

Lark: Low-Rank updates after knowledge localization for Few-shot Class-Incremental Learning

Jinxin Shi¹, Jiabao Zhao^{2*}, Yifan Yang³, Xingjiao Wu¹, Jiawen Li¹, Liang He^{1*}

¹East China Normal University ²Donghua University ³Transwarp Technology (Shanghai) Co., Ltd

jinxinshi@stu.ecnu.edu.cn, jbzhaob@dhu.edu.cn, yifan.yang@transwarp.io

xjwu@pharm.ecnu.edu.cn, jwli@stu.ecnu.edu.cn, lhe@cs.ecnu.edu.cn

Abstract

For Few-Shot Class-Incremental Learning (FSCIL), direct fine-tuning causes significant parameter shifts, resulting in catastrophic forgetting and increased resource consumption. While, freezing the pre-trained backbone exacerbates the inconsistency between the backbone and the evolving classifier. To overcome these challenges, we introduce a method called Low-Rank updates after knowledge localization (Lark). In the knowledge localization phase, the Fisher Information Matrix is calculated to measure the sensitivity of parameters in different layers to previously acquired knowledge. This phase ultimately identifies the parameters within the model that are most suitable for learning new knowledge. In the subsequent incremental editing phase, a low-rank incremental update strategy is applied. This strategy ensures that the model parameter updates adhere to a Rank-One matrix structure. By doing so, it minimizes alterations to the original parameters, thereby enabling the model to integrate new knowledge while retaining as much of the previous knowledge as possible. Extensive experimental results demonstrate that the Lark method achieves significant performance improvements on the CIFAR100, mini-ImageNet, and CUB200 datasets, surpassing current state-of-the-art methods.

1. Introduction

Few-Shot Class-Incremental Learning (FSCIL) aims to enable models to continuously acquire new knowledge in dynamic environments. In this scenario, the model faces challenges related to empirical risk minimization [51], where insufficient data hinder the model’s ability to learn compre-

*Corresponding author.

This work is supported by the National Natural Science Foundation of China (Grant No.62207013, 62477011), the Shanghai Municipal Science and Technology Commission with Grant (No.24511106802) supported by the Fundamental Research Funds for the Central Universities (No.2232025D-35), and Suzhou Key Laboratory of Multimodal Data Fusion and Smart Healthcare (No.25SZZD14).

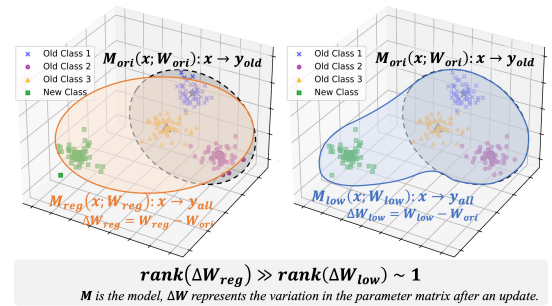


Figure 1. Conceptual illustration. The left panel depicts the space under regular training, whereas the right panel displays the space following Rank-One matrix updates. Notably, the Rank-One matrix updates (ΔW_{low}) exhibit higher precision compared to those produced by regular training (ΔW_{reg}).

hensive knowledge representations. Moreover, there exists the stability-plasticity dilemma [16, 36]. Specifically, when the model learns new knowledge (plasticity), it may result in forgetting previously acquired knowledge. Conversely, overemphasizing the retention of old knowledge (stability) can limit the model’s ability to learn new knowledge.

To address these challenges, existing FSCIL methods have explored various directions, such as meta-learning [7, 38], replaying old samples [22, 29], and enhancing feature representations [1, 37, 56]. However, the vast majority of these works are based on small-scale models such as convolutional neural networks. When Transformer models (e.g., ViTs) are introduced into FSCIL scenarios, the aforementioned challenges become even more severe.

On one hand, in ViTs, classification predictions rely on global attention interaction between the CLS token and all patch tokens, which results in classifier-only optimization methods disrupting the cognitive consistency between the backbone and the classifier [53, 61]. On the other hand, due to the massive number of parameters in ViT models and the lack of the inherent inductive bias of convolutional networks, directly fine-tuning with only a few incremental samples is more prone to overfitting and representation drift [5]. These suggest that effective adaptation in few-shot

scenarios demands not only accurately identifying the subset of parameters best suited for acquiring new knowledge, but also making minimal adjustments to the parameter distribution to preserve the essential pre-trained knowledge. Therefore, we propose Lark, a method that first identifies parameters suitable for fine-tuning and then makes subtle adjustments to those selected parameters.

Our approach is divided into two main stages: Knowledge Localization and Incremental Editing. During the Knowledge Localization stage, our goal is to identify the model components most suitable for incremental learning. We achieve this by leveraging the Fisher Information Matrix [17, 18] to measure the impact of each component on the model’s output. Analyzing the Fisher Matrix enables us to pinpoint parameters that have a minimal contribution to the retention of learned knowledge. However, given the vast number of parameters in large-scale vision models, directly analyzing all parameters would incur prohibitive computational and memory costs [30, 44]. To address this issue, we introduce an equivalent substitution of hidden states as an indirect metric, reducing computational complexity and improving the efficiency of the localization process.

After identifying the parameters suitable for editing, we employ an incremental editing method based on Rank-One [4, 21] matrix to optimize these parameters. Specifically, given a parameter matrix W , we update it by adding an outer product matrix of rank one. This low-rank update introduces new knowledge while minimally perturbing the parameter space (as shown in Figure 1.), thereby preserving the model’s overall structure [11, 15]. We summarize the contributions of this work as follows:

- We propose Lark, a method that locates the model parameters most suitable for learning new knowledge and updates them using a low-rank matrix. It can be used for FSCIL in large visual models.
- To mitigate the high computational cost of gradient calculations over numerous parameters, we use the information contained in hidden states to reflect and assess the importance of parameters across different layers.
- We analyze the information encoding process within the MLP and MHSA modules, clarifying the significance of updating different weight matrices in these modules.
- Our method demonstrates superior performance compared to previous state-of-the-art methods across three datasets. Additionally, we analyze and validate the proposed method’s effectiveness on other visual tasks.

2. Related Work

Few-Shot Class-Incremental Learning. FSCIL aims to enable models to continually learn in few-shot scenarios, and numerous studies have explored this from various perspectives [40, 46, 55]. MetaFSCIL [7] addresses data imbalance between base and incremental sessions by con-

structing multi-stage pseudo-incremental tasks during the base session. Meta-learning methods like FSLL [33] resist overfitting by selectively updating parameters, while Pseudo-Frequency Refinement (PFR) [38] introduces an architecture enhancing ability for fine-grained object recognition. However, these methods primarily focus on adaptability and often neglect improving representation capacity.

