Feature Decomposition-Recomposition in Large Vision-Language Model for Few-Shot Class-Incremental Learning

Zongyao Xue[1,2], Meina Kan[1,2], Shiguang Shan [1,2,3], Xilin Chen[1,2]
[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100090, China
[2]University of Chinese Academy of Sciences, Beijing, 100090, China
[3]Peng Cheng Laboratory, Shenzhen, 518055, China

Abstract

Few-Shot Class-Incremental Learning (FSCIL) focuses on incrementally learning novel classes using only a limited number of samples from novel classes, which faces dual challenges: catastrophic forgetting of previously learned classes and over-fitting to novel classes with few available samples. Recent advances in large pre-trained vision-language models (VLMs), such as CLIP, provide rich feature representations that generalize well across diverse classes. Therefore, freezing the pre-trained backbone and aggregating class features as prototypes becomes an intuitive and effective way to mitigate catastrophic forgetting. However, this strategy fails to address the overfitting challenge, and the prototypes of novel classes exhibit semantic bias due to the few samples per class. To address these limitations, we propose a semantic ***Feature Decomposition-Recomposition (FDR)*** *method based on VLMs. Firstly, we decompose the CLIP features into semantically distinct segments guided by text keywords from base classes. Then, these segments are adaptively recomposed at the attribute level given text descriptions, forming calibrated prototypes for novel classes. The recomposition process operates linearly at the attribute level but induces nonlinear adjustments across the entire prototype. This fine-grained and non-linear recomposition inherits the generalization capabilities of VLMs and the adaptive recomposition ability of base classes, leading to enhanced performance in FSCIL. Extensive experiments demonstrate our method's effectiveness, particularly in 1-shot scenarios where it achieves improvements between 6.70% and 19.66% for novel classes over previous prototype-based methods on CUB200. Code is available at https://github.com/herb1999/FDR.*

1. Introduction

The remarkable success of deep neural networks (DNNs) is largely based on training with large-scale static datasets [23]. However, real-world scenarios such as autonomous driving and robotic field operations often involve continuously arriving data streams, where new categories emerge incrementally with only sparse samples available. Although humans excel at adapting to such dynamic environments, existing learning-based algorithms struggle under these constraints. To bridge this gap, Few-Shot Class-Incremental Learning (FSCIL)[22] is introduced as a more practical and challenging task, aiming to incrementally recognize novel classes from few-shot examples while preserving performance on base classes. The FSCIL task inherits dual challenges from few-shot learning and continual learning. First, models must overcome the unreliable empirical risk minimization problem [26] when generalizing with scarce novel class samples. Second, they must maintain stability against catastrophic forgetting [16] of base classes while adapting to incremental updates.

Since its introduction in 2020 [22], FSCIL has attracted significant attention due to its potential in real-world applications, leading to a variety of early methodological developments, including meta learning[9, 31, 33], specialized model architectures[28, 29], optimization strategies [19] and replay-based methods[10, 15]. However, these approaches face a longstanding plasticity-stability dilemma, wherein models struggle to simultaneously preserve base class knowledge while effectively accommodating novel class acquisition.

Recently, leveraging the generalizable representations of pre-trained models such as CLIP, prototype-based metric learning approaches [1, 21, 30, 32] have emerged as a prominent solution to mitigate catastrophic forgetting. These methods typically freeze the pre-trained backbone networks and aggregate features from samples within each class to construct prototypes, enabling effective knowledge preservation and incremental learning. However, the scarce samples of novel classes hinder the full exploitation of features from large pre-trained models, yielding semantically biased novel prototypes that deviate from true class distributions. To address these biases, existing methods [1, 7, 10] leverage supplementary information from base class prototypes to calibrate novel ones. Nevertheless, these approaches directly employ entire similar base prototypes to compensate for the biased novel ones, lacking mechanisms to identify which specific components are transferable. This oversight leads to reduced inter-class distances between novel and base classes, causing feature space entanglement.

To address these limitations, we propose a **Feature Decomposition-Recomposition (FDR)** framework that enables fine-grained prototype calibration through attribute-level knowledge transfer. *Our key insight is that although the feature representations from large pre-trained models contain sufficient information for classifying diverse classes, their correct classification depends on strategically extracting generalizable features, particularly for few-shot classes.* Specifically, given features from pre-trained VLMs like CLIP, we first decompose the visual prototypes of base classes into semantic-aware feature segments by establishing mappings between CLIP visual features and attribute keywords through a keyword grounding function. These semantic feature segments are transferable across classes. Next, for novel classes, given a set of keywords describing them, each corresponding feature segment of the biased prototype derived from few-shot samples is calibrated by recombining segments from similar base class prototypes (e.g., calibrating the **"cannon"** prototype by combining **"wheel"** segments from classes like bicycles and **"cylindrical"** segments from classes like lipsticks). The combinatorial possibilities for synthesizing novel prototypes grow exponentially as the number of base classes increases, effectively addressing the prototype entanglement problem.

The contributions of this work are threefold:

1. We propose a semantic feature decomposition and recomposition method, FDR, that divides CLIP's frozen features into reusable attribute-level feature segments, enabling precise and easy knowledge transfer from base classes to novel ones for enhanced incremental learning.
2. We design a keyword grounding function to decompose CLIP's frozen features into semantic-aware segments, and introduce a segment-wise recomposition method to calibrate biased prototypes at the fine-grained attribute-level.
3. Our approach is training-free, simple yet highly effective, achieving improvements of 6.70%-19.66% in novel class accuracy over state-of-the-art methods, showcasing its practical utility.

