# DnA: Improving Few-shot Transfer Learning with Low-Rank Decomposition and Alignment

Ziyu Jiang[1,2†], Tianlong Chen[3], Xuxi Chen[3], Yu Cheng[1], Luowei Zhou[1], Lu Yuan[1], Ahmed Awadallah[1], and Zhangyang Wang[3‡]

[1] Microsoft Corporation
[2] Texas A&M University
[3] University of Texas at Austin

**Abstract.** Self-supervised (SS) learning has achieved remarkable success in learning strong representation for in-domain few-shot and semi-supervised tasks. However, when transferring such representations to downstream tasks with domain shifts, the performance degrades compared to its supervised counterpart, especially at the few-shot regime. In this paper, we proposed to boost the transferability of the self-supervised pre-trained models on cross-domain tasks via a novel self-supervised alignment step on the target domain using only unlabeled data before conducting the downstream supervised fine-tuning. A new reparameterization of the pre-trained weights is also presented to mitigate the potential catastrophic forgetting during the alignment step. It involves low-rank and sparse decomposition, that can elegantly balance between preserving the source domain knowledge without forgetting (via fixing the low-rank subspace), and the extra flexibility to absorb the new out-of-the-domain knowledge (via freeing the sparse residual). Our resultant framework, termed Decomposition-and-Alignment (**DnA**), significantly improves the few-shot transfer performance of the SS pre-trained model to downstream tasks with domain gaps. [4]

**Keywords:** Self-supervised Learning; Transfer Few-shot; Low-Rank

## 1 Introduction

Employing Self-Supervised (SS) models pre-trained on large datasets for boosting downstream tasks performance has become de-facto for many applications [10], given it could save the expensive annotation cost and yield strong performance boosting for downstream tasks [6, 17, 8]. Recent advance in the SS pre-training method points out its potential on surpassing its supervised counterpart for few-shot and semi-supervised downstream tasks [39, 7].

For the transfer learning of SS models, most previous works followed the many-shot setting [6, 17, 14]. However, recent discoveries state that when the target domain has a domain gap with the source data and it has only limited label

[†] Work done during an intership at Microsoft Corporation
[‡] Correspondence to: Zhangyang Wang (atlaswang@utexas.edu)
[4] The code is released at https://github.com/VITA-Group/DnA

samples, the transferability of SS models is still inferior to its supervised counter part [28]. The authors argued that comparing to SS pre-training, supervised pre-training encourages the learned representation to be more compactly distributed, and the label supervision also enforces stronger alignment across different images. As a result, the supervised representations display better clustering properties on the target data, facilitating the few-shot learning of classifier boundaries. The authors thus proposed to progressively sample and mix the unlabeled target data into the unsupervised pretraining stage, for several rounds. Such "target-aware" unsupervised pretraining (TUP) improves few-shot SS transfer performance. Yet it would be too expensive if we re-conduct pre-training (even in part) for every downstream purpose. Moreover, when pre-training is conducted on privileged data that is inaccessible to downstream users, the above solution will also become practically infeasible.

As previous findings reveal the few-shot transfer performance degradation due to the source-target domain discrepancy, in this paper, we are inspired to boost the transferability through a *self-supervised domain adaptation* perspective. Following the assumption of [28, 32] that a small target dataset can be available, instead of "mixing" it with the pre-training data, we "fine-tune" the general pre-trained model with the small target data, under self-supervision. This extra step between pre-training and downstream fine-tuning, called **Alignment**, incurs a much smaller overhead compared to re-conducting pre-training on the mixed data [28], and avoids accessing the pre-training data.

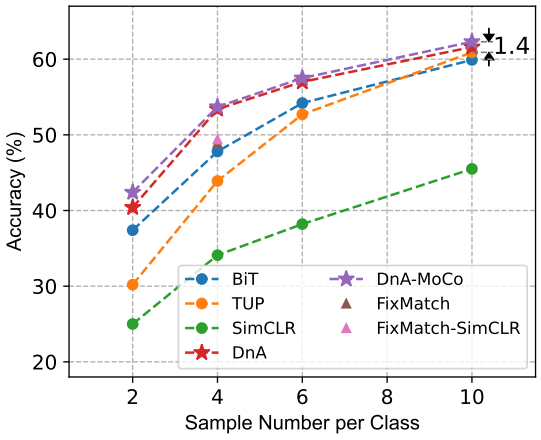



Fig. 1: Comparison with State-Of-The-Art (SOTA) methods on few-shot transfer tasks (source: ImageNet-1k; target: CIFAR100, with different labeled sample numbers per class). While the SimCLR performance struggles when directly transferring to cross-domain down-stream tasks, the proposed DnA method (implemented on top of SimCLR backbone) can significantly improve it by a large margin (>14%). DnA also remarkably surpasses the previous SOTAs (FixMatch [35], BiT [26] and TUP [28]) when combining with MoCo (DnA-MoCo) by at least 1.4%. FixMatch-SimCLR denotes FixMatch initilized from SimCLR pre-training.

One specific challenge arising from the alignment step is the potential catastrophic forgetting of the pre-training knowledge [25]. To mitigate this risk, we introduce a **Decomposition** of the pre-trained weights before the alignment step, which involves no re-training. Specifically, we re-parameterize