To address this limitation, some works train robust pre-trained models [52] during the base session and integrate them with prototype classifiers for incremental learning. CLOSER [37] ensures good inter-class separability through extensive contrastive learning on base class data, using mean vectors as classification prototypes. OrCo [1] adjusts class prototype distributions via orthogonal constraints, enabling incremental classes to align with the original feature space. However, a static pre-trained model cannot consistently align with a classifier that continuously evolves during incremental sessions [27, 53]. Thus, we argue that further optimization of the pre-trained model is necessary to maintain alignment with the evolving classifier.

Global optimization of pre-trained models is undesirable, as it cause forgetting of prior knowledge and overfitting to new classes [39]. EWC [18] proposes evaluating parameter importance through gradient, noting that different parameters vary in importance for task cognition. Building on this, WaRP [17] dynamically adjusts parameter weights using rotation in weight space, effectively adapting to new knowledge while preserving old knowledge. However, computing gradients for all parameters in large models is inefficient [2, 12]. Therefore, we assess parameter importance using the hidden states of each layer.

Model Editing. ME aims to modify pre-trained models to retain existing knowledge while integrating new information. Mainstream approaches typically employ low-rank matrix update strategies. For instance, ROME [35] identifies parameter regions associated with specific knowledge via causal mediation analysis and applies precise low-rank updates. Similarly, PEMT [25] highlights the importance of Multi-Head Self-Attention (MHSA), introducing low-rank updates into MHSA modules to improve the integration of new knowledge. We further analyzed the roles of Multi-Layer Perceptrons (MLP) and MHSA in visual tasks and refined model editing methods accordingly. Other studies [43, 49] utilize low-rank properties for continual learning by performing gradient updates orthogonal to the subspace of previous tasks, thereby mitigating catastrophic forgetting. WaRP [17] preserves existing knowledge within a smaller low-rank matrix, allowing the model to primarily accommodate new information. However, large-scale pre-trained models generally store substantial old knowledge. Thus, we argue for preserving existing knowledge to the greatest extent possible and representing new knowledge through a low-rank matrix.

3. Method

3.1. Preliminaries

FSCIL Setting. Following the mainstream FSCIL setting [46], in which the training process is divided into $T + 1$ sessions $\{D_{base}, D_1, \dots, D_T\}$ with corresponding disjoint label sets $\{C_{base}, C_1, \dots, C_T\}$. The first session D_{base} contains a large training dataset. The goal of FSCIL is to recognize the incremental dataset D_t during the incremental step t while maintaining a good discriminative ability for the already learned classes, which can be represented as follows:

$$\mathcal{F}_t(x) = \begin{cases} y & \text{if } x \in D_t \\ \mathcal{F}_{t-1}(x) & \text{if } x \notin D_t \end{cases}, \quad (1)$$

where \mathcal{F}_t is the model after incremental editing at stage t .

Specifically, the model \mathcal{F} is composed of a feature extractor $f_\theta(\cdot)$, and a classifier $g_\phi(\cdot)$. The feature extractor $f_\theta(\cdot) : X \rightarrow Z$ maps an input x to a feature vector $z = f_\theta(x)$ in the feature space Z . The classifier $g_\phi(\cdot) : Z \rightarrow Y$ then produces the class prediction. Notably, in this work, $g_\phi(\cdot)$ is a Nearest Neighbor Classifier (See Section 6.1 of the supplement) that remains unaffected by updates to $f_\theta(\cdot)$.

Hidden State. How can we identify the differences in the perception of learned knowledge by different parameters? A straightforward and effective method is to compute the gradients of the objective function with respect to different parameters [17, 18]. However, for models with numerous parameters, calculating gradients directly is computationally inefficient [12, 23]. Instead, the hidden states generated dynamically during inference reflect each layer’s contribution to the current task more effectively [34, 45].

The hidden states of the sample in the l -th layer of the encoder include the output of the Multi-Head Self-Attention mechanism (MHSA), denoted as $h_a^{(l)}$, and the output of the Multi-Layer Perceptron (MLP), denoted as $h_m^{(l)}$:

$$\begin{aligned} h_a^{(l)} &= [z_a^{(l)}; p_{a,1}^{(l)}; p_{a,2}^{(l)}; \dots; p_{a,N}^{(l)}] = \text{MHSA}(h_m^{(l-1)}) + h_m^{(l-1)}, \\ h_m^{(l)} &= [z_m^{(l)}; p_{m,1}^{(l)}; p_{m,2}^{(l)}; \dots; p_{m,N}^{(l)}] = \text{MLP}(h_a^{(l)}) + h_a^{(l)}, \end{aligned} \quad (2)$$

where $l = 1, \dots, L$, with L denoting the total number of encoder blocks in $f_\theta(\cdot)$. And p is the different patches, N represents the number of patches for the image.

3.2. Knowledge Localization

Localization aims to identify the model components most effective at assimilating new knowledge. Although some approaches compute the Fisher information matrix for each parameter [18, 31], this process becomes computationally prohibitive for large-scale models because of the extensive gradient calculations required. To address this limitation, we instead use hidden states. By applying our method, we can determine how various layers respond to previously

learned information. As shown in Figure 2, layers 8, 9, and 10 exhibit heightened sensitivity to category 1, whereas layers 5 and 6 minimally influence its recognition.

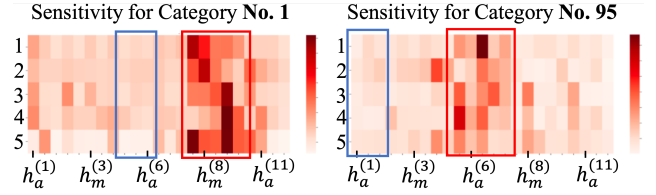


Figure 2. Sensitivity of categories 1 and 95 in the CUB200 dataset across different hidden layers. The blue boxes denote non-sensitive regions, whereas the red boxes signify sensitive regions.

