Compositional Few-Shot Class-Incremental Learning

Yixiong Zou [1] **Shanghang Zhang** [2] **Haichen Zhou** [1] **Yuhua Li** [1] **Ruixuan Li** [1]

Abstract

Few-shot class-incremental learning (FSCIL) is proposed to continually learn from novel classes with only a few samples after the (pre-)training on base classes with sufficient data. However, this remains a challenge. In contrast, humans can easily recognize novel classes with a few samples. Cognitive science demonstrates that an important component of such human capability is compositional learning. This involves identifying visual primitives from learned knowledge and then composing new concepts using these transferred primitives, making incremental learning both effective and interpretable. To imitate human compositional learning, we propose a cognitive-inspired method for the FSCIL task. We define and build a compositional model based on set similarities, and then equip it with a primitive composition module and a primitive reuse module. In the primitive composition module, we propose to utilize the Centered Kernel Alignment (CKA) similarity to approximate the similarity between primitive sets, allowing the training and evaluation based on primitive compositions. In the primitive reuse module, we enhance primitive reusability by classifying inputs based on primitives replaced with the closest primitives from other classes. Experiments on three datasets validate our method, showing it outperforms current state-of-the-art methods with improved interpretability. Our code is available at https://github.com/Zoilsen/Comp-FSCIL.

1. Introduction

With advancements in hardware, deep neural networks have demonstrated considerable success across various areas using pre-defined large-scale datasets (Simonyan & Zisserman, 2015; He et al., 2016). However, real-world scenarios present novel knowledge continuously, often with limited data (Hou et al., 2019; Rebuffi et al., 2017), such as rare diseases. Addressing this challenge requires models to learn novel knowledge from just a few samples without forgetting previously learned ones (Castro et al., 2018; Tao et al., 2020). This necessity gives rise to the Few-Shot Class-Incremental Learning (FSCIL) task (Zhang et al., 2021; Zhou et al., 2022). In this task, models are initially (pre-)trained on base classes during a base session with sufficient training data. Then, models learn from novel classes in incremental sessions with only a few samples, and finally classify test samples across all encountered classes. While various approaches, including metric-based ones (Zou et al., 2022; Zhang et al., 2021), adaptation-based ones (Zhou et al., 2022; Yibo Yang, 2023), etc. have been explored for this task, FSCIL remains a challenge due to the scarcity of training data and the risk of catastrophic forgetting.

In contrast to machines, humans can easily learn from lim-

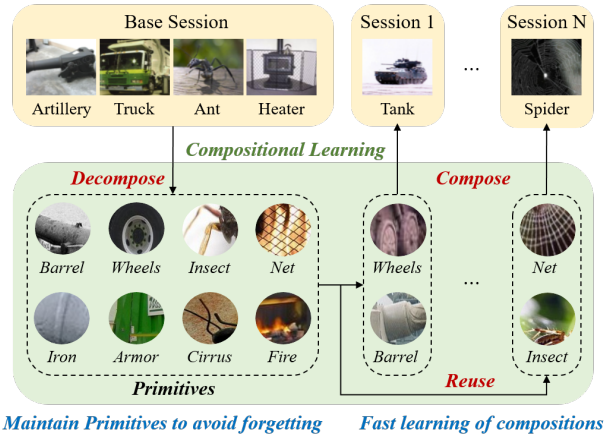


Figure 1. As studied by cognitive science (Biederman, 1987), humans can compositionally learn knowledge by dividing learned ones into primitives, and then compose them to learn novel knowledge, which leads to the good human capability of incremental learning with only scarce data. To imitate the human ability of compositional learning, we propose a compositional learning method for the few-shot class-incremental learning (FSCIL) task. We briefly plot the primitives automatically found by our methods with the possible meanings, where we can see good reusability and interpretability of primitives. Detailed plots are in Fig.7.

[1]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, China [2]School of Computer Science, Peking University, Beijing, China. Correspondence to: Ruixuan Li <rxli@hust.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

ited data without forgetting learned knowledge (Schwartz et al., 2019). Cognitive science demonstrates that an important component of such human capability is compositional learning (Biederman, 1987; Zou et al., 2020), which enables humans to divide knowledge, such as semantic objects, into visual primitives (Hoffman & Richards, 1984) (like object parts), and then compose novel or learned knowledge by transferred primitives (Fodor, 1975), as shown in Fig.1. Since primitives are reusable among base and novel classes, it enables us to not only avoid forgetting by maintaining learned primitives, but also efficiently learn from few-shot novel classes by learning the composition of primitives. Moreover, primitives can be viewed as the foundational elements guiding human decision-making, providing insights into why a particular sample is classified into a specific class. This enhances the interpretability of black-box deep learning models (e.g., Fig.1 depicts a spider as a composition of an ant-like insect and nets). Consequently, this paper aims to imitate the human ability for compositional learning to tackle the challenging problem of FSCIL.

Specifically, we first define primitive composition based on set similarities and then build our model by modifying the FSCIL base method. Since primitives always refer to object parts (Zou et al., 2020), we define image patches as candidate primitives, which may contain sample-specific candidate primitives such as background, and common primitives shared across samples. For each class, we employ a set of prototypes to form its primitive set, which encodes common primitives shared within this class. We then propose to utilize the Centered Kernel Alignment (CKA) similarity (Kornblith et al., 2019) to approximate the similarity between primitive sets, which enables the training and evaluation based on primitive compositions. To enhance the reusability of primitives across classes, we further design a primitive reuse module, which classifies input samples based on primitives replaced with the closest primitives from other classes. Our model is firstly trained in the base session, and then transferred to incremental sessions with a fixed backbone network. The reusability of primitives is achieved both implicitly through the reuse of the backbone network and explicitly through the primitive reuse module.

In summary, our contributions can be listed as:

• We propose a cognitive-inspired compositional learning method for the FSCIL task, which first defines and builds a compositional model based on set similarities, and then equips it with a primitive composition module and a primitive reuse module.

• In the primitive composition module, we propose to utilize the CKA similarity to approximate the similarity between primitive sets, allowing the training and evaluation based on primitive compositions.

• In the primitive reuse module, we enhance primitive reusability by classifying inputs based on primitives that are replaced with the nearest primitives from other classes.

• Extensive experiments on three public benchmarks validate the rationale of our compositional learning method, and demonstrate it outperforms current state-of-the-art works while providing enhanced interpretability.

2. Related Work

Few-shot class-incremental learning can be roughly categorized into adaptation-based (Hou et al., 2019; Rebuffi et al., 2017; Castro et al., 2018; Tao et al., 2020) and metric-based methods (Zhang et al., 2021; Zou et al., 2022). The first group adapts the model during novel-class training, with the backbone network often frozen to prevent catastrophic forgetting (Zhou et al., 2022). In the second group, each class is represented by prototypes averaged from samples (Zou et al., 2022), with network parameters similarly frozen to mitigate the risk of catastrophic forgetting. However, there is a scarcity of works exploring the compositional structure of FSCIL models, and as far as we know, our study is the first to delve into this aspect.