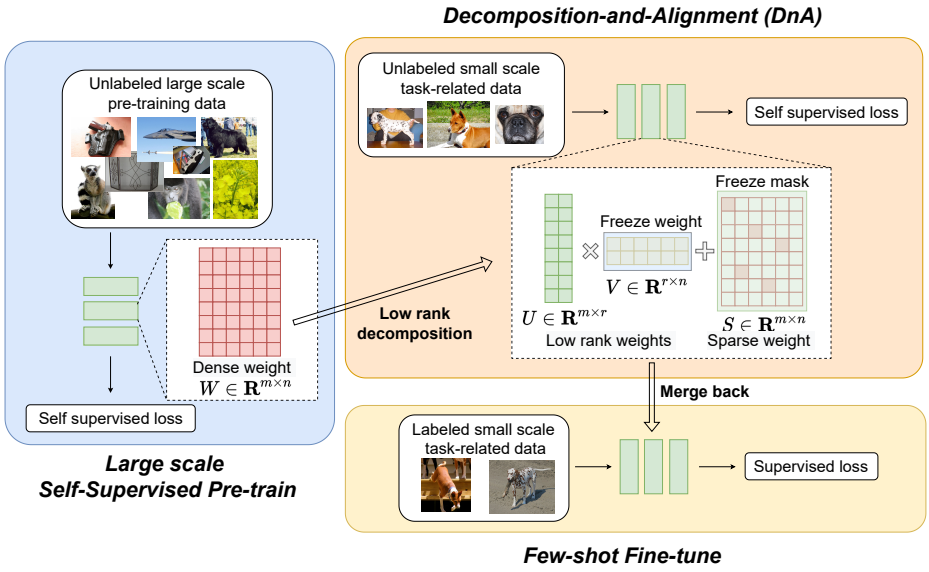



Fig. 2: The overview of the proposed DnA framework. It is applied on top of any self-supervised pre-trained model, to boost its few-shot transfer performance for the downstream tasks on the target data with a domain shift from the pre-training source data.

the pre-trained weight into the sum of the low-rank term (involving the produce of two matrix factors), and a sparse residual term. That is inspired by the findings that big pre-trained models have a low "intrinsic dimension" [1, 20]. Then during the alignment, we freeze the low-rank subspace (but different subspace dimensions can be reweighted) in order to preserve the "in-domain" pre-training knowledge, while allowing the sparse residual to freely change for encapsulating the "out-of-domain" target knowledge. Although low-rank and sparse decomposition is a canonical idea [55, 3, 47], this is its first time to be connected the large model fine-tuning, to our best knowledge.

Our contributions can be summarized as following:

- We present a simple and effective self-supervised *alignment* method for mitigating the domain gaps between pre-training and downstream transfer, in order to enhance the few-shot transferability of self-supervised pre-trained models, without the (expensive and often infeasible) re-training with the source domain data.
- We further present a novel *decomposition* of the pre-trained weights, to mitigate the potential catastrophic forgetting during the alignment. It draws inspirations from the classical low rank and sparse decomposition algorithm, and gracefully balances between preserving pre-training knowledge (through low-rank) and absorbing new target knowledge (through sparse term).

– The overall framework, named *Decomposition-and-Alignment* (**DnA**), demonstrates highly competitive performance over challenging few-shot transfer benchmarks. It improves the pre-trained SimCLR model and outperforms the latest state-of-the-arts (SOTA) [35, 26, 28]: see Figure 1 for an example.

## 2 Related works

**Self-Supervised Learning:** Given the expensive cost of label annotation, SS learning from unlabeled data has received much attention. Earlier SS learning methods employs proxy task like colorization [49], jigsaw [31], rotation [13], selfie [40]. Recently, contrastive learning becomes the most popular SS regime because of its strong performance [6, 17, 14]. SS pre-training can significantly boost various downstream tasks [39, 42, 45, 5, 24]; for semi-supervised learning, it often leads to SOTA performance [39, 7, 48, 36].

**Transferability of Self-Supervised Models:** For transferring SS models, most previous works studied the many-shot benchmarks [6, 17, 14]. The more practical yet challenging few-shot transferability of SS models is under-explored. Pioneer works investigated the effect of downstream few-shot transfer learning for different pre-training opinions [11, 22]. [22] reveals the improvement room of SS transferability via combining supervised learning, but it can only work on labeled data. TUP [28] explores to improve few-shot transferability via minimizing the domain discrepancy between pre-training and downstream tasks. It significantly boosts few-shot transferability by mixing the pre-training dataset with small-scale unlabeled samples progressively acquired from the target domain.

**Low Rank and Sparsity in Deep Networks:** Low-rank has been studied with a long history in deep networks [23, 33, 37, 34, 50, 53], for multiple contexts including model compression, multi-task learning and efficient training. Recent literature [1] reveals that the low-rank structure exists in the pre-trained model, which motivates the parameter efficient tuning [20]. The same prolific research can be found in the field of sparsity for deep networks, which is perhaps best known as a model compression means [16, 12]. Sparsity also effectively regularizes few-shot learning [56, 51, 4]; and naturally emerges during fine-tuning [15, 52]. We note that the current fine-tuning works relying on either low rank or sparsity [15, 52, 20] are all in the natural language processing (NLP) domain, and none of them operates in the few-shot setting with domain shifts.

The marriage of low rank and sparsity is well known as the robust principal component analysis (RPCA) algorithm [55, 3]. In deep networks, the most relevant work to this idea is perhaps [47], which reconstructed the weight matrices by using sparse plus low-rank approximation, for model compression - an orthogonal purpose to ours. Other applications include combining those two priors in deep compressive sensing [21]. To the best of our knowledge, no previous work has linked low rank and sparse decomposition to transfer learning.

# 3 Method

## 3.1 Overview

In this paper, we employ SimCLR [6] as a strong SS pre-training backbone. SimCLR [6] learns visual representation via enforcing the consistency between different augmented views while enlarging the difference from other samples. Formally, the loss of SimCLR is

$$\mathcal{L}_{\mathrm{CL,i}} = -\log \frac{s^{\tau}\left(v_i^1, v_i^2\right)}{s^{\tau}\left(v_i^1, v_i^2\right) + \sum_{v^- \in V} s^{\tau}\left(v_i^1, v^-\right)} \tag{1}$$

where $v_i^1$ and $v_i^2$ are the normalized features of two augmented views for the same image, while $V$ is the set of negative samples for $i$th image, which is composed by the features for other images in the same batch. All features are calculated sequentially with the feature encoder and projection head. $s^{\tau}$ is the feature similarity function with temperature $\tau$ that can be formalized as

$$s^{\tau}\left(v_i^1, v_i^2\right) = \exp(\frac{v_i^1 \cdot v_i^2}{\tau}) \tag{2}$$

As an overview, our holistic framework is demonstrated in Figure 2: DnA first decomposes the SS pre-trained weight to the low-rank terms ($U$ and $V$) and sparse term $S$. Afterwards, the model is aligned by self-supervised tuning over small target domain data, by fixing $V$ while tuning $U$ and $S$ in this step. The aligned model then goes through the typical supervised fine-tuning for the downstream few-shot task. Below we present step-by-step method details.