Calculation of the Fisher Information Matrix. To quantify the layer-wise influence on model decisions, we use a perturbation-driven analysis. Taking the MHSA module in layer l as an example, we introduce Gaussian perturbation $\epsilon \sim \mathcal{N}(0, \sigma^2)$ into the hidden state $h_a^{(l)}$ (Details in the supplement section 6.2). We then measure the resulting change in the final class token $z_m^{(L)}$ to assess the sensitivity of the output to the current layer. Formally, the contribution of token a at layer l is quantified by τ_a^l as defined in:

$$\tau_a^l = \frac{\left\| \frac{\partial \mathcal{L}(z_m^{(L)}, \tilde{z}_m^{(L)})}{\partial z_a^{(l)}} \right\|^2 + \sum_{n=1}^N \left\| \frac{\partial \mathcal{L}(z_m^{(L)}, \tilde{z}_m^{(L)})}{\partial p_{a,n}^{(l)}} \right\|^2}{N + 1}, \quad (3)$$

where the objective function $\mathcal{L}(z_m^{(L)}, \tilde{z}_m^{(L)}) = 1 - \frac{z_m^{(L)} \cdot \tilde{z}_m^{(L)}}{\|z_m^{(L)}\| \|\tilde{z}_m^{(L)}\|}$ captures the directional change of the class token before ($z_m^{(L)}$) and after ($\tilde{z}_m^{(L)}$) perturbation via cosine similarity. In particular, this objective aligns naturally with the nearest-neighbor classifier $g_\phi(\cdot)$, ensuring consistency between the backbone and the classifier.

Subsequently, to assess the parameter sensitivity for each category (for example, the k -th category), we randomly selected J samples from the k -th category to estimate the Fisher Information Matrix $\mathbb{F}_k \in \mathbb{R}^{J \times 2L}$.

$$\mathbb{F}_k \approx \begin{bmatrix} \tau_a^{(1,1)}, & \dots, & \tau_m^{(1,1)}, & \dots, & \tau_a^{(L,1)}, & \tau_m^{(L,1)} \\ \tau_a^{(1,2)}, & \dots, & \tau_m^{(1,2)}, & \dots, & \tau_a^{(L,2)}, & \tau_m^{(L,2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \tau_a^{(1,J)}, & \dots, & \tau_m^{(1,J)}, & \dots, & \tau_a^{(L,J)}, & \tau_m^{(L,J)} \end{bmatrix}. \quad (4)$$

The sum of each column in \mathbb{F}_k reflects the influence of a specific layer’s parameters on the designated category. Consequently, the sensitivity of parameters from different layers to the given category can be approximate as:

$$\left\{ \sum_{j=1}^J \tau_a^{(1,j)}, \dots, \sum_{j=1}^J \tau_m^{(l,j)}, \dots, \sum_{j=1}^J \tau_a^{(L,j)}, \sum_{j=1}^J \tau_m^{(L,j)} \right\}, \quad (5)$$

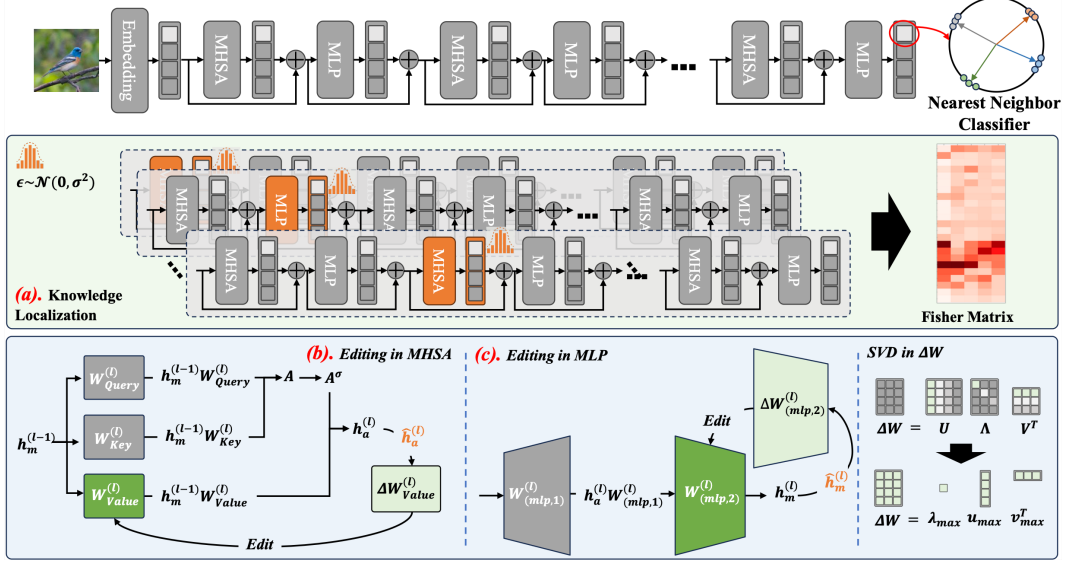


Figure 3. Overview of the Lark Method. (a) illustrates the Knowledge Localization process, in which gradients of the noise-augmented hidden state are sequentially computed to obtain the Fisher Information Matrix. (b) and (c) represent the incremental editing processes for MHA and MLP, respectively.

where $\sum_{j=1}^J \tau_a^{(1,j)}$ represents the impact of the first layer’s MHA on a specific category.

Ultimately, we aggregated the Fisher information matrices for each layer’s parameters across all categories and employed the Lowest-K criterion to identify parameters exhibiting minimal interference with old knowledge for incremental editing (refer to Supplementary Section 6.3).

3.3. Incremental Editing

After locating parameters suitable for editing, we edit them to incorporate new knowledge. To preserve previously learned knowledge, we aim to introduce minimal-norm perturbations to the original parameters. Therefore, the rank-one update method is adopted to minimize parameter adjustments during editing. In contrast to previous approaches that focus solely on MLP modules [25, 35], our work conducts a rigorous structural analysis of the multi-head self-attention (MHA) module. Our analysis reveals that modifications to the Query and Key matrices exert a limited influence, whereas changes to the Value matrix directly affect information weighting. Building on these insights, we propose a more detailed method for incremental editing.

Editing in MHA. MHA demonstrates a high capacity for information integration during the model’s knowledge extraction process [13] while encompassing general knowledge extraction patterns associated with specific facts [25]. In visual pre-trained models like ViT, the class token can only represent sample class information after continuously aggregating the patch tokens representing the image itself. Consequently, it is considered necessary to modify MHA to maintain its information extraction capabilities in line

with updates in knowledge.

Specifically, W_{Query} and W_{Key} govern the relationships between tokens and the attention distribution, while W_{Value} predominantly determines the final output. As illustrated in Figure 3(b), the hidden state $h_m^{(l-1)}$ passes through MHA to produce the original output $h_a^{(l)}$:

$$\begin{aligned} h_a^{(l)} &= \text{softmax}\left(\frac{A}{\sqrt{d_k}}\right) h_m^{(l-1)} W_{Value}^{(l)} \\ &= A^\sigma h_m^{(l-1)} W_{Value}^{(l)}, \end{aligned} \quad (6)$$

where $A = h_m^{(l-1)} W_{Query}^{(l)} \cdot (h_m^{(l-1)} W_{Key}^{(l)})^T$ represents the attention matrix, which is subsequently transformed non-linearly and denoted as A^σ . From the perspective of the graph, A^σ can be viewed as an adjacency matrix [54]. And, $A^\sigma h_m^{(l-1)}$ can be considered as input, relative to $W_{Value}^{(l)}$.