Compositional learning seeks to learn knowledge through its primitives or components, which is a concept extensively explored in cognitive science (Biederman, 1987; Hoffman & Richards, 1984; Fodor, 1975). This approach has found applications in various domains, such as CPDE (Zou et al., 2020) decomposing classes into channels for few-shot learning, (Purushwalkam et al., 2019) breaking down visual features into attributes for zero-shot learning, (Kato et al., 2018) decomposing human-object interactions into actions and objects, (Cao et al., 2021) learning a dictionary for visual concepts, and (Tang et al., 2020) aligning object parts with pose normalization. However, there has been limited exploration in the FSCIL task. In contrast, our decomposition operates within the spatial dimension and does not need additional annotations for primitives. Due to space limitations, we present further details on related works in the appendix.

3. Method

We first define primitive composition by set similarities and then design each component to implement compositional few-shot class-incremental learning (FSCIL) (Fig.2).

3.1. Preliminaries

FSCIL (Zhang et al., 2021; Zhou et al., 2022) aims to continually learn from novel classes with only a few samples in incremental sessions, after (pre-)training on base classes with abundant training data in the base session. Initially, the model is trained on the base session dataset

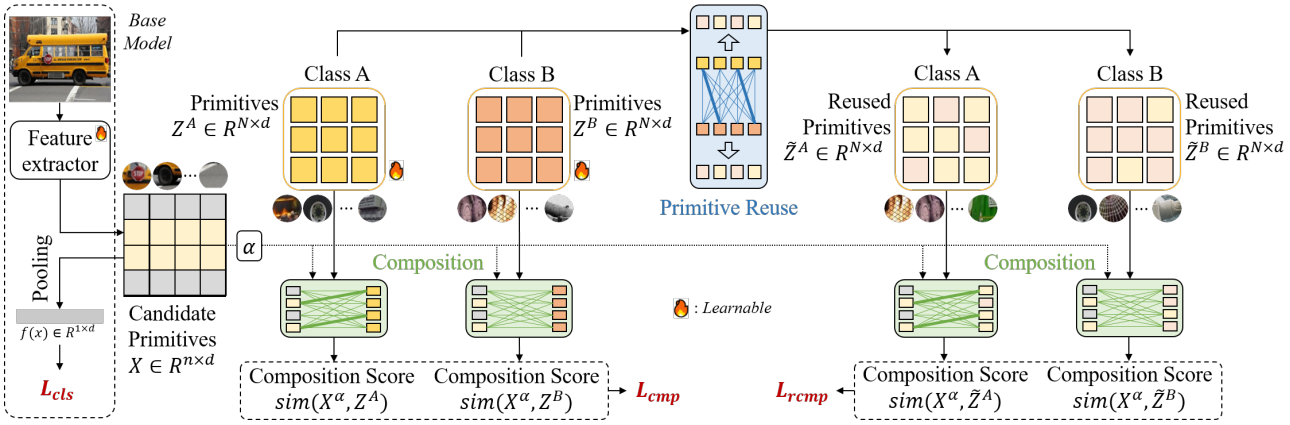


Figure 2. We take image patches as candidate primitives, and utilize a set of prototypes to construct the primitive set for each class. Given an input sample, our method tries to compose it with primitive sets (e.g., Z^A and Z^B) from different classes (e.g., class A and B), which is measured as the composition score by the CKA similarity. These composition scores are then utilized to be the classification score for the model training and evaluation. To improve the reusability of primitives across classes, each primitive is replaced with the closest primitive in other classes. The replaced primitive sets (e.g., $\tilde{Z}^A$ and $\tilde{Z}^B$) will then be applied in the classification with the composition score. Finally, our model is trained with the combination of L_{cls}, L_{cmp}, and L_{rcmp} during both the base and incremental sessions.

$D^0 = \{(x_i, y_i)\}_{i=1}^{n_0}$ with the label space Y_0, by minimizing the loss $\sum_{(x_i,y_i)\in D^0} L(\phi(W, x_i), y_i)$, where $L(\cdot,\cdot)$ denotes the cross-entropy loss and $\phi(\cdot,\cdot)$ gets the prediction of x_i. Typically, $\phi(\cdot,\cdot)$ consists of a feature extractor $f(\cdot)$ and a classifier (e.g., fully connected (FC) layer) with the parameter W, where $\phi(W, x) \in R^{1\times|Y_0|}$, $W \in R^{|Y_0|\times d}$ and $f(x) \in R^{1\times d}$. During the kth incremental session, the model learns from the dataset $D^k = \{(x_i, y_i)\}_{i=1}^{n_k}$ with the label space Y_k. The classifier's weight W will be extended by incorporating the classifier obtained from D^k as $W = \{w_1^0, w_2^0, ..., w_{|Y_0|}^0\} \cup ... \cup \{w_1^k, ..., w_{|Y_k|}^k\} \in R^{\sum_{i=0}^k |Y_i|\times d}$ where $w_j^k \in R^{1\times d}$ denotes the classifier weight for the jth class in the kth session. A prevailing baseline method (Zhang et al., 2021) freezes $f(\cdot)$ during incremental sessions and only trains the classifier. Finally, the model will be applied to classify test samples from all encountered $\sum_{i=0}^k |Y_i|$ classes.

3.2. Defining the Compositional Recognition

Humans' compositional learning first divides knowledge into primitives (Hoffman & Richards, 1984; Zou et al., 2020), and then composes novel knowledge using these primitives (Fodor, 1975). This learning mechanism avoids forgetting by maintaining learned primitives, and facilitates few-shot learning by efficiently learning the composition of reused primitives. To imitate this human ability, we begin with the following definition.

Definition 3.1. Given a class y, compositional learning divides it into a set of primitives $\{P_i^y\}_i^N$, representing shared components in this class. Each sample x is divided into a set of components $\{C_i(x)\}_i^n$, where components shared with other samples in this class construct $\{P_i^y\}_i^N$, while other components are specific to this sample.

Therefore, we refer to $C_i(x)_i^n$ as the candidate primitive set. We use the notations Z^y and X as abbreviations for $\{P_i^y\}_i^N$ and $C_i(x)_i^n$, respectively. Note that primitives should be transferable or even reusable across classes. Considering that each sample and class are represented by sets, if all elements in set X are present in set Z^y, we can assert that X is composed of elements from Z^y. Therefore, we define composition using the Jaccard Similarity as:

Definition 3.2. Set $X = \{C_i(x)\}_i^n$ is composed of elements from set $Z^y = \{P_i^y\}_i^N$ if $\frac{|Z^y \cap X|}{|Z^y \cup X|} = 1.0$.

However, this criterion is strict and hard to apply due to two reasons: (1) the term $|Z^y \cap X|$ is not continuous, and (2) achieving $\frac{|Z^y \cap X|}{|Z^y \cup X|} = 1.0$ is difficult because X may include elements specific to the sample (e.g., background).

Therefore, we relax this criterion to $\frac{|Z^y \cap X|}{|Z^y \cup X|} > t$. Considering the sparsity of primitives (Zou et al., 2020), we assume $|Z^y| << |X|$ and simplify the Jaccard Similarity as

$$\frac{|Z^y \cap X|}{|Z^y \cup X|} = \frac{|Z^y \cap X|}{|X| + |Z^y| - |Z^y \cap X|} \approx \frac{|Z^y \cap X|}{|X|}, \quad (1)$$

since $|X| >> |Z^y| > |Z^y| - |Z^y \cap X|$. Moreover, we relax the concrete union of sets to the similarity between sets as

$$|Z^y \cap X| \approx sim(X, Z^y) = \sum_i^{|X|} \sum_j^{|Z^y|} s(X_i, Z_j^y), \quad (2)$$

where s denotes the similarity between primitives. Ideally, $s(\cdot,\cdot)$ outputs 1 if X_i could totally match Z_i^y, and 0 if X_j

and Z_i^y are not matched (e.g., cosine similarity). Since $sim(X, Z^y)$ is continuous, we can utilize it to represent the probability of classifying the sample x into the class y as

$$P(y|x) = \frac{e^{\tau \cdot sim(X, Z^y)/|X|}}{\sum_k e^{\tau \cdot sim(X, Z^k)/|X|}}. \quad (3)$$

where τ is a temperature parameter. Naturally, $P(y|x)$ can be used in training and evaluation. Therefore, there remain three issues to implement the compositional learning: (1) designing the primitives; (2) designing the set similarity function $sim(\cdot, \cdot)$ and (3) designing the reuse of primitives.