2. Related Work

Incremental learning (IL), also referred to as continual or lifelong learning, distinguishes itself from conventional machine learning by addressing dynamic data distributions where models must continually integrate new knowledge from sequential tasks while preserving performance on previously learned ones [25]. The core challenge lies in balancing learning plasticity for new knowledge and memory stability for old knowledge [16]. An intuitive approach to ensure the stability of old knowledge is to retain representative samples from previous tasks and replay them during training on new tasks[8, 13, 18]. However, growing memory demands and privacy concerns limit their scalability. To enhance storage efficiency, retaining features of old samples and replaying them later is also an option[12]. Another method for preserving old knowledge is generative replay[20, 27], where a generative model learns the distribution of old data and replays generated samples or features. These methods primarily focus on preserving old knowledge rather than leveraging it to enhance new task learning.

Few-Shot Learning (FSL) requires models to learn from a limited number of labeled samples, making it suitable for real-world scenarios where data is scarce due to safety, privacy, or ethical concerns[26]. Metric-based methods [14, 21, 24] have proven to be simple and effective solutions by learning transferable embedding spaces. Prototypical Networks[21], for instance, define a metric space where classification is performed by computing distances to prototype representations of each class. These prototypes, computed as class-mean feature vectors, naturally accommodate incremental updates by freezing the embedding backbone – a property that makes them particularly suitable for few-shot class-incremental learning [22]. However, prototype quality degrades significantly under extreme data scarcity (e.g., 1-shot settings), as limited samples fail to capture true class distributions, leading to biased prototypes that misalign with semantic realities.

Few-Shot Class-Incremental Learning (FSCIL) aims to incrementally learn novel classes given only its few shots[22], which merges challenges from IL and FSL: preventing catastrophic forgetting of base classes while generalizing to novel classes with sparse samples. A common strategy to mitigate this is to use semantically similar base classes to correct biases in novel classes[1, 7, 10]. Additionally, incorporating side information from language has been explored to enhance model performance[7, 10]. The closest to our work is [1], which reduces novel prototype bias by globally calibrating them with similar base prototypes. However, such coarse-grained adjustments compress inter-class distances, leading to prototype entanglement in later phases. Our method addresses these limitations through fine-grained calibration, decomposing prototypes into semantically meaningful sub-features for targeted composition. This enables precise inheritance of relevant visual attributes from multiple base classes while maintaining inter-class separability. The integration of language semantics also resolves ambiguity in visual similarity measurements, particularly for classes sharing partial attributes rather than global appearances.

3. Background

Following the existing works [1, 7, 10, 22], the Few-Shot Class-Incremental Learning (FSCIL) task is defined as a process consisting of $(1 + T)$ sequential learning sessions, which are divided into two consecutive phases:

- **Base phase** (the first session S_0): during S_0, the model is trained on M base classes $\mathcal{C}^0 = \{C_1, ..., C_M\}$ with abundant samples per class.
- **Incremental phase** (sessions $S_1 \sim S_T$): following the N-way K-shot protocol, at each session $S_t|_{t=1}^{T}$, the model learns N novel classes with K samples per class (typically $K \leq 5$). The novel classes are denoted as $\mathcal{C}^t = \{C_{M^t+1}, ..., C_{M^t+N}\}$ with $M^t = M + (t-1) \times N$. The label spaces across distinct sessions remain strictly disjoint, i.e. $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset \quad \forall i, j \in \{0, ..., T\}, i \neq j$.

The model in few-shot class-incremental learning is expected to maintain classification accuracy across all base classes while achieving precise classification for novel classes. When leveraging a large pre-trained model as the feature extractor, it can extract rich and informative features for both base and novel classes attributing to its generalization capability. Consequently, obtaining a well-defined prototype is often sufficient for classification.

For base classes, prototypes can be easily derived by aggregating abundant samples, as demonstrated by existing methods like [21]. Specifically, each class C is represented by a prototype vector $\mathbf{p}_c \in \mathbb{R}^d$, computed as the mean feature of all samples belonging to that class:

$$\mathbf{p}_c = \frac{1}{|\mathcal{I}_c|} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{I}_c} f_\theta(\mathbf{x}_i), \tag{1}$$

where f_θ denotes the frozen feature extractor (large pre-trained model like CLIP in this work), and $\mathcal{I}_c$ contains all available samples for class C.

For novel classes at session S_t, the prototypes can also be derived from few-shot samples following Eq. (1). However, these prototypes are often biased due to the limited number of training examples. Such bias can result in sub-optimal classification performance, presenting a significant challenge in FSCIL.

During inference at session S_t, the classification probability for a test sample $\mathbf{x}$ is calculated by measuring the distance between its feature representation $f_\theta(\mathbf{x})$ and the prototype vectors of all classes encountered (i.e. $\bigcup_{i=0}^{t} \mathcal{C}^i$) so far.

4. Method

To mitigate semantic bias in novel-class prototypes derived from scarce samples, we propose a semantically feature decomposition-recomposition approach for calibrating the prototype in a fine-grained manner at keyword attribute level. An overview is illustrated in Fig. 1.

Our proposed method consists of two stages: 1) **semantic feature decomposition stage**, which establishes a grounding of text attribute keywords to CLIP features using base classes. It produces a set of meaningful feature segments, each corresponding to a specific semantic keyword. These feature segments are transferable across classes, ensuring their help to novel classes. 2) **Feature segments recomposition stage**, which adaptively calibrates the prototypes of novel classes by recomposing the semantic-aware feature segments according to the keywords associated with the novel classes. This process enables fine-grained refinement of the prototypes at keyword attribute level, mitigating bias and improving classification performance.