Therefore, we argue that directly modifying W_{Value} to adjust the output of MHA does not impact the attention distribution among tokens. To ensure that the edited parameters effectively capture new categorical information while preserving the model’s existing knowledge. The key issue is how to construct a matrix $\Delta W_{Value}^{(l)}$ with the NN-matrix norm that is applicable to all token vectors:

$$\min_{\Delta W_{Value}^{(l)}} \left\| \Delta W_{Value}^{(l)} \right\|_* \quad \text{s.t.} \quad \nu = \left(W_{Value}^{(l)} + \Delta W_{Value}^{(l)} \right) \mu, \quad (7)$$

where μ is the input vector of one token from $A^\sigma h_m^{(l-1)}$. ν is the target output of this token, which is contained in the hidden state $\hat{h}_a^{(l)}$ of the new knowledge and can be obtained through a regular training process (See Section 7.1 in sup-

plementary). Based on the Rank-One principle, $\Delta W_{Value}^{(l)}$ can be expressed as:

$$\Delta W_{Value}^{(l)} = \frac{\nu_z \otimes \mu_z^T}{\|\mu_z\|^2} \cup \left\{ \frac{\nu_i \otimes \mu_i^T}{\|\mu_i\|^2} \mid i = 1, 2, \dots, N \right\}, \quad (8)$$

where, $\frac{\nu_z \otimes \mu_z^T}{\|\mu_z\|^2}$ and $\frac{\nu_i \otimes \mu_i^T}{\|\mu_i\|^2}$ denote the outer product matrices derived from the class token and patch tokens, respectively.

Additionally, to investigate the necessity of modifying W_{Query} and W_{Key} , we conducted two experiments. As shown in Figure 4, the attention distribution of the class token over various image patches shows only minor differences between the trained model and the original model.

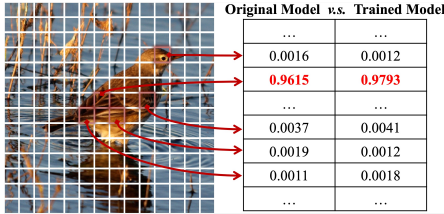


Figure 4. Comparison of the attention distribution of the class token when aggregating patch information before and after training.

And, in Table 1, we calculated the similarity of the attention matrices for the second layer of the model before and after training. As shown in the table, the similarity values are relatively high, with the lowest value being 0.7617 (Details in Section 7.2 in supplementary). Based on these findings, we consider that editing W_{Value} is sufficient.

	No.1	No.2	No.3	No.4	No.5
Sample 1	0.8321	0.8136	0.7884	0.7741	0.8072
Sample 2	0.8217	0.8429	0.8720	0.8817	0.9093
Sample 3	0.7958	0.8173	0.8325	0.8575	0.8627
Sample 4	0.8305	0.8581	0.8622	0.8823	0.8829
Sample 5	0.8052	0.7617	0.8438	0.8612	0.8087

Table 1. Similarity of attention matrices between the models before and after training on CIFAR100 Session 1. We consider a similarity greater than 0.75 to be similar.

Editing in MLP. In Transformer models, the *MLP* module plays a crucial role in knowledge storage [14]. The first layer of the MLP module generates a key that encodes the semantic properties of the input, While the second layer serves as an associative memory, retrieving the corresponding factual association [10]. In line with approaches like MEMIT [34] and ROME [35], we edit the second linear layer of the *MLP* module. As shown in Figure 3(c), we use the *MLP* module of the l -th layer as an example. Following Equation 7 and 8, we obtain the following inference:

$$\Delta W_{(mlp,2)}^{(l)} = \frac{\nu_z \otimes \mu_z^T}{\|\mu_z\|^2} \cup \left\{ \frac{\nu_i \otimes \mu_i^T}{\|\mu_i\|^2} \mid i = 1, 2, \dots, N \right\}, \quad (9)$$

here, μ_z and μ_i are the input vector of one token from $h_a^{(l)} W_{(mlp,1)}^{(l)}$, and ν_z and ν_i come from $\hat{h}m^{(l)}$.

It can be observed that, each hidden state is a sequence of tokens, and each token can be derived as a Rank-One matrix in the form of an outer product. The ΔW in Equation 8 and 9 is a sequence composed of $N + 1$ Rank-One matrices. Thus, a new problem arises: *how can we obtain a Rank-One matrix that can be added to the original parameter matrix?*

Considering that the model’s output aggregates other patch information, we believe that each token contributes to the final output. However, due to the subadditivity property of matrices, the rank of ΔW obtained in the form of summation or averaging cannot be guaranteed to be 1 (Details in Section 6.4 in supplementary). Hence, we further employ the Singular Value Decomposition (SVD) to decompose ΔW into orthogonal matrices U and V^T , and a diagonal matrix Λ . By retaining only the largest singular value, we preserve the most crucial information in ΔW and achieve an optimal low-rank approximation:

$$\Delta W \approx \lambda_{max} u_{max} v_{max}^T, \quad (10)$$

where λ_{max} is the largest singular value, and u_{max} and v_{max} are the corresponding left and right singular vectors.

4. Experiments

4.1. Datasets and Experimental Details

Datasets. We evaluate the performance on CIFAR100 [20], mini-ImageNet [42], and CUB200 [48]. Following [46], for CIFAR100 and mini-ImageNet, we use 60 classes as base classes and reserve the remaining 40 classes for new class introduction. These classes are divided into 8 incremental sessions, each consisting of a 5-way 5-shot incremental task. For CUB200, 100 classes are base classes, with the remaining 100 classes organized into 10 sessions, each comprising a 10-way 5-shot incremental task.

Evaluation Metrics. We primarily evaluate the performance using the accuracy of each session, the average accuracy across all sessions, and the performance drop rate (PD). PD measures the absolute drop in accuracy in the last session relative to the accuracy in the base session.