## 3.2 Basic Alignment Step

In this work, we leverage unlabeled training data from the downstream dataset , yet avoiding the (expensive and often infeasible) re-pre-training for every downstream transfer, and assuming no access to the pre-training data. To reduce the discrepancy between source and target domains, we design the extra **Alignment** step between pre-training and fine-tuning: we continue to tune the pre-trained model on a small-scale unlabeled dataset from the target domain, with self-supervised loss (here we use the same loss of SimCLR). Note that we follow [28] to assume that small-scale unlabeled data from the domain of the target few-shot task is available. (e.g., the unlabeled training set of the downstream dataset.)

The proposed alignment step is rather simple and efficient. Notably, it only requires the pre-trained model but not the pre-training dataset, which we believe is a more practical setting. Perhaps surprisingly, we observe that this vanilla alignment already suffices to outperform TUP [28] in some experiments.

**Challenge:** However, the alignment might run into the risk of *catastrophic forgetting* of the pre-training knowledge, due to tuning with only the target domain data. That will also damage the transferred model's generalization. We started by trying off-the-shelf learning-without-forgetting strategies, such as enforcing the $\ell_2$ norm similarity between the pre-trained model with the aligned

model weights [44]. That indeed yields empirical performance improvements, but mostly only marginal. We are hence motivated to look into more effective mitigation for the fine-tuning scenarios.

### 3.3 Decomposition before Alignment

Inspired by the findings that the weight of the pre-trained model resides with a low "intrinsic dimension" [1, 20], we propose to leverage the low-rank subspace assumption to effectively "lock in" the pre-training knowledge, with certain flexibility to adjust the subspace coefficients. However, the low-rank structure alone might be too restricted to learn the target domain knowledge that might lie out of this low-rank subspace, and we extend another sparse residual term to absorb such. Adopting the decomposed weight form for alignment is hence assumed to well balance between memorizing the source-domain knowledge and flexibly accommodating the new out-of-the-domain knowledge.

This idea of combining low rank and sparsity is a canonical one [3, 55], yet has not been introduced to fine-tuning before. In the following sections, we first introduce how to decompose the weight in Section 3.3. Then, we discuss the details of applying in Section 3.3.

**Low rank and Sparse Weight Decomposition** For a convolutional or a fully connected layer, the forward process can be formalized as

$$\mathbf{y} = W\mathbf{x} \tag{3}$$

where $\mathbf{y} \in \mathbf{R}^m$ is the output, $W \in \mathbf{R}^{m\times k}$ is the weight matrix, and $\mathbf{x} \in \mathbf{R}^k$ is the input. It is worth noting that, for the convolutional layer, we follow [38] to reshape the 4D convolutional kernel $W \in \mathbf{R}^{C_{in}\times H\times W\times C_{out}}$ to a 2D matrix $W \in \mathbf{R}^{(C_{in}H)\times(WC_{out})}$.

The resultant 2D weight $W$ can be decomposed as

$$W = UV + S \tag{4}$$

where $U \in \mathbf{R}^{m\times r}$ and $V \in \mathbf{R}^{r\times k}$ are two low rank matrix factors with $r < \min(m, k)$, and together denotes the low-rank subspace component of the pre-trained weight. Meanwhile $S$ denotes the sparse residual.

The low rank and sparse decomposition were found to well capture the long-tail structure of trained weights [47]. To solve this decomposition, we resort to the fast and data-free matrix decomposition method called GreBsmo [54]. GreBsmo formalizes the low rank decomposition as:

$$\begin{aligned} &\min_{U,V,S} \|W - UV - S\|_F^2 \\ &\text{s.t. } \operatorname{rank}(U) = \operatorname{rank}(V) \le r, P_\Omega S = 0 \end{aligned} \tag{5}$$

where $1 - \mathcal{P}_\Omega$ is the sparse mask of $S$, defined as

$$\mathcal{P}_\Omega = \begin{cases} 1, (i,j) \in \Omega \\ 0, (i,j) \in \Omega^C \end{cases} \tag{6}$$

where $\Omega$ denotes the set of points that has value, $(i, j)$ indicates the coordinate of a point in 2d weight matrix. GreBsmo then solves this optimization with an alternative updating algorithm (its derivation can be found at [54]) as:

$$\begin{cases} U_k = Q, D_{\mathrm{QR}}\left(\left(W - S_{k-1}\right) V_{k-1}^T\right) = QR \\ V_k = Q^T \left(W - S_{k-1}\right) \\ S_k = \mathcal{S}_\lambda \left(W - U_k V_k\right) \end{cases} \tag{7}$$

where $D_{\mathrm{QR}}$ is the fast QR decomposition algorithm to generate two inter-media matrix $Q$ and $R$. The subscription $k$ of $U$, $V$, $S$ indicates the $U$, $V$, $S$ in the $k$th iteration. $\mathcal{S}_\lambda$ is an element-wise soft threshold function formally as:

$$\mathcal{S}_\lambda W = \{\operatorname{sgn}\left(W_{ij}\right) \max\left(|W_{ij}| - \lambda, 0\right) : (i, j) \in [m] \times [k]\} \tag{8}$$

where $\operatorname{sgn} x$ would output the sign for $x$.

Practically, the decomposed weight would inevitably have a slight loss $E = W - UV - S$ compared to the origin pre-trained weight, due to the limited optimization precision. Such difference is usually very small at each layer, but might be amplified during the forward pass. To mitigate that, we treat the $E$ term as a fixed bias for each layer, and never tune it during the alignment step.