4.1. Semantic feature decomposition

In this stage, for each base class C, we begin by generating visual descriptions and attribute keywords using Large Language Models (LLMs). Next, leveraging the vision-language alignment capabilities of CLIP models, a fine-grained mapping from these keywords to visual features is established, producing a set of semantically meaningful feature segments.

Firstly, with the help of LLMs, a textual description $\mathcal{D}_c$ for each class C is generated conditioned on class names (*e.g.* "toaster"). For instance, $\mathcal{D}_c$ might be "A rectangular kitchen appliance with metal or plastic housing, slots for inserting bread, and controls like dials/buttons." Additionally, m attribute keywords $\mathcal{K}_C = \{k_1, ..., k_m\}$ are extracted from $\mathcal{D}_c$. Each keyword k_i describes a visual attribute of the class C (*e.g.* "rectangular" and "slots" for class "toaster"). These textual descriptions and keywords capture the semantic attributes of the class.

Secondly, we design a keyword grounding function $\mathcal{G}_\phi$ to establish fine-grained mappings from these keywords to visual feature segments of base prototypes. Leveraging CLIP's frozen visual encoder f_θ and textual encoder g_ϕ , we obtain aligned visual prototype $\mathbf{p}_c$ (same as Eq. (1)) and textual representations $\mathbf{t}_c = g_\phi(\mathcal{D}_c)$, where $\mathbf{t}_c \in \mathbb{R}^d$ denotes the text embedding of description $\mathcal{D}_c$. Since CLIP's text and image embeddings are aligned, mapping a keyword k_i to a segment of the text feature space inherently implies a corresponding mapping to the associated segment in the image feature space. Therefore, for each keyword $k_i \in \mathcal{K}_C$, the keyword grounding function $\mathcal{G}_\phi$ generates a grounding vector $\mathbf{g}_i$ by evaluating the relevance of the keyword to the entire textual representation:

$$\mathbf{g}_i = \mathcal{G}_\phi(k_i, \mathcal{D}_c), \in \{0, 1\}^d. \tag{2}$$

Here $\mathbf{g}_i$ is a sparse binary mask, where entries with a value of 1 indicate the dimensions at which the keyword k_i contributes most significantly to the text embedding $\mathbf{t}_c$ of description $\mathcal{D}_c$.

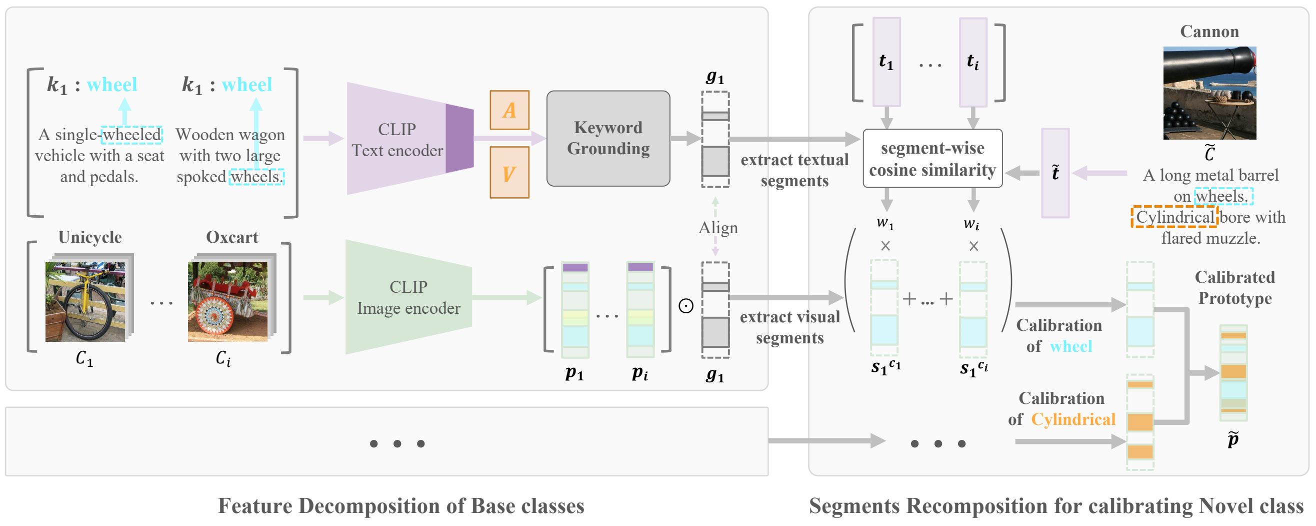

k_1 : wheel
k_1 : wheel
A single-wheeled vehicle with a seat and pedals.
Wooden wagon with two large spoked wheels.
CLIP Text encoder
A
V
Keyword Grounding
g_1
extract textual segments
Align
Unicycle
Oxcart
C_1
C_i
CLIP Image encoder
p_1
p_i
g_1
extract visual segments
t_1
t_i
segment-wise cosine similarity
$\tilde{t}$
Cannon
$\tilde{C}$
A long metal barrel on wheels. Cylindrical bore with flared muzzle.
w_1
w_i
$s_1^{c_1}$
$s_1^{c_i}$
Calibration of wheel
Calibration of Cylindrical
Calibrated Prototype
$\tilde{p}$
Feature Decomposition of Base classes
Segments Recomposition for calibrating Novel class

Figure 1. **Overall Architecture of Our Feature Decomposition-Recomposition (FDR) method.** The calibration process is as follows: 1) **Feature Decomposition of Base classes**: For all classes, LLMs are leveraged to generate category descriptions and extract semantic keywords. CLIP encodes both image features (aggregated as prototypes) and textual descriptions. A Keyword Grounding Function then decomposes textual embeddings into semantic-aware feature segments aligned with specific keywords, achieving attribute-level decomposition of visual and textual features. 2) **Segments Recomposition for calibrating Novel class**: For novel classes, we match each keyword-derived textual feature segment with semantically similar segments from base classes. The corresponding visual feature segments from multiple base prototypes are recomposed to calibrate the raw novel prototype. For instance, when calibrating the cannon prototype (which suffers from missing wheel features due to limited samples), our method identifies wheeled segments from unicycle and oxcart prototypes and cylindrical segments from lipstick and water tower prototypes. This fine-grained recomposition supplements missing attributes while preserving class-discriminative characteristics.