Baseline and Implementation Details. To comprehensively validate Lark’s versatility, we integrated it into two methods, OrCo [1] and CLOSER [37], which served as our baselines. Subsequently, we replaced their backbone with ViT-B/16 [9], referring to the revised implementations as OrCo-ViT and CLOSER-ViT. During knowledge localization, we select five samples per class to calculate perceptual differences and edit three parameter matrices to learn new knowledge. In the incremental editing stage, we use cosine scheduling with a maximum learning rate of 0.1 and set the number of epochs to 50. These configurations are consistently applied across all datasets. All experiments are

Dataset	Method	Acc. in each session (%) \uparrow										Avg \uparrow	PD \downarrow	
		Base	1	2	3	4	5	6	7	8	9			10
CIFAR100	[CVPR'21] CEC [55]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	/	/	59.53	23.93
	[TPAMI'22] LIMIT [60]	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	/	/	61.84	22.58
	[ICLR'23] WaRP [17]	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	/	/	65.82	25.57
	[ICCV'23] SV-T \ddagger [41]	86.77	82.82	80.36	77.20	76.06	74.00	72.92	71.68	69.75	/	/	76.84	17.02
	[ICME'23] CPE-CLIP \ddagger [8]	87.83	85.86	84.93	82.85	82.64	82.42	82.27	81.44	<u>80.52</u>	/	/	83.42	7.31
	[TIP'24] MTE-FSCIL [50]	80.71	76.17	73.11	69.28	65.57	62.65	60.44	58.06	57.62	/	/	67.07	23.09
	[ECCV'24] CLOSER-ViT \dagger [37]	<u>90.38</u>	87.31	85.94	84.31	83.55	82.24	80.19	78.54	76.28	/	/	83.19	14.10
	[CVPR'24] OrCo-ViT \dagger [1]	93.78	87.26	86.26	84.28	82.24	79.86	76.53	74.72	73.79	/	/	82.08	19.99
	[TPAMI'24] LRT \ddagger [57]	87.02	82.40	77.84	73.31	70.18	66.74	64.50	61.99	59.49	/	/	71.50	27.53
	[Ours]Lark in CLOSER-ViT \dagger	<u>90.38</u>	<u>88.79</u>	<u>87.29</u>	<u>86.84</u>	86.26	84.51	<u>84.13</u>	83.47	82.64	/	/	<u>86.03</u>	<u>7.74</u>
[Ours]Lark in OrCo-ViT \dagger	93.78	90.45	88.89	87.31	<u>85.55</u>	<u>84.24</u>	84.19	<u>82.54</u>	80.28	/	/	86.36	13.50	
mini-ImageNet	[CVPR'21] CEC [55]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	/	/	57.75	24.37
	[TPAMI'22] LIMIT [60]	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.14	/	/	59.05	23.18
	[ICLR'23] WaRP [17]	72.99	68.10	64.31	61.30	58.64	56.08	53.40	51.72	50.65	/	/	59.69	22.34
	[ICCV'23] SV-T \ddagger [41]	90.55	89.20	86.80	85.44	84.78	83.38	81.91	81.90	81.65	/	/	85.07	8.90
	[ICME'23] CPE-CLIP \ddagger [8]	90.23	89.56	87.42	86.80	86.51	85.08	83.43	83.38	82.77	/	/	86.13	7.46
	[TIP'24] MTE-FSCIL [50]	78.86	75.40	72.15	68.38	65.02	61.78	58.90	56.68	56.31	/	/	65.94	22.55
	[ECCV'24] CLOSER-ViT \dagger [37]	<u>93.12</u>	<u>91.18</u>	<u>88.54</u>	85.39	82.76	80.94	78.23	79.52	77.33	/	/	84.11	15.79
	[CVPR'24] OrCo-ViT \dagger [1]	93.58	86.26	83.17	81.51	78.53	77.59	76.11	75.56	75.03	/	/	80.82	18.55
	[TPAMI'24] LRT \ddagger [57]	90.17	85.82	81.70	78.12	75.04	71.71	68.88	66.74	65.34	/	/	75.94	24.83
	[Ours]Lark in CLOSER-ViT \dagger	<u>93.12</u>	92.14	90.56	89.73	87.77	86.47	85.30	84.61	83.69	/	/	88.15	9.43
[Ours]Lark in OrCo-ViT \dagger	93.58	90.12	88.46	<u>87.19</u>	<u>86.94</u>	<u>84.55</u>	<u>83.54</u>	<u>81.85</u>	<u>79.12</u>	/	/	<u>86.15</u>	14.46	
CUB200	[CVPR'21] CEC [55]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	23.57
	[TPAMI'22] LIMIT [60]	75.89	73.55	71.99	68.14	67.42	63.61	62.40	61.35	59.91	58.66	57.41	65.48	18.48
	[ICLR'23] WaRP [17]	77.74	74.15	70.82	66.90	65.01	62.64	61.40	59.86	57.95	57.77	57.01	64.66	20.73
	[ICCV'23] SV-T \ddagger [41]	84.19	82.63	81.21	78.97	79.38	77.64	77.55	75.71	75.91	75.77	76.17	78.65	8.02
	[ICME'23] CPE-CLIP \ddagger [8]	81.58	78.52	76.68	71.86	71.52	70.23	67.66	66.52	65.09	64.47	64.60	70.79	16.98
	[TIP'24] MTE-FSCIL [50]	78.94	75.72	72.46	68.25	67.86	64.82	63.70	62.89	61.25	60.47	60.36	66.97	18.58
	[ECCV'24] CLOSER-ViT \dagger [37]	<u>87.08</u>	<u>85.92</u>	83.50	81.47	79.24	78.34	77.14	74.68	73.10	73.22	73.03	78.79	14.05
	[CVPR'24] OrCo-ViT \dagger [1]	87.22	83.49	82.90	81.93	80.16	78.34	76.41	76.25	73.03	72.17	72.43	78.58	14.79
	[Ours]Lark in CLOSER-ViT \dagger	<u>87.08</u>	86.21	85.08	84.21	83.69	82.21	81.46	78.58	78.04	78.16	77.31	82.00	<u>9.77</u>
	[Ours]Lark in OrCo-ViT \dagger	87.22	84.48	83.46	<u>82.77</u>	81.54	<u>80.69</u>	<u>79.43</u>	<u>77.61</u>	<u>75.62</u>	<u>75.45</u>	<u>74.78</u>	<u>80.28</u>	12.44

Table 2. Sota comparison on **CIFAR100**, **mini-ImageNet** and **CUB200**. The best results are **bolded** and the second-best results are underlined. **Avg** is the average accuracy across all sessions, and **PD** is the performance drop rate. Methods marked with \dagger use the ViT model for their backbone, while those marked with \ddagger are prompt-based methods integrated with CLIP.

conducted on two A100 GPUs, and the results are reported as the average of three runs.

4.2. Comparison to state-of-the-art

In this section, we conducted a comparative analysis of the proposed Lark with the latest state-of-the-art methods (as shown in Table 2). Among the compared methods, CLOSER [37] and OrCo [1] both employed a pretraining strategy. Specifically, they first trained the backbone on the base session and then fine-tuned the classification head during the subsequent incremental learning stages to accommodate new classes. This aligns with the usage scenarios of pre-trained visual models. Therefore, we replaced their backbone with ViT-B/16 [9] and reproduced their results.

