprint arXiv:2301.05345*, 2023.

Yu, H. and Wu, J. Compressing transformers: Features are low-rank, but weights are not! 2023.

Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022a.

Yu, S., Chen, T., Shen, J., Yuan, H., Tan, J., Yang, S., Liu, J., and Wang, Z. Unified visual transformer compression. *arXiv preprint arXiv:2203.08243*, 2022b.

Zhu, M., Tang, Y., and Han, K. Vision transformer pruning. *arXiv preprint arXiv:2104.08500*, 2021.