During the text encoding process of CLIP, the embedding of the entire text description $\mathcal{D}_c$ inherently incorporates the contributions of tokens from each word including those visual keywords $k_i \in \mathcal{K}_C$. This means that the attention mechanisms and other internal structures implicitly reveal how and where each keyword contributes to the overall textual representation. Therefore, the keyword grounding function $\mathcal{G}_\phi$ is designed to leverage CLIP's internal attention mechanisms to identify these contributions.

Specifically, in the last transformer layer of CLIP textual encoder g_ϕ, $\mathcal{D}_c$ is tokenized as n tokens $\{[SOS], token_1, ..., token_{n-2}, [EOS]\}$. Given the attention matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and value matrix $\mathbf{V} \in \mathbb{R}^{d\times n}$, $\mathbf{t}_c$ is calculated as the output of the last transformer layer as follows:

$$\mathbf{t}_c = \mathbf{V}(:,:)\mathbf{A}(:,eos) + \mathbf{h} = (a_1\mathbf{v}_1 + ... + a_n\mathbf{v}_n) + \mathbf{h}, \quad (3)$$

where $\mathbf{A}$ represents the inter-token attention weights, $\mathbf{V}$ represents transformed token representations of all words, $a_i = A[i, p_{\text{eos}}]$ represents the attention weight from the i-th token to the [EOS] token, and $\mathbf{v}_i$ means the value vector of the i-th token from CLIP's final transformer layer.

As can be seen from the calculation of $\mathbf{t}_c$ in Eq. (3), the attention matrix $\mathbf{A}$ and value matrix $\mathbf{V}$ implicitly reveal how and where each keyword contributes to the overall textual representation. Therefore, we extract these token-wise contribution values as a contribution indication matrix $\mathbf{B} \in \mathbb{R}^{d\times n}$:

$$\mathbf{B} = [a_1\mathbf{v}_1, \ \ldots, \ a_n\mathbf{v}_n]. \quad (4)$$

Here, the columns of $\mathbf{B}$ describe different text tokens, while the rows represent feature channels. Each element $\mathbf{B}(j,i)$ indicates the contribution of the i-th token's j-th feature dimension to the final [EOS] embedding, serving as an effective indicator for keyword-related feature dimensions.

Therefore, to quantify the influence of each token on the text embedding $\mathbf{B}$, we apply a softmax operation along the token dimension (horizontal) of $\mathbf{B}$. This results in a matrix $\mathbf{C}$, where each element represents the proportion of $token_i$'s contribution to $\mathbf{t}_c$ at feature dimension j, relative to all text tokens:

$$\mathbf{C}(i,:) = Softmax(\mathbf{B}(i,:)) \quad (5)$$

Then, we perform top-q binarization along the feature dimension (vertical) of $\mathbf{B}$, converting each column $\mathbf{C}[:,i]$ into into a sparse binary mask $\mathbf{g}_i$ (previously introduced in Eq. (2)) as follows:

$$\mathbf{g}_i(j) = \begin{cases} 1, & \text{if } \mathbf{C}(i,j) \text{ is within top-}q \text{ responsive of } \mathbf{C}(:,j) \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

This process isolates the most significant feature dimensions associated with each keyword, enabling grounding of textual keywords to text feature segments.

According to Eqs. (2) to (6), we can derive a fine-grained mapping from each keyword to its corresponding feature segments. While this mapping is computed independently for each class, ideally, the same keyword associated with different classes should yield identical mappings. However, in practice, slight discrepancies often arise due to variations in class contexts. To ensure consistency, for keywords that appear across multiple base classes, we unify all selected feature dimensions by taking the union of their respective binary masks. This strategy preserves all potentially relevant dimensions for the keyword, enhancing robustness and consistency in the grounding process.

Since CLIP's visual and textual feature spaces are aligned, given a keyword k_i, we can obtain the visual feature segment $\mathbf{s}_i^c$ of prototype $\mathbf{p}_c$, by simply performing an element-wise multiplication as follows:

$$\mathbf{s}_i^c = \mathbf{p}_c \odot \mathbf{g}_i. \tag{7}$$

In this way, we achieve semantic-aware decomposition of base class prototypes guided by textual keywords of visual attributes, where each feature segment $\mathbf{s}_i^c$ retains only the dimensions most relevant to the corresponding semantic keyword.

4.2. Feature segments recomposition for fine-grained calibration

The prototypes of novel classes are often biased due to the limited number of shots available per class. While the full-dimensional features of these prototypes can only be derived from the novel class samples, certain partial features—such as visual attribute features—frequently overlap with those of base class prototypes. For instance, features like "wheels" are shared between the novel class "cannon" and base classes "unicycle", "oxcart", and "race car". Leveraging this inherent property, we calibrate the prototypes of novel classes by piecewise refinement of their semantic feature segments. This is achieved by ensembling similar segments from the base classes, thereby enhancing the representational accuracy and reducing bias in the novel class prototypes.