Lark demonstrates outstanding performance across all three datasets, surpassing most methods in terms of accuracy and forgetting rate. Although its PD is marginally below that of some prompt-based approaches—primarily because they utilize learnable prompt vectors and ancillary semantic information from CLIP—Lark still achieves substantial improvements across diverse backbone archi-

tures (e.g., CLOSER-ViT and OrCo-ViT), indicating its wide applicability and robust stability.

4.3. Ablation Study

In this section, we conduct ablation experiments and visual analyses on the CIFAR100 dataset to verify the effectiveness and characteristics of each component.

Frozen	Localization	Rank-All	Rank-One	Avg \uparrow	PD \downarrow
\checkmark	\times	\times	\times	82.08	19.99
\times	\times	\times	\times	77.33	29.81
\times	\checkmark	\times	\times	81.07	22.32
\times	\checkmark	\checkmark	\times	84.38	16.59
\times	\checkmark	\times	\checkmark	86.36	13.50

Table 3. Ablation study of different modules on **CIFAR100** in this work. The best results are **bolded**. **Avg** is the average accuracy across all sessions, and **PD** is the performance drop rate.

Effectiveness of Localization and Editing. Table 3 presents the performance of each component. Here, **Frozen** indicates whether the backbone is frozen, where a frozen

backbone corresponds to the OrCo-ViT result. **Localization** means that during incremental learning, only the located parameter matrices are fine-tuned. **Rank-All** sums all the update matrices corresponding to each token and then averages them to obtain the final update. **Rank-One** performs singular value decomposition on the sum of all these token-related update matrices, as shown in Equation 10.

It can be observed that when the backbone is unfrozen, the performance is inferior to that of OrCo-ViT. Even when we locate the parameters suitable for learning new knowledge, the model’s performance still does not match that of OrCo-ViT. We believe this is mainly because the distribution of the updated parameter matrices has shifted, causing the backbone’s ability to recognize old classes to decline. After determining the desired low-rank matrix ΔW , both mean processing and singular value decomposition result in a performance that exceeds that of OrCo-ViT. This demonstrates that updating low-rank matrices has the potential to overcome the stability-plasticity dilemma.

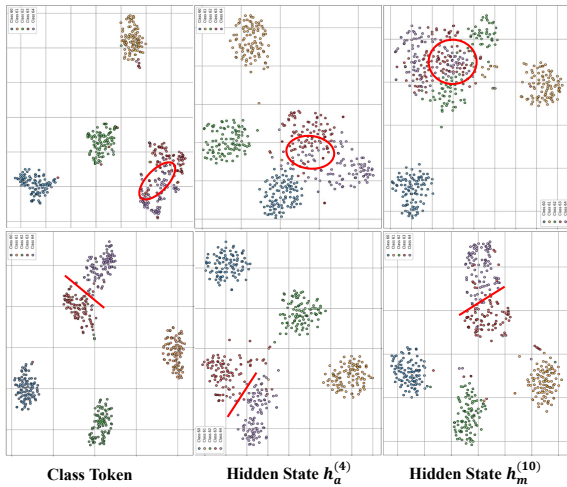


Figure 5. Cluster Distribution on the test set of session 1 of CIFAR100, generated using t-SNE [47]. The top row corresponds to OrCo-ViT, while the bottom row represents Lark in OrCo-ViT. Red circles indicate boundaries that are relatively blurred, while red solid lines indicate boundaries that are relatively clear.

Intermediate Layer Adapted New Knowledge. In Figure 5 we present a comparison of the feature distributions of Class Tokens and hidden states at two different layers in OrCo-ViT and Lark. We observe that in OrCo-ViT (top row), the decision boundaries between certain classes are relatively blurred. For example, in the hidden state $h_m^{(10)}$, the red, green, and purple classes overlap. In contrast, Lark (bottom row) effectively draws clear boundaries among all classes. These clustering plots suggest that although the pre-trained base model possesses some clustering capability when facing new samples, it outperforms the model after localization and editing. Based on the distribution of the class token, we hold that editing identified parameters allows for

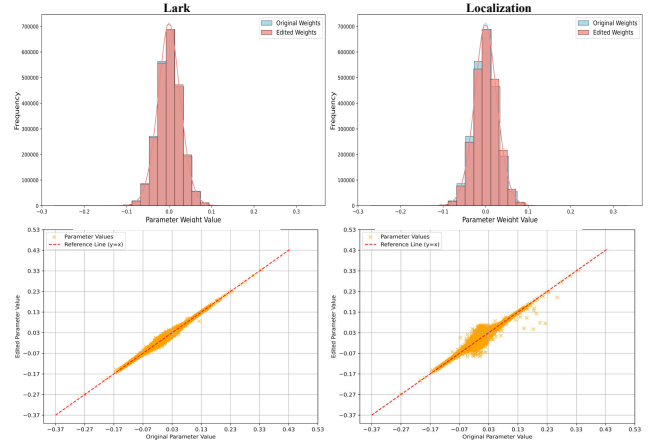


Figure 6. Parameter distribution of the second linear layer in the MLP module of the 10th encoder. The parameter changes under Localization are significantly excessive than those under Lark.

better adaptation to all samples in the new classes.

Smaller perturbations from Rank-One. The purpose of low-rank matrix updates is to minimize perturbations to the original parameters while learning new knowledge. Therefore, we conducted the visual analyses shown in Figure 6. First, we performed histogram statistics on the frequency of parameter occurrences, where the horizontal axis represents the weight values and the vertical axis represents the frequency of different values. It can be observed that in Lark, the distribution of **edited weights** is very close to that of **original weights**. However, in the Localization plot, the distribution of Edited Weights changes significantly, especially in the frequencies of peak and extreme values.

Second, we conducted scatter plot analyses where each point’s horizontal coordinate is the original parameter value and the vertical coordinate is the edited value. Points closer to the reference line indicate smaller degrees of perturbation. It is clearly observed that, compared to the noticeable deviations from the reference line in Localization, most points in Lark are densely distributed near the reference line, forming a relatively narrow band. In summary, the contrasts in the histograms and scatter plots demonstrate that Lark causes minimal perturbation to the original parameter distribution while learning new knowledge.