**How to Align over the Decomposed Weights** After decomposition, we freeze the low-rank subspace $V$ as the "fixed support" from the pre-trained weight. For flexibility, we consider $U$ as representation coefficients over this subspace, and allow that to be "alignable" (i.e, we allow for "re-composing" the existing knowledge). Meanwhile, $S$ represents the out-of-the-subspace component and is always set to be "alignable" for new target knowledge.

Hence, the alignment step over the decomposed weights could be represented as:

$$\begin{aligned} W' =& (U + \Delta U)V + S + \Delta S \\ \text{s.t. } & \operatorname{rank}(U) = \operatorname{rank}(V) \leq r \\ & P_\Omega S = 0, P_\Omega \Delta S = 0 \end{aligned} \tag{9}$$

where $\Delta U$ and $\Delta S$ are the two variables that will be tuned in the alignment step; the original $U, V, S$ are all fixed for alignment meanwhile. Note that $\Delta U$ has the same dimension as the low-rank factor $U$, and $\Delta S$ has the same sparse support (i.e., locations of non-zero elements) as $S$.

After the alignment step, we re-combine the decomposed form into one weight matrix $W$, and proceed to the fine-tuning step as normal.

## 4 Experiments

### 4.1 Settings

**Datasets:** To evaluate the proposed method, we exploit the following datasets: iNaturalist [41], CIFAR100 [27], EuroSAT [19], Food101 [2]. Particularly, we

adopt a random sampled 1000 classes subset of iNaturalist (Denoted as iNaturalist-1k) to validate the proposed method's performance on an imbalanced dataset. As shown in Table 1, these datasets are different in terms of resolutions (ranging from 32-224) and the number of classes (ranging from 10-1000). In addition, we consider both general classification tasks (e.g., CIFAR100) and fine-grain classification tasks (e.g., iNaturalist-1k, Food101). The imbalanced dataset is also included (e.g., iNaturalist-1k). The datasets also have different levels of similarity to the ImageNet (ranging from natural images like iNaturalist to the satellite images like EuroSAT) . These differences indicate the chosen datasets can represent practical cases. We report the Top1 accuracy for all datasets. For all datasets, we upsample/downsample it to $224 \times 224$ following the common practice of applying ImageNet pre-trained model [28, 6]. We follow [28] and assume that the available unlabeled dataset is the full training set for each dataset.

**Training settings:** For all the experiments, we employ the network architecture of Resnet-50 [18]. When conducting SS, the fully connected layer is replaced with a two-layer projection head as in SimCLR [6]. For SS pre-trained model, we employ the official SimCLR and Moco model pre-trained on ImageNet-1k for 800 epochs.

Table 1: Summary of the datasets employed in this work for resolution, number of train images (#train images), and number of categories (#Categories).

| Dataset | Resolution | #Train images | #Categories |
|---|---|---|---|
| iNaturalist-1k | >224×224 | 51984 | 1000 |
| CIFAR100 | 32×32 | 50000 | 100 |
| EuroSAT | 64×64 | 19000 | 10 |
| Food101 | >224×224 | 75750 | 101 |

For all DnA experiments, we follow SimCLR to use LARS optimizer [46], augmentation settings and cosine learning rate decay. We employ a batch size of 256 instead of larger batch size for ensuring the proposed method's practicability. On iNaturalist-1K, we train 200 epochs for ablation study. On other datasets, We train the model for 800 epochs following [28]. The initial learning rate is set as 0.2. For few-shot fine-tuning, we follow the setting of SimCLR-v2 [7] with LARS optimizer, cosine learning rate decay, and tune from the first layer of the projection head for 200 epochs.

We also consider Linear Probing (LP) performance for SimCLR, where a linear classifier is trained on the frozen features, following the setting of [39].

**Configuration for the GreBsmo algorithm**: To improve the practical speed, GreBsmo invokes a greedy rank $r$ for both $U$ and $V$. It starts from a very small rank of $r_0$ for $V$, Then $V \in \mathbf{R}^{r_0 \times k}$ iterates Equation 7 for $M$ times. Afterwards, the rank of $V$ would increase to $r_1 = r_0 + \Delta r$ by adding $\Delta r$ extra rows to $V$, where $\Delta r$ is the rank step size. The added $\Delta r$ rows are selected greedily as the top $\Delta r$ row basis that can minimize the objective in Equation 5, which are obtained with a small SVD [54].

Benefiting from the aforementioned, the weight decomposition for the entire ResNet50 can be done in less than one hour on a single 1080 Ti, which is a

small time overhead compared to either pre-training or finetuning. Moreover, the decomposition can be reused among different downstream datasets, further amortizing the computation overhead.

Empirically, we initialize with $U_0 = 0$, $V_0 \sim \mathcal{N}(0, 0.02)$ $S_0 = 0$. $r_0$, $\Delta r$ set as 1 and iterations $M$ is set as 100. As Resnet-50 [18] has a large variety for the weight dimension for different layers, we thus set the rank adaptively with the size of weight matrix as $r = \min(\lceil \alpha_r \cdot m \rceil, k)$, $\alpha_r$ is the rank ratio by default set as 0.25. The soft sparsity threshold $\lambda$ is by default 0.2. For the stability of training, we further fix sparsity to 99.7% after decomposing via assigning the top 0.3% large magnitude parameters in $W - UV$ as $S$.

### 4.2 DnA improves few-shot performance

**DnA improves the transfer few-shot performance.** To verify the effectiveness of the proposed methods, we study each component at iNaturalist-1k in terms of the 5-shot accuracy. As illustrated in Table 2a, the model without any pre-training yields a performance of 12.7%. By leveraging the SimCLR model pre-trained on ImageNet, the performance increases to 45.0%. Further, the proposed Align can significantly improve the performance by 1.4%. Finally, when combining the weight decomposition with the Align, the resultant DnA can further improve the accuracy over Align by an obvious margin of 0.9%.