In this process, given any novel class $\widetilde{C}$, we first generate its initial prototype $\mathbf{p}_{\widetilde{c}}$ and visual keywords $\mathcal{K}_{\widetilde{C}}$ following the same process as base classes.

Next, for each keyword $k_i \in \mathcal{K}_{\widetilde{C}}$, we retrieve the corresponding feature segments $\mathbf{s}_i^{c_j}$ from the base-class prototypes that share the same keywords. These segments are then combined using a weighted summation to calibrate the feature segment of the novel class:

$$\mathbf{r}_i = \sum\nolimits_{j=1}^{M} w_j \cdot \mathbf{s}_i^{c_j}. \tag{8}$$

Here, the $\mathbf{s}_i^{c_j}$ represents the prototype segment of base class C_j corresponding to the keyword k_i. The weight w_j reflects

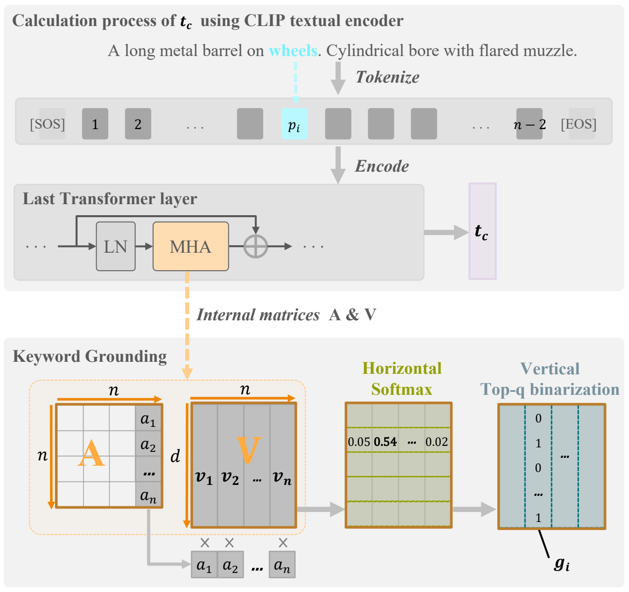


Figure 2. Implementation of keyword grounding function $\mathcal{G}_\phi$. The upper describes the computation of $\mathbf{t}_c$ using CLIP, while the lower details the process for generating the grounding vector $\mathbf{g}_c$.

the similarity between the feature segments of $\widetilde{C}$ and base class C_j. This similarity is approximated by computing the textual embedding similarity between the textual segment of novel class $\mathbf{t}_{\widetilde{c}} \odot \mathbf{g}_i$ and that of every base class $\mathbf{t}_{c_j} \odot \mathbf{g}_i, C_j \in C^0$. The top-r similarity scores are preserved after using a softmax function, while the remaining similarity scores are set to zero. This ensures that only the most semantically relevant base-class feature segments are retained, yielding the final weights w_j.

After obtaining feature segments for all keywords associated with novel class $\widetilde{C}$, keyword-specific adjustments are iteratively applied to novel prototype $\mathbf{p}_{\widetilde{c}}$ to obtain the final calibrated prototype as below:

$$\mathbf{p}_{\widetilde{c}} \leftarrow (1-\alpha)\cdot \mathbf{p}_{\widetilde{c}} + \alpha \cdot \mathbf{r}_i, \quad i \in \{1, ..., m\} \tag{9}$$

where α is a trade-off hyperparameter that balances the prototype directly from the few shots and calibrated one.

In Eq. (9), each calibration $\mathbf{r}_i$ is performed individually and is a linear combination of similar feature segments from base-class prototypes. Importantly, this linear combination varies for different feature segments, as the weights and contributing segments depend on the specific keyword k_i. *Consequently, the overall calibration is semantic-segment-wise linear but nonlinear overall, ensuring fine-grained and more accurate prototype calibration for few-shot novel classes.*

Table 1. Comparison with existing methods (5-shot setting). A_{base} for base session accuracy, A_{avg} for AvgAcc across all sessions

Method	Backbone	Training Free	CIFAR100		miniImageNet		CUB200		IN1K-FSCIL	
			A_{base}	A_{avg}	A_{base}	A_{avg}	A_{base}	A_{avg}	A_{base}	A_{avg}
CLOSER[2]	ResNet-18/20	No	75.7	63.1	76.0	62.8	79.4	69.1	-	-
Yourself[6]	ViT-B-IN21K	No	82.9	67.0	84.0	68.8	83.4	69.9	-	-
LP-DiF[4]	ViT-B-CLIP	No	80.2	75.1	96.3	93.8	83.9	74.0	-	-
PriViLege[5]	ViT-B-CLIP	No	84.2	77.2	95.4	92.7	82.1	74.2	72.1	68.9
EASE[3]	ViT-B-IN21K	No	85.1	78.0	**96.8**	**93.9**	83.0	75.1	73.3	70.4
FDR(Ours)	ViT-B-CLIP	No	**87.7**	**79.3**	95.3	89.2	**84.4**	**77.7**	**73.6**	**70.9**
Baseline	ViT-B-CLIP	Yes	75.7	67.9	92.0	88.6	80.9	73.6	69.2	65.9
TEEN[1]	ViT-B-CLIP	Yes	75.7	68.4	92.0	88.7	80.9	73.7	69.2	66.1
FDR(Ours)	ViT-B-CLIP	Yes	**75.7**	**68.5**	**92.0**	**89.1**	**80.9**	**74.1**	**69.2**	**66.4**

Table 2. Comparison with existing methods on **CUB200-1-shot** benchmark (Accuracy %). **Bold** indicates best performance. Δ shows improvement of our method over the previous best. Due to space limitations, only the first three and last three sessions are shown.