3.3. (Candidate) Primitive Design

To achieve this goal, we look back into the FSCIL base method (Fig.2). Given an input image x, this model extracts its feature as $f(x) \in R^{1\times d}$, and then forwards it to the classifier with the parameter $W \in R^{|Y_0|\times d}$, where $W_y \in R^{1\times d}$ is viewed as the prototype of the yth class. Typically, these features have been processed by the Global-Average-Pooling layer, such as ResNet (He et al., 2016) and Swin Transformer (Liu et al., 2021). Therefore, we have $f(x) = \frac{1}{S}\sum_i^S F(x)_i$, where $F(x) \in R^{S\times d}$ denotes the feature map and S is the spatial dimension of the map. Given that the object in the input image is partitioned into image patches, the patch features $F(x)$ can be seen as compositionally representing the input x. Therefore, it can be regarded as the candidate primitive set containing object parts, i.e.,

$$X = \{C_i(x)\}_i^n = \{F(x)_i\}_i^S. \quad (4)$$

Furthermore, this design satisfies the transferability requirements of primitives, as patch features are more readily transferable compared with image features. With this design, the term $|X|$ in Eq.3 is a constant number S. Therefore, the designing of the compositional model is simplified into finding a suitable similarity function between X and Z^y.

On the other hand, since W_y inherently learns to represent the centroid of class y during the training of the base model, it captures the shared patterns of the class y while disregarding the sample-specific patterns like the background. Similarly, in line with Eq.4, we replace W_y of class y with a collection of prototypes to be the primitive set $\{P_i^y\}_i^N$ where $P_i^y \in R^{1\times d}$. With this choice of primitive set, similarly, $\{P_i^y\}_i^N$ would also learn the common image patches (i.e., candidate primitives) that are shared among different samples from the class y, which serves as the centroid of candidate primitives and ignores sample-specific candidate primitives such as background. Therefore, this choice of primitive set satisfies the definition 3.1.

Consequently, based on the above designs, we can directly modify the architecture of the baseline network to implement our compositional model, by replacing $f(x) \in R^{1\times d}$ with $X \in R^{n\times d}$ where $n = S$, and replacing $W \in R^{|Y_0|\times d}$ with $Z \in R^{|Y_0|\times N\times d}$.

3.4. Set Similarity Function Design

Next, we need to design the similarity function $sim(\cdot, \cdot)$, with a crucial issue to find matches between two sets. A straightforward way is to enumerate all matches as:

$$\frac{1}{nN}\sum_{i,k}^{n,N} \frac{X_i}{||X_i||}\frac{Z_k^y}{||Z_k^y||} = (\frac{1}{n}\sum_i^n \frac{X_i}{||X_i||})(\frac{1}{N}\sum_k^N \frac{Z_k^y}{||Z_k^y||}), \quad (5)$$

where $Z^y \in R^{N\times d}$ and the cosine similarity is used as $s(\cdot, \cdot)$ to measure the similarity between primitives.

However, Eq.5 indicates that this strategy degenerates the separated primitives into the averaged feature $\frac{1}{n}\sum_i^n \frac{X_i}{||X_i||}$ which closely resembles $f(x) = \frac{1}{S}\sum_i^S F(x)_i$. This approach lacks the flexibility to capture the compositional information inherent in each class and sample. Furthermore, it may incorporate sample-specific candidate primitives, such as the background, into the primitive set, because the matching score between these sample-specific candidate primitives and Z^y could be mistakenly high.

Therefore, we set a weight for each matching as

$$sim(X, Z^y) = \frac{1}{nN}\sum_{i,k} w_{ik}^A \frac{X_i}{||X_i||}\frac{Z_k^y}{||Z_k^y||} \quad (6)$$

where $W^A = \{w_{ik}^A\}_{ik}^{N\cdot n} \in R^{n\times N}$ is a weight matrix, which filters out sample-specific candidate primitives in X by assigning low weights to their similarity with the primitive set, and highlights important ones.

Inspiration from Representation Comparison

To obtain W^A, we draw inspiration from a related field: comparing representations across different models (Kornblith et al., 2019). This area focuses on comparing outputs from various neural networks to study the behavior of deep models, typically based on features extracted from the same set of images (Kornblith et al., 2019). However, as representations extracted by different models may not share the same set of channels, simple calculations like Euclidean or cosine similarity may not be effective. Therefore, various approaches have been proposed to handle unmatched channels, including linear regression(Romero et al., 2015), CCA(Raghu et al., 2017), SVCCA(Raghu et al., 2017), DeepEMD(Oh et al., 2022), etc. Among these, CKA(Kornblith et al., 2019) shows better reliability and lower computational cost(Davari et al., 2022).

To compare representations of different models (RC), **the same batch of inputs** is employed, but **channels are disordered** and challenging to compare directly. In the comparison between X and Z^y, (candidate) primitives are described by **the same set of channels**, but **primitives are disordered** and hard to be compared directly. Such symmetry inspires us to *view the channel dimension in RC as the primitive dimension in* $sim(X, Z^y)$ *and the batch dimension in RC*

as the channel dimension in $sim(X, Z^y)$. Therefore, we propose to use CKA as a better similarity function to obtain W^A in Eq.6 by simply transposing X and Z^y. In summary, CKA captures the correlation between channels given an ordered batch, and we propose to use it to capture the correlation between primitives given ordered channels.

Specifically, in RC, given two models $h(\cdot)$ and $g(\cdot)$ for comparison, features are extracted from X^r with the batch size b^r as $h(X^r) \in R^{b^r \times d_h}$ and $g(X^r) \in R^{b^r \times d_g}$ to obtain the CKA similarity as

$$\text{CKA} = \frac{\text{HSIC}(h(X^r), g(X^r))}{\sqrt{\text{HSIC}(h(X^r), h(X^r)) \cdot \text{HSIC}(g(X^r), g(X^r))}},$$
$$\text{HSIC}(K, L) = \frac{1}{(b^r - 1)^2} tr(KHLH) \tag{7}$$

where K and L denotes $h(X^r)$ and $g(X^r)$, H is the centering matrix $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$ (Kornblith et al., 2019).