**DnA surpasses the semi-supervised methods.** For the proposed DnA framework, we assume the existence of a small scale unlabeled dataset following [28]. This setting is close to semi-supervised learning, which has been widely studied in previous works [35, 43, 7]. Therefore, we further conduct experiments to compare our methods against well-established semi-supervised methods. Here, we exploit the SOTA semi-supervised strategy introduced in SimCLR-v2 [7] for

Table 2: (a) The few-shot and semi-supervised performance on iNaturalist-1k under the 5-shot setting. We employ the *distillation with unlabeled examples* method introduced in [7] as the semi-supervised method. (b) Comparing with FixMatch [35] on CIFAR100 4-shot in terms of semi-supervised performance.

| Pre-train | Semi-supervised | Accuracy |
|---|---|---|
| None | | 12.7 |
| | ✓ | 13.0 |
| SimCLR | | 45.0 |
| | ✓ | 46.3 |
| Aligned (Ours) | | 47.4 |
| | ✓ | 48.5 |
| DnA (Ours) | | 48.3 |
| | ✓ | 49.6 |

(a)

| Pre-train | Method | Accuracy |
|---|---|---|
| None | FixMatch | 48.9 |
| SimCLR | FixMatch | 49.4 |
| SimCLR | DnA (ours) | 53.4 |
| SimCLR | DnA+Semi (ours) | 56.4 |

(b)

comparison. As illustrated in Table 2a, when employing the semi-supervised method, the performance consistently improves compared to the corresponding few-shot performance. However, the improvement is marginal when the few-shot performance is not optimized (e.g. when the pre-training is not applied). While semi-supervised learning can improve the SimCLR model by 1.3%, it is still inferior by 1.1% compared to the Aligned model, indicating that the proposed Align method can yield better performance than semi-supervised method when using the same amount of unlabeled and labeled data. Besides, when combined with semi-supervised learning, the performance of both Aligned and DnA could further improve by 1.1% and 1.3%, respectively. Further, we compare with the state-of-the-art semi-supervised method FixMatch [35] on the official CIFAR100 4-shot benchmark (corresponding to 400 labels setting in the original paper). As shown in Table 2b, while FixMatch achieves a promising performance of 48.9% when training from scratch, initializing FixMatch with self-supervised pre-training can hardly improve its performance even though we have tried smaller learning rate and warm-up. This may be because the forgetting problem is serious given it requires a long training schedule. In contrast, the proposed method can yield a significantly higher performance of 52.6 and 56.4 for w/o and w/ semi-supervised methods, respectively.

**Weight decomposition better prevents forgetting.** As the effectiveness of the proposed weight decomposition method works by preventing forgetting source information, we also include a baseline for comparing with the previous learning without forgetting methods. We choose L2-SP [44], Delta [29] and BSS [9] as our baselines. The results are shown in Table 3: for 5-shot accuracy in iNaturalist-1k, while applying [L2-SP, Delta, Delta+BSS] for Align yield an improvement over the naive Aligned by [0.3%, 0.7%, 1.0%], respectively. While the performance Align+Delta+BSS is on par with DnA, the proposed DnA can also be combined with Delta+BSS and further yield accuracy of 48.5%.

Table 3: Compare with learning without forgetting methods in terms of the 5-shot performance on iNaturalist-1k.

| Method | Accuracy |
|---|---|
| Align+L2-SP | 47.7 |
| Align+Delta | 48.1 |
| Align+Delta+BSS | 48.3 |
| DnA (Ours) | 48.3 |
| DnA+Delta+BSS(Ours) | 48.5 |

**DnA surpass the State-Of-The-Art (SOTA).** To further study the effectiveness of the proposed method, we compare it with the SOTA supervised method BiT and SOTA self-supervised method TUP [28] on three different datasets. As illustrated in the Table 4, while TUP [28] can surpass the state-of-the-art supervised method BiT in terms of the Mean accuracy by [1.0%, 1.9%, 3.5%, 4.2%] for [2-shot, 4-shot, 6-shot, 10-shot] performance, respectively. The proposed DnA can further yield an improvement with a significant margin of [8.3%, 8.7%, 5.0%, 3.5%] compared to TUP. For a fair comparison with TUP,

Table 4: The few-shot fine-tuning performance comparison on different datasets for different pre-trained models. SimCLR here represents the SimCLR pre-trained on ImageNet-1k. The average accuracy and standard deviation (%) on five different labeled subsets are reported. #Few-shot means the number of few-shot samples for each class. The performance of BiT [26] and TUP [28] are from [28]. We consider few-shot performance on the random sampled few-shot subsets, which corresponds to the oracle few-label transfer setting of [28]. LP denotes linear probing. DnA-MoCo is the combination of ours and MoCo v2.