Session	0	1		2		3		8		9		10	
Method	AvgAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc
Baseline [21]	80.85	74.18	14.25	68.86	18.66	65.00	18.39	52.12	18.22	51.64	20.30	49.75	19.86
TEEN [1]	80.85	76.40	35.16	71.89	38.26	66.70	30.00	56.60	29.75	56.14	32.19	54.48	31.56
FDR(Ours)	**80.85**	**77.26**	**54.82**	**72.93**	**50.09**	**68.38**	**39.74**	**58.33**	**36.45**	**58.76**	**38.91**	**57.37**	**38.40**
Δ	-	+0.86	+19.66	+1.04	+11.83	+1.68	+9.74	+1.73	+6.70	+2.62	+6.72	+2.89	+6.84

5. Experiments

This section begins by detailing the datasets and implementation specifics. Subsequently, it presents a comparison with existing approaches to benchmark performance, followed by an ablation study to assess the effectiveness of individual components of our method.

5.1. Datasets and Implementation Details

The evaluations are conducted on four benchmarks, following standard Few-Shot Class-Incremental Learning (FSCIL) protocols: CIFAR100, miniImageNet, CUB200, and IN1K-FSCIL.

CIFAR100, miniImageNet, and CUB200 are widely used benchmarks for FSCIL. CUB200 contains 200 bird species, while miniImageNet and CIFAR100 each comprise 100 common everyday object categories. This section details the experimental results on CUB200; results for the other two datasets are provided in the supplementary material. We adopt the same dataset split for CUB200 as used in existing works [1, 30, 32], denoted as CUB200-5-shot. Specifically, this follows an 11-session split: Session 0 uses 100 base classes (3,000 images), followed by 10 incremental sessions, each introducing 10 novel classes with 5-shot support per class. To further evaluate the effectiveness of prototype calibration under conditions of extreme data scarcity, we also evaluate a 10-way 1-shot configuration for CUB200, where only one sample per novel class is available. This configuration is denoted as CUB200-1-shot.

IN1K-FSCIL. Furthermore, to assess scalability in real-world scenarios with more classes, we construct a challenging ImageNet1K[11] variant by reorganizing 1,000 classes into 800 base classes (200 images per class) and 200 novel classes divided across 10 incremental sessions (20-way 1-shot), which is denoted as **IN1K-FSCIL**. This setting introduces higher inter-class density and diversity compared to existing benchmarks, better simulating practical applications requiring large-scale incremental learning. The results are provided in the supplementary material.

Implementation: For our method, all experiments are implemented in PyTorch with CLIP-ViT/B-16 [17] as the backbone. 512-dimensional visual and textual features are extracted from CLIP's aligned embedding space, with the backbone frozen during both the base and incremental phases, making FDR a **training-free** method. To further bridge the domain gap between CLIP's pretraining data and the target dataset, we apply LoRA fine-tuning on CLIP during the base phase by leveraging an image-text contrastive loss, while refraining from any training in the incremental phase.

Evaluation Metrics: For evaluation, we adopt the commonly used metric of average accuracy (AvgAcc). Specifically, at session t, the average classification accuracy is computed across all classes encountered up to that session (i.e. $M + t * N$). While AvgAcc provides a comprehensive measure of overall performance, it is less sensitive to the accuracy of newly learned novel classes. To address this, we also introduce the average accuracy for novel classes, denoted as NAcc. NAcc is calculated similarly to AvgAcc (i.e., classification over all classes), but the proportion of correctly classified instances is computed exclu-

Table 3. Comparison with existing methods on **CUB200-5-shot** benchmark (Accuracy %). **Bold** indicates best performance. Δ shows improvement of our method over the previous best. Due to space limitations, only the first three and last three sessions are shown.

Method	Backbone	Accuracy in each session (%)												
		0	1		2		3		8		9		10	
		AvgAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc	AvgAcc	NAcc
Baseline[21]	ResNet18	74.14	68.09	12.33	61.86	9.59	57.62	8.56	45.55	8.85	43.33	10.18	41.48	10.06
TOPIC[22]	ResNet18	68.68	62.49	-	54.81	-	49.99	-	32.22	-	28.31	-	26.26	-
CEC[30]	ResNet18	75.85	71.94	-	68.50	-	63.50	-	54.83	-	53.52	-	52.28	-
FACT[32]	ResNet18	75.90	73.23	-	70.84	-	66.13	-	58.41	-	57.89	-	56.94	-
TEEN[1]	ResNet18	77.26	76.13	-	72.81	-	68.16	-	61.19	-	60.32	-	59.31	-
Baseline[21]	CLIP	80.85	79.11	65.02	77.49	65.02	73.98	57.20	69.53	56.11	69.35	58.31	69.20	59.13
TEEN [1]	CLIP	80.85	79.21	67.23	77.54	66.18	73.87	58.21	69.56	56.86	69.45	58.01	69.37	58.83
FDR(Ours)	CLIP	**80.85**	**79.38**	**73.76**	**77.70**	**70.73**	**74.28**	**62.51**	**70.13**	**59.68**	**70.08**	**61.81**	**70.15**	**62.79**
Δ		-	+0.17	+6.53	+0.16	+4.55	+0.30	+4.30	+0.57	+2.82	+0.63	+3.80	+0.78	+3.96

sively within the novel classes.

5.2. Comparison with Existing Methods

We have made comparisons against recent SOTA methods [2–6]. As shown in Table 1, our FDR method surpasses all training-free approaches, consistent with the results of the main paper. For approaches that require training[2–6], we apply the CLIP backbone fine-tuned on base classes. The fine-tuning significantly improves the base session accuracy of our method, making it outperform other methods on CIFAR100, CUB200, and IN1K-FSCIL.