For our $sim(X, Z^y)$, we follow Eq.7 to replace $h(X^r)$ and $g(X^r)$ with $X^\top$ and ${Z^y}^\top$ respectively. Then, the similarity between X and Z^y can be obtained as

$$sim(X, Z) = \frac{\text{HSIC}(X^\top, {Z^y}^\top)}{\sqrt{\text{HSIC}(X^\top, {Z^y}^\top) \cdot \text{HSIC}(X^\top, {Z^y}^\top)}} \tag{8}$$
$$\overset{\text{linear}}{=} \frac{||\tilde{X}{\tilde{Z}^y}^\top||^2}{||\tilde{X}\tilde{X}^\top|| \cdot ||\tilde{Z}{\tilde{Z}^y}^\top||}. \tag{9}$$

Recent studies have demonstrated that the linear CKA similarity exhibits reliability comparable to the kernel CKA similarity. Hereafter, we employ the linear CKA similarity as our CKA-based similarity function to approximate $|Z^y \cap X|$ in Eq.2. In the linear CKA, Eq.9 is expressed equivalently with the centered features $\tilde{X} = X - \frac{1}{d}\sum_j X_{:,j}$ and $\tilde{Z}^y = Z^y - \frac{1}{d}\sum_j Z^y_{:,j}$, where the dot product is computed in the channel dimension instead of the primitive dimension.

Benefiting Compositions with Less Computational Overhead and Robustness to Primitive Noises

By expanding Eq.9, we have

$$sim(X, Z^y) = \sum_{i,k}^{n,N} \frac{(\tilde{X}_i {\tilde{Z}^y_k}^\top)}{||\tilde{X}\tilde{X}^\top|| \cdot ||\tilde{Z}^y{\tilde{Z}^y}^\top||} \cdot (\tilde{X}_i {\tilde{Z}^y_k}^\top) \tag{10}$$
$$= \sum_{i,k}^{n,N} \underbrace{\frac{(\frac{\tilde{X}_i}{||\tilde{X}_i||}\frac{\tilde{Z}^y_k}{||\tilde{Z}^y_k||}^\top)}{||\frac{\tilde{X}}{||\tilde{X}||}\frac{\tilde{X}}{||\tilde{X}||}^\top|| \cdot ||\frac{\tilde{Z}^y}{||\tilde{Z}^y||}\frac{\tilde{Z}^y}{||\tilde{Z}^y||}^\top||}}_{(w^A_{ik})} \cdot \underbrace{(\frac{\tilde{X}_i}{||\tilde{X}_i||}\frac{\tilde{Z}^y_k}{||\tilde{Z}^y_k||}^\top)}_{(\frac{X_i}{||X_i||}\frac{Z^y_k}{||Z^y_k||}^\top)} \tag{11}$$

where $||\tilde{X}||$ and $||\tilde{Z}||$ denotes the row-wise norm.

Compare Eq.6 and Eq.11, we can see Eq.11 well matches Eq.6 by automatically generating the W^A_{ik} and replacing X and Z with $\tilde{X}$ and $\tilde{Z}$ respectively. Since W^A_{ik} is obtained through the matrix multiplication, no extra computations are needed, **reducing the computational overhead** compared with DeepEMD or FRN(Wertheimer et al., 2021).

Table 1. Evaluation of models trained by the baseline method. CKA and the power transformed CKA shows less computational overhead and better robustness to primitive noises.

	FRN	DeepEMD	CKA	Power transformed CKA
Last session accuracy (%) ↑	18.15	24.03	38.42	**39.47**
Time (sec. / 100 images) ↓	0.0233	12.3166	**0.0139**	0.0161

Moreover, since DeepEMD or FRN obtain the matching weight by globally taking all patches into account, it makes them vulnerable to primitive noises. Such noise exists for two reasons: (1) each sample contains sample-specific candidate primitives, but the features for them are not well trained, due to their marginal contribution to classification; (2) features are not discriminative enough at the early of training. Such noises would make the complex linear programming in DeepEMD or FRN fragile, harming the set comparison. In contrast, in Eq.11, each matching weight is generated by mainly taking the local comparison of two patches ($\tilde{X}_i$ and $\tilde{Z}^y_k$). Such simplicity makes the comparison less vulnerable to noisy patch features and more robust.

To verify the computational efficiency and the noise robustness, we train a model with only the baseline classification loss on CIFAR100, and then conduct an evaluation based on CKA, DeepEMD, and FRN. The average time and accuracy of the last session are reported in Tab.1. Since the model is not trained with the corresponding distance metric, the feature extracted by it could be understood to be ineffective and thus noisy. We can see CKA shows the highest performance under such noisy features with the lowest time cost.

As CKA is robust to patch noises, we further propose to apply a power transformation on it to enhance such robustness. Specifically, we introduce the transformation on feature maps by replacing X_i with X_i^α element-wisely, where $\alpha < 1.0$. This action smooths the distribution of matching weights to avoid outliers caused by primitive noises.

Based on Eq.3 and Eq.11, we can classify x by trying to compose it with primitive sets from different classes, which also brings the training loss as

$$L_{cmp} = -ln\frac{e^{\frac{sim(X^\alpha, Z^y)}{|X|}}}{\sum_k e^{\frac{sim(X^\alpha, Z^k)}{|X|}}} = -ln\frac{e^{\tau \cdot sim(X^\alpha, Z^y)}}{\sum_k e^{\tau \cdot sim(X^\alpha, Z^k)}} \tag{12}$$

where τ is a temperature parameter to absorb $|X|$ since $|X|$ is a constant. Since we have $|Z^y \cap X| \approx sim(X, Z^y)$ in Eq.2, we call $sim(X^\alpha, Z^y)$ the composition score.

3.5. Primitive Reuse Design

Primitives are shared among classes, enhancing the interpretability of the compositional model. However, in the

Table 2. Comparison on the *mini*ImageNet dataset. PD: lower performance drop indicates less forgetting.

Backbone	Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	PD ↓
ResNet18	DeepEMD(Zhang et al., 2020)	69.77	64.59	60.21	56.63	53.16	50.13	47.79	45.42	43.41	26.36
	CLOM(Zou et al., 2022)	73.08	68.09	64.16	60.41	57.41	54.29	51.54	49.37	48.00	25.08
	SoftNet(Yoon, 2023)	76.63	70.13	65.92	62.52	59.49	56.56	53.71	51.72	50.48	26.15
	ALICE(Can Peng, 2022)	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	24.90
	SAVC(Song et al., 2023)	81.12	76.14	72.43	68.92	66.48	62.95	59.92	58.39	57.11	24.01
	Comp-FSCIL	**82.78**	**77.82**	**73.70**	**70.57**	**68.26**	**65.11**	**62.19**	**60.12**	**59.00**	**23.78**
ResNet12	NC-FSCIL(Yibo Yang, 2023)	**84.02**	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	25.71
	Comp-FSCIL	84.00	**78.49**	**74.44**	**71.51**	**69.30**	**66.61**	**63.66**	**61.64**	**60.61**	**23.39**

Table 3. Dataset (Zhang et al., 2021). Every dataset provides fixed training and test sets, so the sampling of episodes is not needed.

Dataset	Total	Base	Novel	Inc. Sessions	Shot	Image Size
*mini*ImageNet	100	60	40	8	5	84 × 84
CUB200	200	100	100	10	5	224 × 224
CIFAR100	100	60	40	8	5	32 × 32

current design, primitive sets are learned independently in each class, restricting the reuse of primitives. To address this limitation, we introduce a primitive-reuse module to enhance the correlation between primitives across classes.