| #Few-shot | Method | CIFAR100 | EuroSAT | Food101 | Mean |
|---|---|---|---|---|---|
| 2 | BiT [26] | 37.4 | 68.3 | 24.5 | 43.4 |
| | TUP [28] | 30.2 | 68.8 | 34.3 | 44.4 |
| | SimCLR | 15.2±1.4 | 57.9±1.3 | 13.6±0.4 | 28.9 |
| | SimCLR - LP | 25.0±0.9 | 69.5±3.1 | 20.0±0.8 | 38.2 |
| | DnA (Ours) | 40.4±1.6 | 77.5±1.6 | 40.2±1.4 | 52.7 |
| | DnA-MoCo (Ours) | **42.4±1.3** | **80.7±2.2** | **42.7±1.5** | **55.3** |
| 4 | BiT [26] | 47.8 | 79.1 | 35.3 | 54.1 |
| | TUP [28] | 43.9 | 76.2 | 48.0 | 56.0 |
| | SimCLR | 26.6±1.5 | 65.7±2.8 | 22.7±0.2 | 38.3 |
| | SimCLR - LP | 34.1±0.4 | 77.7±2.4 | 27.7±0.6 | 46.5 |
| | DnA (Ours) | 53.4±0.8 | **86.5±0.7** | **54.2±0.4** | **64.7** |
| | DnA-MoCo (Ours) | **53.7±0.6** | 85.1±2.1 | 53.9±0.9 | 64.2 |
| 6 | BiT [26] | 54.2 | 82.6 | 41.3 | 59.4 |
| | TUP [28] | 52.7 | 80.6 | 55.4 | 62.9 |
| | SimCLR | 34.7±0.9 | 72.0±1.7 | 28.3±0.7 | 45.0 |
| | SimCLR - LP | 38.2±0.8 | 81.2±1.2 | 32.6±0.3 | 50.7 |
| | DnA (Ours) | 57.0±0.3 | **87.8±1.0** | **59.0±0.7** | **67.9** |
| | DnA-MoCo (Ours) | **57.5±0.4** | 87.2±0.9 | 58.9±0.6 | **67.9** |
| 10 | BiT [26] | 59.9 | 86.3 | 48.8 | 65.0 |
| | TUP [28] | 60.9 | 84.1 | 62.6 | 69.2 |
| | SimCLR | 45.5±1.1 | 76.6±1.7 | 37.3±0.6 | 53.1 |
| | SimCLR - LP | 43.2±0.5 | 83.4±1.3 | 38.9±0.5 | 55.2 |
| | DnA (Ours) | 61.6±0.4 | **89.4±1.2** | 63.3±0.5 | 71.4 |
| | DnA-MoCo (Ours) | **62.3±0.3** | 89.3±0.3 | **65.0±0.4** | **72.2** |

we also employ DnA with pre-training of MoCo v2, termed as DnA-Moco. DnA-Moco also yields consistent improvement. It's worth noting that the proposed method is also more efficient than TUP as DnA i) has no sampling step ii) conducts the training only on the target dataset, which is 5 times smaller than the dataset employed in TUP.

Besides, applying DnA for the SS pre-trained model can make a large difference in these datasets as DnA can surpass its start point, the SimCLR model, by a large margin of [23.8%, 26.4%, 22.9%, 18.3%] and [14.5%, 18.2%, 17.2%, 16.2%] for fine-tuning and linear probing setting, respectively. In comparison,

the improvement on iNaturalist-1k is milder, the intuition behind this is that the domain difference between these datasets and ImageNet is much larger.

### 4.3 Ablation study

**Does SS ImageNet pre-training help?** As the proposed DnA improves a lot based on the pre-trained model, a natural question arises: would the in-domain data be enough for few-shot learning? How many benefits can ImageNet pre-training bring? This motivates us to compare the proposed DnA method with SS pre-training on the target dataset. As illustrated in Table 5, when training without ImageNet pre-trained model, in CIFAR100, the performance would degrade from [53.4%, 61.6%] to [32.8%, 49.2%] for [4-shot, 10-shot] performance, respectively. The observation on Food-101 is also consistent, demonstrating the importance of the SS ImageNet pre-trained model. Also, this observation further motivates us to employ the learning with forgetting method.

Table 5: Comparing the few-shot performance on CIFAR100 and Food101 between DnA (with SS ImageNet pre-trained model) and in-domain SS pre-training (without SS ImageNet pre-trained model).

| Dataset | Pretrain | 4-shot | 10-shot |
|---|---|---|---|
| CIFAR100 | | 32.8 | 49.2 |
| | ✓ | 53.4 | 61.6 |
| Food101 | | 29.7 | 44.9 |
| | ✓ | 54.2 | 63.3 |

**The fix components choosing for DnA.** In this part, we ablation study the effect of fixing different components of the weight decomposition. As illustrated in Table 6a, when we free the fixing weight and mask for every component of the three terms, the performance would degrade to 47.4%, which is equal to the performance of Align. This is because the tunable parameters are even more than the original pre-trained model, which means this architecture can not prevent forgetting. By adding the fixing mask on $S$, the performance could improve to 47.9%, showing that only applying the mask can prevent forgetting. By fixing both sparse mask and low-rank sub-spaces via freezing V, the performance could further improve to 48.3% with a small variance of 0.1%, showing that fixing low-rank subspace could further prevent information loss. However, when switching the fixing low-rank component from $U$ to $V$, the performance could decrease by 0.5%. The intuition behind this is that, for resnet-50, the size of $U \in \mathbf{R}^{C_{in}H \times r}$ is usually smaller or equal to $V \in \mathbf{R}^{r \times WC_{out}}$, indicating choosing $V$ as fixed bases could preserve more information. When fixing $S$ and tuning $U$, the performance would decrease by 0.3% compared to tuning $U$ and $S$, showing the $S$ can capture the "out-of-the-domain" knowledge when subspace is fixed. When fixing every term, it would fail back to the origin SimCLR model. Last but not least, as shown in Table 6b, when employing the decomposition strategy with only the low-rank component as $W = UV$, the performance would be weaker to 'tuning $U$,$S$' of $W = UV + S$ for both 'tuning $U$, $V$' and 'tuning $U$'. Because $W = UV$ needs a higher rank to minimize the decomposition loss and thus fail to prevent forgetting.

Table 6: Ablation study for different decomposition and fixing strategies of weight on iNaturalist-1k in terms of 5-shot performance. The ✓ under $U$, $V$ denotes the corresponding terms is adjustable. The ✓ under $S$ denotes the value of $S$ is adjustable while sparse mask is fixed. In contrast, ✓✓ for $S$ means $S$ is adjustable for both value and mask. (a) and (b) employ different decomposition strategies. (a) The employed $W = UV + S$ decomposition, (b) $W = UV$, the decomposition strategy without low rank term.

| $U$ | $V$ | $S$ | 5-shot |
|---|---|---|---|
| ✓ | ✓ | ✓✓ | 47.4 |
| ✓ | ✓ | ✓ | 47.9±0.2 |
| ✓ | | ✓ | **48.3±0.1** |
| | ✓ | ✓ | 47.7 |
| ✓ | ✓ | | 47.8±0.3 |
| ✓ | | | 48.0±0.3 |
| | | | 45.0 |

(a)