4.4. Analysis of Low-Rank Matrix Update

We further analyze the effectiveness of low-rank matrix updates from two complementary perspectives: selective fine-tuning and the determination of an optimal rank. First, we investigate the impact of selectively fine-tuning individual matrices Query (Q), Key (K) and Value (V) within the MHSA. As shown in the right side of Table 4, fine-tuning only the V matrix consistently yields the highest performance. This advantage arises because updates to Q and K significantly alter attention distributions, causing interfer-

Datasets	Metrics	Q	K	V	Q, K, V	1	4	16	64
CIFAR100	Avg	84.45	84.40	86.03	84.93	86.03	85.99	85.99	85.85
	PD	10.43	10.51	7.74	11.27	7.74	7.60	7.91	7.95
CUB200	Avg	80.31	79.93	82.00	80.09	82.00	81.91	81.83	81.72
	PD	12.76	13.37	9.77	13.81	9.77	9.63	9.57	9.49
mini-ImageNet	Avg	85.19	84.72	88.15	84.30	88.15	88.09	87.86	87.63
	PD	12.83	13.33	9.43	14.49	9.43	9.40	9.51	9.57

Table 4. The left side of the table shows results obtained by editing different matrices in MHSA, while the right side shows results obtained using different ranks.

ence with previously learned classes. Conversely, the V matrix simply transforms features, effectively balancing model stability and plasticity. Jointly fine-tuning all three matrices leads to notable performance degradation, highlighting the critical role of selective tuning.

Second, inspired by the Eckart-Young-Mirsky theorem, we validate the choice of rank-1 for low-rank approximations. Theoretically, rank-1 outer products minimize interference with prior knowledge by providing the optimal low-rank representation of weight updates (ΔW) under the Frobenius norm. Practically, higher ranks increase parameter count and memory usage, conflicting with FSCIL’s lightweight learning objectives. Results on the left side of Table 4 support rank-1 as the optimal choice, achieving consistently superior or competitive average accuracy and performance deterioration across all datasets.

4.5. Lark in other task

The main challenge faced by hand keypoint detection tasks is that obtaining sufficient labeled real-world data is both labor-intensive and time-consuming, leading many studies to rely on synthetic datasets. These works [24, 62] pre-train models on synthetic datasets and then transfer the pre-trained models to real-world scenarios, effectively overcoming the challenges posed by data scarcity. Based on this, in this section, we construct a few-shot incremental learning scenario for hand keypoint detection tasks to validate the effectiveness of the proposed method. Specifically, we use RenderedHandPose [62] (RHD) as the base session, with Hand3DStudio [58] (H3D) and FreiHand [63] (FHD) serving as incremental sessions. Similar to FSCIL tasks, the training data for the incremental sessions consist of only a small number of samples, while the test samples include the entire test set data (Details in Section 7.5 in supplementary).

We conducted comparative experiments in Table 5, where Global denotes allowing all parameters to be updated in incremental sessions, and Frozen refers to freezing the backbone, only allowing updates to the output head. It is evident that, Lark achieves significant advantages in both Avg and PD metrics. Specifically, for Avg, Lark shows an improvement of 1.94% over the second-best result, while for PD, it reduces the value by 3.11% (more experiments are in Section 7.6 of the supplement.).

Additionally, we visualized the keypoint detection capa-

Method	Acc. in each session (%) \uparrow			Avg \uparrow	PD \downarrow
	Base	1	2		
Global	65.64	52.73	53.65	57.34	11.99
Frozen	65.64	52.34	54.21	57.39	11.43
Lark (Ours)	65.64	55.04	57.32	59.33	8.32

Table 5. Performance comparison on keypoint detection tasks. The best results are **bolded**. Avg is the average accuracy across all sessions, and PD is the performance drop rate.

bilities of Global, Frozen, and Lark in Figure 7. Overall, the advantages clearly demonstrate that the proposed method of first locating parameters and then performing low-rank matrix updates can effectively learn new knowledge while retaining memory of the old knowledge.

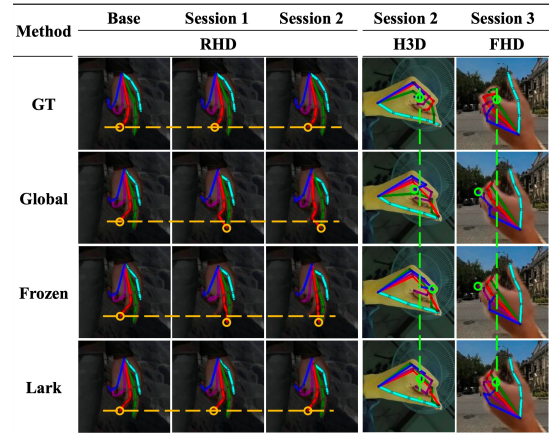


Figure 7. Keypoint detection results under different methods. The yellow horizontal lines measure the shift degree of the same keypoint across different incremental sessions. The green vertical lines measure the shift of new knowledge relative to the ground truth across various methods.

5. Conclusion

In this paper, we propose Lark, a method that performs low-rank matrix updates after knowledge localization. By locating knowledge, the method identifies parameters suitable for learning new knowledge, which can prevent misalignment between the backbone and the classifier. Moreover, under the constraint of the Rank-One matrix, concerns about excessive updates to parameter weights are alleviated. Experimental results on three datasets demonstrate that our proposed method achieves clear advantages over state-of-the-art approaches. Additionally, experiments conducted on hand keypoint detection tasks further illustrate that Lark is a generalizable approach, suitable for incremental learning tasks in various few-shot scenarios.

In future work, we will investigate a generalized few-shot incremental learning method applicable to more visual tasks, including segmentation and detection. Furthermore, time and space efficiency should be evaluated to ensure the method’s suitability for large-scale visual models.