During the base-session training, given the primitive set $Z^y = \{P_i^y\}_i^N$ from the class y, we use primitives from other base classes to replace Z^y. Denote primitives from other base classes as $\{P_k^o\}_k^{N\cdot(|Y_0|-1)}$, for each P_i^y, we obtain its similarity with other primitives as $s_r(P_i^y, P_k^o) = -||P_i^y - P_k^o||^2$. Then we calculate the attention on P_k^o against all other base-class primitives as

$$att_k^{y,i} = \frac{e^{\gamma \cdot s_r(P_i^y, P_k^o)}}{\sum_{k=1}^{N\cdot(|Y_0|-1)} e^{\gamma \cdot s_r(P_i^y, P_k^o)}}, \tag{13}$$

where γ is a pre-defined hyper-parameter. The replacement is then calculated as a weighted sum of all primitives as

$$\hat{P_i^y} = \sum_{k=1}^{N\cdot(|Y_0|-1)} att_k^{y,i} P_k^o. \tag{14}$$

By setting a large γ (e.g., 64), we push the model to focus on only the closest primitive from other classes. Then, we use $\hat{P_i^y}$ to replace P_i^y. The above replacement will be carried out on all primitives $Z \in R^{|Y_0|\times N\times d}$ to obtain the replaced primitive sets $\hat{Z}$. Finally, a classification loss will be applied to the input sample based on the replaced primitive set $\hat{Z}$ as

$$L_{rcmp} = -ln \frac{e^{\tau \cdot sim(X^\alpha, \hat{Z^y})}}{\sum_i e^{\tau \cdot sim(X^\alpha, \hat{Z^i})}}, \tag{15}$$

where τ is a temperature parameter. During training, as X can be effectively classified by the original primitive sets, minimizing L_{rcmp} pushes the model to generate optimal replacements for each primitive P_i^y by reducing its distance with the nearest primitives from different classes, facilitating the reuse of primitives across classes.

In the incremental session, primitives from all base classes are employed to replace novel-class primitive sets.

3.6. Model Training and Evaluation

During the base session, we incorporate the baseline classification loss to ensure the stability of model training. The ultimate model encompasses two classifiers: one for the ordinary classification (with parameters $W \in R^{|Y|\times d}$) and another for the compositional classification (with parameter $Z \in R^{|Y|\times N\times d}$). In the baseline classification loss, we utilize the standard feature (i.e., the global-average-pooling feature or the CLS token feature, $f(x)$) to compute the loss

$$L_{cls} = -ln \frac{e^{\tau \cdot s(f(x), W_y)}}{\sum_i e^{\tau \cdot s(f(x), W_i)}}. \tag{16}$$

In all, the model is trained with all three losses as

$$L = L_{cls} + \lambda_1 L_{cmp} + \lambda_2 L_{rcmp}. \tag{17}$$

In the incremental session, we fix the backbone network and only train the novel-class primitive set Z^{novel} by Eq.17. The base-class primitives are reused in novel classes, both implicitly by the transferring of the backbone network and explicitly by L_{rcmp}. During this period, the model learns the composition of reused primitives by training Z^{novel} suitable for composition. Finally, the model will be deployed to classify all encountered classes based on all primitive sets using the composition score.

4. Experiments

4.1. Dataset and Implementation Details

Datasets are listed in Tab.3. Our method is based on the code of CEC (Zhang et al., 2021). For *mini*ImageNet, we follow NC-FSCIL (Yibo Yang, 2023) to utilize ResNet12 (He et al., 2016) as the backbone network, and we set $\lambda_1 = \lambda_2 = 2.0$, $\alpha = 0.8$. For CIFAR100, we follow NC-FSCIL (Yibo Yang, 2023) to utilize ResNet12 as the backbone network, and we remove the pooling operation for the first two residual blocks following ResNet20 used in (Zhang et al., 2021), to keep the spatial resolution of the output map. We set $\lambda_1 = \lambda_2 = 2.0$, $\alpha = 0.6$. For CUB200, we follow CLOM (Zou et al., 2022) to scale the learning rate of the backbone network to 10% of that in the FC layer, due to the pretraining from ImageNet following (Zhang et al., 2021). We set $\lambda_1 = \lambda_2 = 0.01$, $\alpha = 0.5$.

Table 4. Comparison on the CIFAR100 dataset.

Backbone	Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	PD ↓
ResNet20	DeepEMD(2020)	69.75	65.06	61.20	57.21	53.88	51.40	48.80	46.84	44.41	25.34
	WaPR(2023)	74.21	69.96	65.86	61.92	58.74	55.79	53.50	51.51	49.33	24.88
	MetaFSCIL(2022a)	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	24.53
	Comp-FSCIL	**76.00**	**71.75**	**67.67**	**63.76**	**60.99**	**57.98**	**55.98**	**54.09**	**51.61**	**24.39**
ResNet18	SoftNet((Yoon, 2023))	72.62	67.31	63.05	59.39	56.00	53.23	51.06	48.83	46.63	25.99
	ALICE(2022)	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	24.90
	WaPR(2023)	80.31	75.86	71.87	67.58	64.39	61.34	59.15	57.10	54.74	25.57
	Comp-FSCIL	**80.93**	**76.52**	**72.69**	**68.52**	**65.50**	**62.62**	**60.96**	**59.27**	**56.71**	**24.22**
ResNet12	NC-FSCIL(2023)	**82.52**	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	26.41
	Comp-FSCIL	82.30	**78.58**	**74.47**	**70.27**	**67.29**	**64.49**	**62.78**	**61.38**	**59.05**	**23.25**

Table 5. Comparison with state-of-the-art works on the CUB200 dataset. PD: lower performance drop indicates less forgetting.

Backbone	Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	PD ↓
ResNet18	D-DeepEMD (Zhang et al., 2020)	75.35	70.69	66.68	62.34	59.76	56.54	54.61	52.52	50.73	49.20	47.60	27.75
	MetaFSCIL (Chi et al., 2022b)	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	23.26
	SoftNet(Yoon, 2023)	78.07	74.58	71.37	67.54	65.37	62.60	61.07	59.37	57.53	57.21	56.75	21.32
	WaPR(Kim et al., 2023)	77.74	74.15	70.82	66.90	65.01	62.64	61.40	59.86	57.95	57.77	57.01	20.73
	GKEAL(Zhuang et al., 2023)	78.88	75.62	72.32	68.62	67.23	64.26	62.98	61.89	60.20	59.21	58.67	20.21
	NC-FSCIL (Yibo Yang, 2023)	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	21.01
	CLOM (Zou et al., 2022)	79.57	76.07	72.94	69.82	67.80	65.56	63.94	62.59	60.62	60.34	59.58	19.99
	Comp-FSCIL	**80.94**	**77.51**	**74.34**	**71.00**	**68.77**	**66.41**	**64.85**	**63.92**	**62.12**	**62.10**	**61.17**	**19.77**
Swin-T	CLOM (Zou et al., 2022)	86.28	82.85	80.61	77.79	76.34	74.64	73.62	72.82	71.24	71.33	70.50	15.78
	Comp-FSCIL	**87.67**	**84.73**	**83.03**	**80.04**	**77.73**	**75.52**	**74.32**	**74.55**	**73.35**	**73.15**	**72.80**	**14.87**

4.2. Comparison with State-of-the-Art Methods

The comparison with state-of-the-art works is in Tab.2, 4 and 5, with all sessions in the incremental learning. From these tables, we can see that we consistently outperform current works by over 1.5% in terms of the last session's performance, where all classes are taken into account. Moreover, we utilize the Swin Transformer (Liu et al., 2021) (the tiny version, denoted as Swin-T) as an example to evaluate our method given the pretraining of Large Vision Model (LVM, ImageNet1k in our experiments). For a fair comparison, experiments of Transformers are conducted on CUB200 where the ImageNet pretraining is utilized by other works. To compare with current works, we implement CLOM (Zou et al., 2022) as the state-of-the-art method that has the highest last-session accuracy in Tab.5. We can still outperform it by 2.0% in terms of the last-session accuracy. PD denotes the Performance Drop. It means the first session's accuracy subtracts the last session's accuracy, with lower values indicating less forgetting. We can also achieve the least forgetting due to the reuse of primitives.