Table 2 and Table 3 present the detailed comparison results on the **CUB200** dataset. All methods utilize the same frozen CLIP backbone (without fine-tuning) or ResNet18 to ensure a fair comparison. The baseline represents vanilla Prototypical Networks [21], which directly uses CLIP features without prototype calibration. The recently proposed method, TEEN, demonstrates significantly better performance than the baseline, benefiting from its comprehensive prototype calibration. Furthermore, our method achieves superior performance across all sessions in both 5-shot and 1-shot settings. Notably, in the 1-shot setting, our method achieves a **6.70%∼19.66%** improvement in **NAcc** across all incremental sessions when comparing with holistic calibration method [1], highlighting its effectiveness in bias correction even under extreme data scarcity. Additionally, the progressive increases in AvgAcc, with improvements ranging from +0.86% to +2.89%, suggest cumulative advantages from our composition strategy as the incremental sessions progress. present the comparison results on the **CUB200** dataset. All methods utilize the same frozen CLIP backbone or ResNet18 to ensure a fair comparison. The baseline represents vanilla Prototypical Networks [21], which directly uses CLIP features without prototype calibration. The recently proposed method, TEEN, demonstrates significantly better performance than the baseline, benefiting from its comprehensive prototype calibration. Furthermore, our method achieves superior performance across all sessions in both 5-shot and 1-shot settings. Notably, in the 1-shot setting, our method achieves a **6.70%∼19.66%** improvement in **NAcc** across all incremental sessions when comparing with holistic calibration method [1], highlighting its effectiveness in bias correction even under extreme data scarcity. Additionally, the progressive increases in AvgAcc, with improvements ranging from +0.86% to +2.89%, suggest cumulative advantages from our composition strategy as the incremental sessions progress.

Table 4. Ablation study on CUB200-1-shot (Session 10).

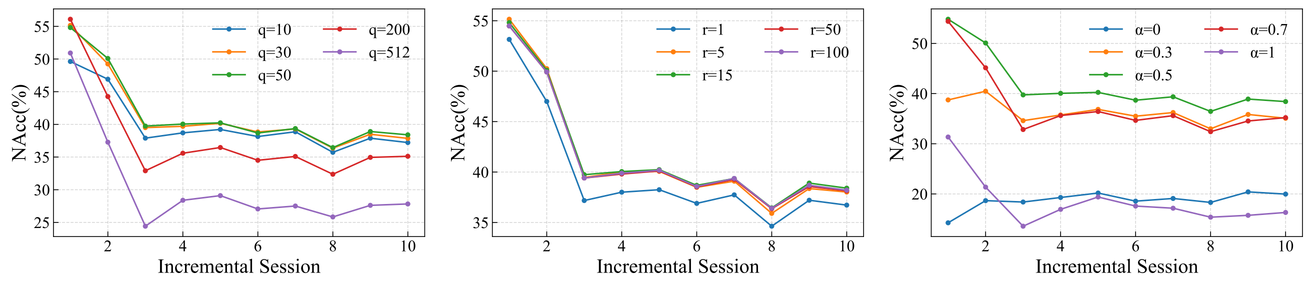
Method	AvgAcc	NAcc
No Calibration	49.75	19.86
Holistic Calibration only	54.48	31.56
Random Feature Segments	54.83	32.08
Naive Keyword Grounding	56.61	36.25
Full (Ours)	**57.37**	**38.40**

5.3. Ablation Studies

This section presents ablation experiments on the CUB200-1-shot dataset to quantify the contribution of each component and assess the impact of hyperparameters. The performance is evaluated using the final session's metrics, including the average accuracy (AvgAcc) for all classes and the novel class accuracy (NAcc) for all novel classes.

Effectiveness of Component. We dissect the key components of our method through several controlled experiments:

- *No Calibration*: This strategy directly aggregates features of each class provided by the CLIP backbone to form prototypes. For novel classes, the prototype is calculated directly from the few-shot samples without any calibration. As a result, this approach serves as a vanilla baseline.
- *Holistic Calibration*: This method corresponds to the TEEN approach [1], which employs holistic calibration for prototypes. It is used as a comparative baseline to dissect and evaluate the effectiveness of the fine-grained calibration proposed in our method.



(a) NAcc w.r.t. Top-q (b) NAcc w.r.t. Top-r (c) NAcc w.r.t. α

Figure 3. Impact of hyperparameters on average accuracy over all novel classes at last session (i.e. NAcc), where q stands for the size of feature segments, r for the number of base-class feature segments for calibrating, and α for the calibration strength of each feature segment.

- *Random Feature Segments*: This strategy represents a degraded version of our method, where the feature segments are replaced by random divisions of visual features without any semantic guidance. This approach is used to dissect the necessity of the keyword grounding function.
- *Naive Keyword Grounding*: This strategy also represents a degraded version of our method, where the mapping of each keyword to the feature segment is simplified by multiplying the [EOS] token embedding with the keyword token embedding, rather than incorporating the attention matrix $\mathbf{A}$ and value matrix $\mathbf{V}$. This approach is used to dissect and demonstrate that whether the design of the keyword grounding function is trivial.

The evaluation results are shown in Table 4. As can be observed, the 11.70% NAcc gain of ”Holistic Calibration only ” over ”No Calibration” confirms that prototype calibration is beneficial. Furthermore, our method outperforms ”Holistic Calibration only ”, demonstrating that fine-grained calibration of few-shot class prototypes is both effective and necessary. The inferior performance of ”Random Feature Segments ” and ”Naive Keyword Grounding” compared to our full method indicates that the effectiveness of fine-grained calibration for few-shot class prototypes relies heavily on its elaborate design. This underscores the value of our carefully designed keyword grounding function, highlighting its critical role in achieving superior performance.