References

- [1] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28762–28771, 2024. 1, 2, 5, 6
- [2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. 2
- [3] James R Bunch and Christopher P Nielsen. Updating the singular value decomposition. *Numerische Mathematik*, 31(2):111–129, 1978. 2
- [4] James R Bunch, Christopher P Nielsen, and Danny C Sorensen. Rank-one modification of the symmetric eigenproblem. *Numerische Mathematik*, 31(1):31–48, 1978. 2
- [5] Lucas Caccia, Rahaf Aljundi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. Reducing representation drift in online continual learning. *arXiv preprint arXiv:2104.05025*, 1(3), 2021. 1
- [6] Yujun Cai, Lihao Ge, Jianfei Cai, and Junsong Yuan. Weakly-supervised 3d hand pose estimation from monocular rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 666–682, 2018. 4
- [7] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafcil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14166–14175, 2022. 1, 2
- [8] Marco D’Alessandro, Alberto Alonso, Enrique Calabrés, and Mikel Galar. Multimodal parameter-efficient few-shot class incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3393–3403, 2023. 6
- [9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*, 2021. 5, 6
- [10] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, 2021. 5
- [11] Yangyang Guo, Guangzhi Wang, and Mohan Kankanhalli. Pela: Learning parameter-efficient models with low-rank approximation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15699–15709, 2024. 2
- [12] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, et al. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, 2021. 2, 3
- [13] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12963–12971, 2021. 4
- [14] Benjamin Heinzerling and Kentaro Inui. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, 2021. 5
- [15] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*. 2, 3
- [16] Dongwan Kim and Bohyung Han. On the stability-plasticity dilemma of class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20196–20204, 2023. 1
- [17] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *International Conference on Learning Representations*, 2023. 2, 3, 6
- [18] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 2, 3
- [19] Frank Kleibergen and Richard Paap. Generalized reduced rank tests using the singular value decomposition. *Journal of econometrics*, 133(1):97–126, 2006. 2
- [20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5
- [21] Richard Kueng, Holger Rauhut, and Ulrich Terstiege. Low rank matrix recovery from rank one measurements. *Applied and Computational Harmonic Analysis*, 42(1):88–116, 2017. 2
- [22] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9020–9029, 2021. 1
- [23] Yann LeCun, John Denker, and Sara Solla. Optimal brain damage. *Advances in Neural Information Processing Systems*, 2, 1989. 3
- [24] Lijun Li, Linrui Tian, Xindi Zhang, Qi Wang, Bang Zhang, Liefeng Bo, Mengyuan Liu, and Chen Chen. Renderih: A large-scale synthetic dataset for 3d interacting hand pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20395–20405, 2023. 8, 4
- [25] Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 18564–18572, 2024. 2, 4
- [26] Xiaojie Li, Yibo Yang, Jianlong Wu, Jie Liu, Yue Yu, Liqiang Nie, and Min Zhang. Continuous knowledge-preserving decomposition for few-shot continual learning. *arXiv preprint arXiv:2501.05017*, 2025. 3

- [27] Zexi Li, Xinyi Shang, Rui He, Tao Lin, and Chao Wu. No fear of classifier biases: Neural collapse inspired federated learning with synthetic and fixed classifier. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5319–5329, 2023. 2
- [28] Chenxi Liu, Zhenyi Wang, Tianyi Xiong, Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Few-shot class incremental learning with attention-aware self-adaptive prompt. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024. 3
- [29] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Proceedings of the European Conference on Computer Vision*, pages 146–162. Springer, 2022. 1
- [30] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024. 2
- [31] Xialei Liu, Marc Masana, Luis Herranz, Joost Van de Weijer, Antonio M Lopez, and Andrew D Bagdanov. Rotate your networks: Better weight consolidation and less catastrophic forgetting. In *International Conference on Pattern Recognition*, pages 2262–2268, 2018. 3
- [32] Ivan Markovskiy. Structured low-rank approximation and its applications. *Automatica*, 44(4):891–909, 2008. 3
- [33] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2337–2345, 2021. 2
- [34] Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*. 3, 5
- [35] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35: 17359–17372, 2022. 2, 4, 5
- [36] Martial Mermillod, Aurélie Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013. 1
- [37] Junghun Oh, Sungyong Baik, and Kyoung Mu Lee. Closer: Towards better representation learning for few-shot class-incremental learning. In *Proceedings of the European Conference on Computer Vision*, 2024. 1, 2, 5, 6
- [38] Zicheng Pan, Weichuan Zhang, Xiaohan Yu, Miaohua Zhang, and Yongsheng Gao. Pseudo-set frequency refinement architecture for fine-grained few-shot class-incremental learning. *Pattern Recognition*, page 110686, 2024. 1, 2
- [39] Keon-Hee Park, Kyungwoo Song, and Gyeong-Moon Park. Pre-trained vision and language transformers are few-shot incremental learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23881–23890, 2024. 2, 3
- [40] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open set perspective. In *Proceedings of the European Conference on Computer Vision*, pages 382–397, 2022. 2
- [41] Wenhao Qiu, Sichao Fu, Jingyi Zhang, Chengxiang Lei, and Qinmu Peng. Semantic-visual guided transformer for few-shot class-incremental learning. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2885–2890. IEEE, 2023. 6
- [42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 5
- [43] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*. 2
- [44] Jinxin Shi, Jiabao Zhao, Xingjiao Wu, Ruyi Xu, Yuan-Hao Jiang, and Liang He. Mitigating reasoning hallucination through multi-agent collaborative filtering. *Expert Systems with Applications*, 263:125723, 2025. 2
- [45] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153, 2017. 3
- [46] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020. 2, 3, 5
- [47] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 7
- [48] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5
- [49] Shipeng Wang, Xiaorong Li, Jian Sun, and Zongben Xu. Training networks in null space of feature covariance for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 184–193, 2021. 2
- [50] Xuan Wang, Zhong Ji, Yunlong Yu, Yanwei Pang, and Jun-gong Han. Model attention expansion for few-shot class-incremental learning. *IEEE Transactions on Image Processing*, 2024. 6
- [51] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys*, 53(3):1–34, 2020. 1
- [52] Zhihang Wei, Jinxin Shi, Jing Yang, and Jiabao Zhao. Vip-fscil: A more robust approach for fscil. In *2024 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2024. 2
- [53] Yibo Yang, Shixiang Chen, Xiangtai Li, Liang Xie, Zhouchen Lin, and Dacheng Tao. Inducing neural collapse in imbalanced learning: Do we really need a learnable classifier at the end of deep neural network? *Advances in Neural Information Processing Systems*, 35:37991–38002, 2022. 1, 2

- [54] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32, 2019. 4
- [55] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12455–12464, 2021. 2, 6
- [56] Jiabao Zhao, Yifan Yang, Xin Lin, Jing Yang, and Liang He. Looking wider for better adaptive representation in few-shot learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 10981–10989, 2021. 1
- [57] Yifan Zhao, Jia Li, Zeyin Song, and Yonghong Tian. Language-inspired relation transfer for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 6
- [58] Zhengyi Zhao, Tianyao Wang, Siyu Xia, and Yangang Wang. Hand-3d-studio: A new multi-view system for 3d hand reconstruction. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2478–2482, 2020. 8, 4
- [59] Yinqiang Zheng, Guangcan Liu, Shigeki Sugimoto, Shuicheng Yan, and Masatoshi Okutomi. Practical low-rank matrix approximation under robust l_1 -norm. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1417. IEEE, 2012. 3
- [60] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12816–12831, 2022. 6
- [61] Jiancai Zhu, Jiabao Zhao, Jiayi Zhou, Liang He, Jing Yang, and Zhi Zhang. Uncertainty-aware few-shot class-incremental learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 1
- [62] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4903–4911, 2017. 8, 4
- [63] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 813–822, 2019. 8, 4