4.3. Ablation Study

The ablation study is reported in Tab.6. We include the performance of the *Overall* accuracy, referring to the last-session accuracy; the *Base* accuracy, referring to the base-session accuracy (S0); and the *Novel* accuracy, referring to the accuracy of classifying all novel-class samples into novel

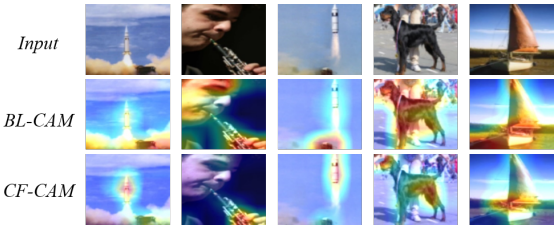


Figure 3. Visualization of class-activation-map (CAM). BL: Baseline model; CF: Our compositional model. CF-CAM activates smaller regions than BL-CAM and filters out sample-specific regions such as background, validating the focus on shared patches.

classes, equivalent to the k-way n-shot evaluation in few-shot learning. We can see each module has its contribution for all training scenarios and all performance measurements, by means of avoiding forgetting via maintaining learned primitives and fast learning of compositions. We also report the sensitivity study of the power transformation parameter α in Fig.4, indicating it could further enhance model robustness to primitive noises.

4.4. Primitive Effectiveness

4.4.1. Quantitative Analysis

Quantitatively, we test the recognition by throwing away sample-specific candidate primitives through W^A, and report the performance against the number of remaining prim-

Table 6. Ablation study of modules on the last incremental session of three datasets.

Method	CUB200			CUB200 (Swin-T)			CIFAR100			*mini*ImageNet		
	Overall	Base	Novel	Overall	Base	Novel	Overall	Base	Novel	Overall	Base	Novel
Baseline	57.18	79.48	45.67	69.18	85.65	61.71	53.98	79.92	44.07	57.72	82.53	42.82
+ Composition	59.25	80.13	49.01	71.42	87.08	63.75	55.30	81.43	47.52	58.84	82.85	44.37
+ Reusing	**61.17**	**81.06**	**51.40**	**72.80**	**87.79**	**65.12**	**59.05**	**82.30**	**51.05**	**60.61**	**84.00**	**46.52**

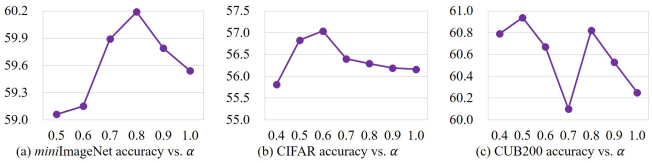


Figure 4. Sensitivity study of the power transformation parameter α, which further improves model robustness to primitive noises.

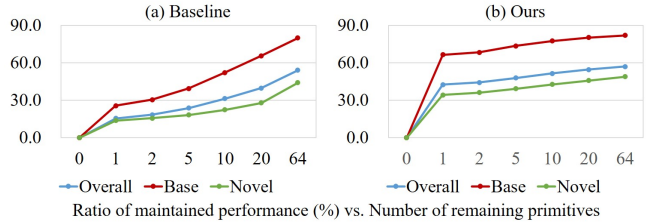


Figure 5. Quantitatively validating our method focuses more on important primitives by only maintaining them on CIFAR100.

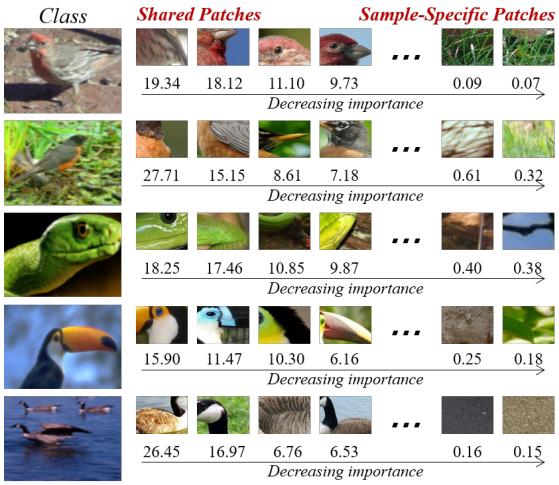


Figure 6. Image patches retrieved in each class, where important patches (candidate primitives) can represent shared patterns.

itives in Fig.5. We can see our model achieves higher performance given the same number of remaining primitives, indicating our model focuses more on important patches.

4.4.2. Qualitative Analysis

To qualitatively validate the discovered primitives, we compare the activation map of the baseline model (BL) and our compositional FSCIL model (CF), by the class-activation-map (CAM) (Zhou et al., 2016). Since CAM relies on the dot-product between W_y and X which cannot be directly applied to our CKA-based model, we rewrite the numerator in CKA as $\sum_i^n [\sum_k^N (\sum_j^d \tilde{X}_{ij} \tilde{Z}_{kj})^2]$ where the $[\cdot]$ denotes the designed CF-CAM. Based on the CF-CAM visualization in Fig.3, we can see CF-CAM shows smaller activated regions compared with BL-CAM, which filers out sample-specific regions such as background areas. This phenomenon validates that our compositional FSCIL method could filter out sample-specific patches and highlight important (shared) ones, which improves primitives and compositions.

Then, we visualize primitives by retrieving image patches according to W^A, which is the importance value in Eq.11, for each class with samples from this class. We report the retrieved patches in Fig.6 sorted by the importance values. We can see that candidate primitives with large importance values can indeed represent shared patterns of each class, such as furs, eyes, and beaks. In contrast, patches at the end of the sort refer to sample-specific patterns such as grass, sand, or other patterns irrelevant to the class, which are filtered out by our model through the small importance value. This phenomenon further validates that our compositional model can effectively focus on important candidate primitives and filter out sample-specific ones.

Table 7. Verification of primitive reuse on CIFAR100.