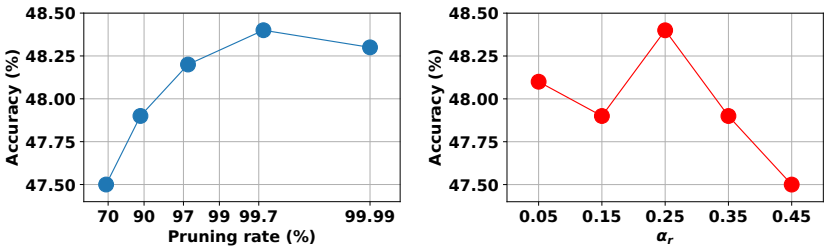
| $U$ | $V$ | 5-shot |
|---|---|---|
| ✓ | ✓ | 47.6 |
| ✓ | | 47.9 |

(b)

**Hyper-parameters for weight decomposition.** We further report the ablation study on the selection of the sparsity threshold $s$ and the value $r$ (We remove the sparsity fixing step here). The performance with different sparsity levels is shown in Figure 3a. We can see that either a too large sparsity or a too small sparsity would lead to inferior performance. Especially, when the sparsity decrease from 99.7% to 70%, the performance could decrease very fast from 48.3% to 47.5%, which is very close to the Align performance of 47.4%. This is because too large flexibility of the $S$ would override the low-rank component and make DnA degrade to Align method. The sweet point is sparsity of 99.7%, showing the "out-of-domain" knowledge can be efficiently encapsulated with only 0.3% free parameters.

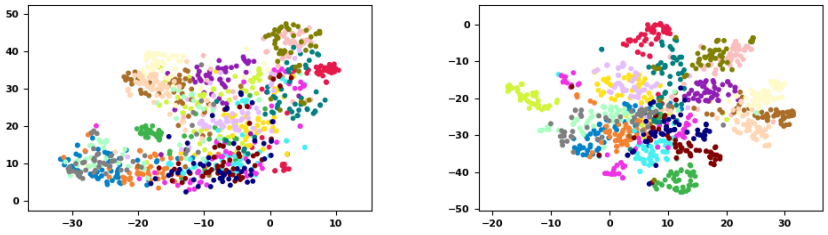
The study of the different number of ranks is shown in Figure 3b. When choosing rank ratio $\alpha_r$ ranging from 0.05 to 0.35, the DnA could yield an improvement of at least 0.5%, demonstrating the choice of rank is not sensitive. For a too large rank of 0.45, the performance would decrease to the level of Align because of too much flexibility. It is also worth noting the proposed DnA can even achieve a performance of 48.1% with a very small $\alpha_r$ of 0.05. Combining a very sparse mask of 99.7%, the proposed DnA can adapt the SS pre-trained model with very less parameters compared to the original network, showing its parameter efficiency.

**Representation visualization** In this section, we utilize t-SNE [30] to analyze the feature distribution before and after applying the proposed DnA. As illustrated in Figure 4, for SimCLR pre-trained on the ImageNet, the features of samples from different classes would overlap with each other. Only a small portion of samples from the same class form a cluster. In contrast, after applying DnA, the overlap can be significantly alleviated. Many samples in the same class




(a) (b)

Fig. 3: Ablation study for hyper-parameter selection on iNaturalist-1k in terms of the 5-shot accuracy. (a) studies the influence different pruning rates, the sparsity of [68.10%, 88.83%, 97.39%, 99.74%, 99.99%] shown in the figure correspond to $s$ of [0.02, 0.05, 0.1, 0.2, 0.5], respectively. (b) studies the different selection for rank ratio $\alpha_r$.



(a) (b)

Fig. 4: t-SNE visualization for SimCLR representations before and after applying DnA on the test dataset of CIFAR100 (We random sample 20 classes for visualization). (a) t-SNE before applying DnA (SimCLR) (b) t-SNE after applying DnA. Different color stands for different classes.

are clustered together. Some clusters even have a gap with a large margin to other clusters. We believe the strong few-shot performance of DnA is closely related to the good learned representations.

## 5 Conclusion

When transferring SS pre-trained model to the downstream task with a domain discrepancy, the few-shot performance of the SS model could degraded more compared to its supervised counterpart. In this paper, we tackle this problem with DnA framework, which is designed from a domain adaptation perspective. The proposed DnA achieved new SOTA performance on three different datasets (CIFAR100, NeurSAT and Food101), demonstrating its effectiveness. We believe our technique is beneficial for realistic few-shot classification.

## References


1. Aghajanyan, A., Zettlemoyer, L., Gupta, S.: Intrinsic dimensionality explains the effectiveness of language model fine-tuning. arXiv preprint arXiv:2012.13255 (2020)
2. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101–mining discriminative components with random forests. In: European conference on computer vision. pp. 446–461. Springer (2014)
3. Candès, E.J., Li, X., Ma, Y., Wright, J.: Robust principal component analysis? Journal of the ACM (JACM) **58**(3), 1–37 (2011)
4. Chen, T., Cheng, Y., Gan, Z., Liu, J., Wang, Z.: Data-efficient gan training beyond (just) augmentations: A lottery ticket perspective. arXiv preprint arXiv:2103.00397 (2021)
5. Chen, T., Liu, S., Chang, S., Cheng, Y., Amini, L., Wang, Z.: Adversarial robustness: From self-supervised pre-training to fine-tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 699–708 (2020)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
7. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. arXiv preprint arXiv:2006.10029 (2020)
8. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
9. Chen, X., Wang, S., Fu, B., Long, M., Wang, J.: Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. Advances in Neural Information Processing Systems **32** (2019)
10. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Ericsson, L., Gouk, H., Hospedales, T.M.: How well do self-supervised models transfer? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5414–5423 (2021)
12. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2019)
13. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728 (2018)
14. Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Pires, B.A., Guo, Z.D., Azar, M.G., et al.: Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733 (2020)
15. Guo, D., Rush, A.M., Kim, Y.: Parameter-efficient transfer learning with diff pruning. arXiv preprint arXiv:2012.07463 (2020)
16. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. arXiv preprint arXiv:1506.02626 (2015)
17. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9729–9738 (2020)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)