Impact of Hyperparameter. We analyze the impact of three key parameters in controlled experiments:

- *Top-q binarization* are conducted on contribution indication matrix $\mathbf{B}$ along the feature dimensions, where q controls the size of feature segments. From Fig. 3a, we observe that feature segments obtained the best performance with $q = 50$ in NAcc. Smaller values of q (e.g., 10 and 30) result in smaller feature segments, which may be insufficiently captured by the corresponding keywords, leading to suboptimal performance. On the other hand, larger values of q (e.g., 200, 512) degrade performance, as they tend to approach holistic calibration and fail to yield improvements. This indicates that an intermediate granularity in feature segmentation, which well corresponds to the keyword, is essential.
- *Top-r similar prototype segments* from base classes are combined to calibrate the novel class, where r controls the number of base-class feature segments for calibrating. In Fig. 3b, we see that increasing the number of similar prototypes (i.e. a larger r) enhances performance up to r=50, after which further increases do not yield substantial benefits. This suggests that a small set of highly relevant prototypes is sufficient for effectively calibrating novel class prototypes. Including too many segments may introduce noise or less relevant information, diminishing the calibration quality.
- *Calibration ratio* α controls the calibration strength of each feature segment. As shown in Fig. 3c, the calibration ratio $\alpha = 0.5$ yields the best results, balancing the original and calibrated features effectively. Lower values of α reduce the impact of calibration, while higher values overly prioritize the reference prototypes, leading to suboptimal results.

6. Conclusion and Future Work

In this work, we propose a semantic Feature Decomposition-Recomposition method that unlocks the full potential of CLIP's visual-semantic features for few-shot class-incremental learning. Our core innovation lies in decomposing CLIP's frozen features into semantic-aware segments through guidance of textual keywords, making these segments transfer-ready through incremental learning to be selectively recomposed for novel class calibration. By establishing explicit mappings between LLM-generated attribute keywords and their corresponding feature segments, our method enables precise inheritance of visual-semantic characteristics from multiple base classes. The improvement of our method over existing methods on several settings demonstrates its unique capability to mitigate prototype bias while preserving discriminative features.

7. Acknowledgment

This work is partially supported by the National Natural Science Foundation of China (No.62495082 and No.62461160331).

References

[1] Few-shot class-incremental learning via training-free prototype calibration. In *NIPS*, 2023. 1, 2, 3, 6, 7

[2] Closer: Towards better representation learning for few-shot class-incremental learning. In *ECCV*, 2024. 6, 7

[3] Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *CVPR*, 2024. 6

[4] Learning prompt with distribution-based feature replay for few-shot class-incremental learning. *arXiv preprint*, 2024. 6

[5] Pre-trained vision and language transformers are few-shot incremental learners. In *CVPR*, 2024. 6

[6] Rethinking few-shot class-incremental learning: Learning from yourself. In *ECCV*, 2024. 6, 7

[7] Afra Feyza Akyürek, Ekin Akyürek, Derry Tanti Wijaya, and Jacob Andreas. Subspace regularizers for few-shot class incremental learning. *arXiv preprint arXiv:2110.07059*, 2021. 1, 2, 3

[8] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. On tiny episodic memories in continual learning. *arXiv preprint arXiv:1902.10486*, 2019. 2

[9] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *International conference on learning representations*, 2021. 1

[10] Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *CVPR*, pages 2534–2543, 2021. 1, 2, 3

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[12] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 2

[13] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[14] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, pages 1–30. Lille, 2015. 2

[15] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *ICCV*, 2021. 1

[16] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*. Elsevier, 1989. 1, 2

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 2021. 6

[18] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *NIPS*, 2019. 2

[19] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *NIPS*, 34:6747–6761, 2021. 1

[20] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *NIPS*, 30, 2017. 2

[21] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *NeurIPS*, 30, 2017. 1, 2, 3, 6, 7

[22] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *CVPR*, 2020. 1, 2, 3, 7

[23] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. A survey on few-shot class-incremental learning. *Neural Networks*, 169, 2024. 1

[24] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *NIPS*, 29, 2016. 2

[25] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE TPAMI*, 2024. 2

[26] 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 (csur)*, 53(3), 2020. 1, 2

[27] Chenshen Wu, Luis Herranz, Xialei Liu, Joost Van De Weijer, Bogdan Raducanu, et al. Memory replay gans: Learning to generate new categories without forgetting. *NIPS*, 31, 2018. 2

[28] Boyu Yang, Mingbao Lin, Binghao Liu, Mengying Fu, Chang Liu, Rongrong Ji, and Qixiang Ye. Learnable expansion-and-compression network for few-shot class-incremental learning. *arXiv preprint arXiv:2104.02281*, 2021. 1

[29] Boyu Yang, Mingbao Lin, Yunxiao Zhang, Binghao Liu, Xiaodan Liang, Rongrong Ji, and Qixiang Ye. Dynamic support network for few-shot class incremental learning. *IEEE TPAMI*, 45(3):2945–2951, 2022. 1

[30] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *CVPR*, pages 12455–12464, 2021. 1, 6, 7

[31] Guangtao Zheng and Aidong Zhang. Few-shot class-incremental learning with meta-learned class structures. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 421–430. IEEE, 2021. 1

[32] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-

shot class-incremental learning. In *CVPR*, pages 9046–9056, 2022. 1, 6, 7

[33] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *CVPR*, pages 6801–6810, 2021. 1