Ratio (%)	Baseline	All Match	Max Match	+ Comp	+ Reuse
1	100.0	100.0	100.0	100.0	100.4
2	99.80	99.69	99.44	100.0	100.04
5	99.63	99.56	99.50	100.04	100.10
10	98.14	98.14	98.44	100.14	100.10
20	93.56	95.72	96.01	100.35	101.30
50	82.18	90.83	92.17	99.02	99.74
80	69.80	80.42	82.39	87.86	89.48
90	62.38	67.28	70.01	73.14	76.83

4.5. Primitive Reusability and Composition

4.5.1. Quantitative Analysis

To verify our model can better transfer and reuse primitives than ordinary methods, we replace novel-class primitives with the nearest primitives in base classes. For the ordinary method, we use the class prototype as their primitives. We plot the ratio of replaced primitives and the ratio of re-

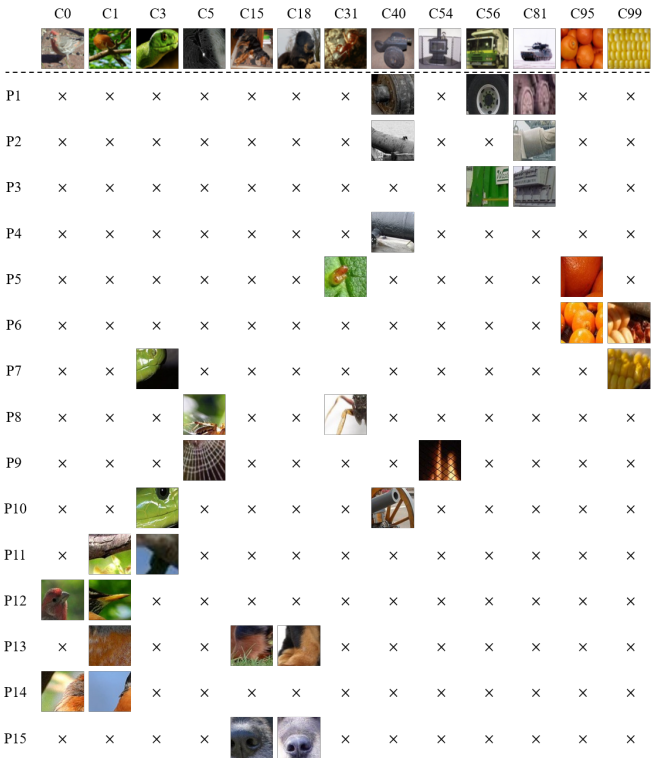


Figure 7. Image patches are retrieved across classes to validate the composition of primitives, where each row refers to a primitive. Based on it, we can interpret novel-class recognition such as *The spider web is composed of a net and a spider like an ant* (P8, P9). An extended version is included in the appendix.

maining novel-class performance in Tab.7. We can see our model can achieve higher performance when primitives are replaced, validating the reusability of primitives, which lays the ground for our interpretation by primitives.

We also compare our method with other set-similarity-based methods (Afrasiyabi et al., 2022) in Tab.7. We can see the compositional structure (i.e., reuse of primitives) is lower than ours (the last row, when reusing ratio=90%, the recovered ratio is much lower than ours). Therefore, the naturally arisen reusability is far from being perfect, which needs to be strengthened by our methods.

Moreover, since in Tab.7 the performance begin to drop only when the replace ratio reaches 80%, we only need to re-learn 20 of primitives and can reuse other base-class primitives. Since the base session is fixed for each novel class, this means the novel-class primitive space has been compressed to 20 of its original size. These verify the potential to compress the primitive size.

4.5.2. Qualitative Analysis

Finally, in Fig.7, we retrieve primitives across classes to validate the composition of primitives. We first retrieve important primitives within each class according to the importance value in each column. Then, primitives retrieved across classes with the smallest distances are in the same row. $\times$ denotes the primitive is not activated in the given class. We can see although classes are not the same, the image patches of primitives are similar, validating the reusability of primitives. Moreover, primitives reused across classes can be viewed to compose each novel class, therefore we can interpret the recognition of each novel class in the following way: P8 + P9: *The spider web is composed of a net and a spider like an ant*. P1 + P2 + P3: *A tank is composed of a vehicle with armor and a gun barrel.*

Table 8. Ablate the primitive number on CIFAR100.

Ratio (%)	Base Classes	Novel Classes	All Classes
1	57.20	34.90	33.20
4	75.01	40.22	48.32
9	75.35	41.47	51.55
16	76.00	42.57	51.61
25	75.25	42.32	51.49
36	75.50	42.00	51.28
49	76.16	41.37	51.23
64	76.18	42.27	51.60
81	76.28	41.27	51.06
100	75.76	41.10	51.37

4.5.3. Number of Primitives

We report experiments on CIFAR in Tab. 8 to ablate primitive numbers. Since the feature map of CIFAR is at the size of 8×8, we increase the number of primitives squarely. We can see the performance reaches the top after the primitive num reaches 9 or 16, which is not a heavy burden compared with the parameters in the deep networks (e.g., 512 * 9 = 4k parameters in the primitive size vs. millions of parameters in the ResNet backbone). If the primitive size is too small, the model will lack the flexibility to represent base knowledges. If the primitive size keeps increasing, although the capacity is larger, it also imports less effective primitives and thus fails to keep improving the performance. In our experiments, we choose 16 as the primitive size.

5. Conclusion

To imitate human's ability of compositional learning, we propose a compositional FSCIL method to divide knowledge into primitives and learn novel knowledge by the composition of primitives. Experiments on three datasets validate the rationale and effectiveness of our method.

Acknowledgements

This work is supported by National Natural Science Foundation of China under grants 62206102, 62376103, 62302184, U1936108 and Science and Technology Support Program of Hubei Province under grant 2022BAA046.

Impact Statement

We propose a cognitive-inspired method to handle the FSCIL problem by simulating humans' ability to compositional learning. This work can also be adopted in other fields like few-shot learning, and image retrieval, since the compositional structure of knowledge exists in many other domains. The limitation of this work is the neglect of the many-shot scenarios where the update of primitives cannot be ignored. However, as our method can provide a good initialization for the future update of primitives, it will also benefit the many-shot scenarios.

References

Afrasiyabi, A., Larochelle, H., Lalonde, J.-F., and Gagné, C. Matching feature sets for few-shot image classification. *IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Biederman, I. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94 (2):115, 1987.

Can Peng, e. a. Few-shot class-incremental learning from an open-set perspective. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pp. 382–379. Springer, 2022.

Cao, K., Brbic, M., and Leskovec, J. Concept learners for few-shot learning, 2021.

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision*, pp. 233–248, 2018.

Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., and Tang, J. Metafscil: A meta-learning approach for few-shot class incremental learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14146–14155, 2022a. doi: 10.1109/CVPR52688.2022.01377.

Chi, Z., Gu, L., Liu, H., Wang, Y., Yu, Y., and Tang, J. Metafscil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14166–14175, 2022b.

Davari, M., Horoi, S., Natik, A., Lajoie, G., Wolf, G., and Belilovsky, E. Reliability of cka as a similarity measure in deep learning, 2022.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

Fodor, J. A. *The language of thought*, volume 5. Harvard University Press, 1975.

He, J., Kortylewski, A., and Yuille, A. L. COMPAS: representation learning with compositional part sharing for few-shot classification. *CoRR*, abs/2101.11878, 2021. URL https://arxiv.org/abs/2101.11878.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Hoffman, D. D. and Richards, W. A. Parts of recognition. *Cognition*, 18(1-3):65–96, 1984.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 831–839, 2019.

Kato, K., Li, Y., and Gupta, A. Compositional learning for human object interaction. In *Proceedings of the IEEE/CVF European Conference on Computer Vision*, pp. 234–251, 2018.

Kim, D.-Y., Han, D.-J., Seo, J., and Moon, J. Warping the space: Weight space rotation for class-incremental few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=kPLzOfPfA2l.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

Mishra, S., Zhu, P., and Saligrama, V. Learning compositional representations for effective low-shot generalization, 2022.

Oh, J., Kim, S., Ho, N., Kim, J.-H., Song, H., and Yun, S.-Y. Understanding cross-domain few-shot learning based on domain similarity and few-shot difficulty, 2022.

Purushwalkam, S., Nickel, M., Gupta, A., and Ranzato, M. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3593–3602, 2019.

Raghu, M., Gilmer, J., Yosinski, J., and Sohl-Dickstein, J. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability, 2017.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets, 2015.