19. Helber, P., Bischke, B., Dengel, A., Borth, D.: Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **12**(7), 2217–2226 (2019)
20. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
21. Huang, W., Ke, Z., Cui, Z.X., Cheng, J., Qiu, Z., Jia, S., Ying, L., Zhu, Y., Liang, D.: Deep low-rank plus sparse network for dynamic mr imaging (2021)
22. Islam, A., Chen, C.F., Panda, R., Karlinsky, L., Radke, R., Feris, R.: A broad study on the transferability of visual representations with contrastive learning. arXiv preprint arXiv:2103.13517 (2021)
23. Jaderberg, M., Vedaldi, A., Zisserman, A.: Speeding up convolutional neural networks with low rank expansions. arXiv preprint arXiv:1405.3866 (2014)
24. Jiang, Z., Chen, T., Chen, T., Wang, Z.: Robust pre-training by adversarial contrastive learning. In: NeurIPS (2020)
25. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the national academy of sciences **114**(13), 3521–3526 (2017)
26. Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., Houlsby, N.: Big transfer (bit): General visual representation learning. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16. pp. 491–507. Springer (2020)
27. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
28. Li, S., Chen, D., Chen, Y., Yuan, L., Zhang, L., Chu, Q., Liu, B., Yu, N.: Improve unsupervised pretraining for few-label transfer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10201–10210 (2021)
29. Li, X., Xiong, H., Wang, H., Rao, Y., Liu, L., Chen, Z., Huan, J.: Delta: Deep learning transfer using feature map with attention for convolutional networks. arXiv preprint arXiv:1901.09229 (2019)
30. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
31. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. pp. 69–84. Springer (2016)
32. Phoo, C.P., Hariharan, B.: Self-training for few-shot transfer across extreme task differences. arXiv preprint arXiv:2010.07734 (2020)
33. Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., Khudanpur, S.: Semi-orthogonal low-rank matrix factorization for deep neural networks. In: Interspeech. pp. 3743–3747 (2018)
34. Sainath, T.N., Kingsbury, B., Sindhwani, V., Arisoy, E., Ramabhadran, B.: Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6655–6659. IEEE (2013)
35. Sohn, K., Berthelot, D., Li, C.L., Zhang, Z., Carlini, N., Cubuk, E.D., Kurakin, A., Zhang, H., Raffel, C.: Fixmatch: Simplifying semi-supervised learning with consistency and confidence. arXiv preprint arXiv:2001.07685 (2020)

36. Su, J.C., Maji, S., Hariharan, B.: When does self-supervision improve few-shot learning? In: European Conference on Computer Vision. pp. 645–666. Springer (2020)
37. Sun, M., Baytas, I.M., Zhan, L., Wang, Z., Zhou, J.: Subspace network: deep multi-task censored regression for modeling neurodegenerative diseases. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2259–2268 (2018)
38. Tai, C., Xiao, T., Zhang, Y., Wang, X., et al.: Convolutional neural networks with low-rank regularization. arXiv preprint arXiv:1511.06067 (2015)
39. Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P.: Rethinking few-shot image classification: a good embedding is all you need? In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16. pp. 266–282. Springer (2020)
40. Trinh, T.H., Luong, M.T., Le, Q.V.: Selfie: Self-supervised pretraining for image embedding. arXiv preprint arXiv:1906.02940 (2019)
41. Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., Belongie, S.: The inaturalist species classification and detection dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8769–8778 (2018)
42. Xie, E., Ding, J., Wang, W., Zhan, X., Xu, H., Sun, P., Li, Z., Luo, P.: Detco: Unsupervised contrastive learning for object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8392–8401 (2021)
43. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10687–10698 (2020)
44. Xuhong, L., Grandvalet, Y., Davoine, F.: Explicit inductive bias for transfer learning with convolutional networks. In: International Conference on Machine Learning. pp. 2825–2834. PMLR (2018)
45. Yang, Y., Xu, Z.: Rethinking the value of labels for improving class-imbalanced learning. arXiv preprint arXiv:2006.07529 (2020)
46. You, Y., Gitman, I., Ginsburg, B.: Large batch training of convolutional networks. arXiv preprint arXiv:1708.03888 (2017)
47. Yu, X., Liu, T., Wang, X., Tao, D.: On compressing deep models by low rank and sparse decomposition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7370–7379 (2017)
48. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: Self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1476–1485 (2019)
49. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: European conference on computer vision. pp. 649–666. Springer (2016)
50. Zhang, Y., Chuangsuwanich, E., Glass, J.: Extracting deep neural network bottleneck features using low-rank matrix factorization. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp. 185–189. IEEE (2014)
51. Zhang, Z., Chen, X., Chen, T., Wang, Z.: Efficient lottery ticket finding: Less data is more. In: International Conference on Machine Learning. pp. 12380–12390. PMLR (2021)
52. Zhao, M., Lin, T., Mi, F., Jaggi, M., Schütze, H.: Masking as an efficient alternative to finetuning for pretrained language models. arXiv preprint arXiv:2004.12406 (2020)

53. Zhao, Y., Li, J., Gong, Y.: Low-rank plus diagonal adaptation for deep neural networks. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5005–5009. IEEE (2016)
54. Zhou, T., Tao, D.: Greedy bilateral sketch, completion & smoothing. In: Artificial Intelligence and Statistics. pp. 650–658. PMLR (2013)
55. Zhou, Z., Li, X., Wright, J., Candes, E., Ma, Y.: Stable principal component pursuit. In: 2010 IEEE international symposium on information theory. pp. 1518–1522. IEEE (2010)
56. Zhu, H., Wang, Z., Zhang, H., Liu, M., Zhao, S., Qin, B.: Less is more: Domain adaptation with lottery ticket for reading comprehension. In: Findings of the Association for Computational Linguistics: EMNLP 2021. pp. 1102–1113 (2021)