Schwartz, E., Karlinsky, L., Feris, R. S., Giryes, R., and Bronstein, A. M. Baby steps towards few-shot learning with multiple semantics. *CoRR*, abs/1906.01905, 2019. URL http://arxiv.org/abs/1906.01905.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

Song, Z., Zhao, Y., Shi, Y., Peng, P., Yuan, L., and Tian, Y. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24183–24192, June 2023.

Tang, L., Wertheimer, D., and Hariharan, B. Revisiting pose-normalization for fine-grained few-shot recognition, 2020.

Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., and Gong, Y. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12183–12192, 2020.

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., and Wierstra, D. Matching networks for one shot learning. In *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 3637–3645, 2016.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. 2011.

Wertheimer, D., Tang, L., and Hariharan, B. Few-shot classification with feature map reconstruction networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8012–8021, 2021.

Yibo Yang, e. a. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *Proceedings of the International Conference on Learning Representations*, 2023.

Yoon. On the soft-subnetwork for few-shot class incremental learning. In *Proceedings of the International Conference on Learning Representations*, 2023.

Zhang, C., Cai, Y., Lin, G., and Shen, C. Deepemd: Few-shot image classification with differentiable earth mover's distance and structured classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12203–12213, 2020.

Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., and Xu, Y. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12455–12464, 2021.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. Learning deep features for discriminative localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929, 2016.

Zhou, D.-W., Wang, F.-Y., Ye, H.-J., Ma, L., Pu, S., and Zhan, D.-C. Forward compatible few-shot class-incremental learning. *arXiv preprint arXiv:2203.06953*, 2022.

Zhuang, H., Weng, Z., He, R., Lin, Z., and Zeng, Z. Gkeal: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7746–7755, June 2023.

Zou, Y., Zhang, S., Chen, K., Tian, Y., Wang, Y., and Moura, J. M. Compositional few-shot recognition with primitive discovery and enhancing. In *Proceedings of the ACM International Conference on Multimedia*, pp. 156–164, 2020.

Zou, Y., Zhang, S., Li, Y., and Li, R. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. In *Proceedings of the International Conference on Neural Information Processing Systems*, 2022.

Appendix for Compositional Few-Shot Class-Incremental Learning

A. Detailed Dataset Description

***mini*ImageNet** (Vinyals et al., 2016) contains 100 classes with 600 samples in each class randomly sampled from ImageNet (Deng et al., 2009), which is relevant to the recognition of general objects such as cats, dogs, instruments and so on. Some samples of *mini*ImageNet are shown in Fig. 1. Following current works (Zhang et al., 2021; Zhou et al., 2022), images are resized to 84 $\times$ 84, and 60 classes are utilized as base classes, while the remaining 40 classes are divided into 8 sessions for incremental learning, where only 5 training samples are available for each novel class.

*mini*ImageNet

Figure 1. Samples of *mini*ImageNet.

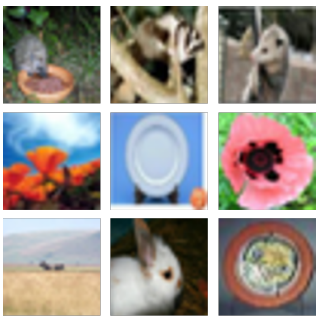
CIFAR100 (Krizhevsky et al., 2009) also contains 100 classes relevant to the recognition of general objects. Samples are shown in Fig. 2, where each image is at the size of 32 $\times$ 32. Similar to *mini*ImageNet, following current works (Zhang et al., 2021; Zhou et al., 2022), 60 classes are selected as base classes, and the remaining 40 classes are divided into 8 incremental sessions with 5 training samples in each novel class.

CUB-200-2011 (CUB200) (Wah et al., 2011) is a fine-grained dataset of birds with 200 classes in all. Samples are shown in Fig. 3, where the input size for each image is 224 $\times$ 224. Following current works (Zhang et al., 2021; Zhou et al., 2022), 100 classes are selected as base classes, and the remaining 100 classes are separated into 10 sessions for incremental learning.

B. More Experiments

B.1. Extended Primitive Visualization

We provided an extended visualization of primitive across. Similar to section 4.5.2, in Fig.4, each column refers to a *mini*ImageNet class, and each row refers to a primitive. We can see that primitives are reused across classes by sharing similar semantic meanings.



CIFAR100

Figure 2. Samples of CIFAR100.

CUB200

Figure 3. Samples of CUB200.

B.2. Sensitivity Study

We also provide the sensitivity study of hyper-parameters from the primitive diversification module on CIFAR100, *mini*ImageNet and CUB200 in Fig.5. We can see these three datasets show similar trends. Take CIFAR100 for an example. λ, as the importance of the CKA similarity, achieves the highest accuracy around 2.0, meaning both the classical and CKA similarity are important in learning effective primitives. Moreover, on the CUB200 dataset, the optimal λ is significantly smaller than that on the other datasets. This is because the ImageNet pretraining is utilized on CUB200, which requires the model to make better use of the pretraining. Since the pretraining is based on the classical similarity function, the weight of the loss generated by the classical similarity function should be larger.

Figure 4. Extended primitive visualization across classes. Image patches are retrieved across classes to validate the composition of primitives, where each row refers to a primitive. Based on it, we can interpret novel-class recognition such as *The spider web is composed of a net and a spider like an ant* (P10, P11).

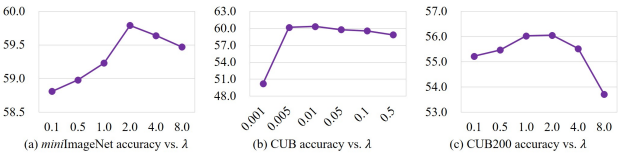


Figure 5. Sensitivity study of hyper-parameters λ.

B.3. Comparison with Compositional Learning Works

To compare our work with other compositional-learning-based works, we also implemented CPDE (Zou et al., 2020) and RPC (Mishra et al., 2022) on the CIFAR100 dataset following the setting provided in section 4.1. The results are reported in Tab. 1, where we can outperform it in terms of the accuracy on each session. This is because CPDE build primitives from the aspect of channels, but we build primitives based on image patches. Although each channel can indeed represent semantic patterns, it still takes the whole image as input, which makes it vulnerable to noisy patches such as background. Moreover, the comparison between primitive sets of CPDE, however, is still modeled as the cosine similarity between every two holistic features, which can hardly prevent it from being affected by noisy patterns. In contrast, our method can efficiently filter out noisy patches and highlight important ones, which therefore benefits our model with higher performance. For the RPC method, this method also learns primitives from the spatial dimension. However, it forces the model to learn and recognize through a fixed dictionary of primitives, which lacks the flexibility to capture the sample-specific primitives. Therefore, our method can also outperform RPC.

Table 1. Comparison with compositional learning works on the CIFAR100 dataset.

Method	S0	S1	S2	S3	S4	S5	S6	S7	S8	PD ↓
CPDE (Zou et al., 2020)	80.85	76.09	71.67	67.69	64.31	61.49	59.08	56.79	54.54	26.31
RPC (Mishra et al., 2022)	80.65	76.22	72.11	68.04	64.61	61.93	59.60	57.41	55.28	25.37
Ours	**82.30**	**78.58**	**74.47**	**70.27**	**67.29**	**64.49**	**62.78**	**61.38**	**59.05**	**23.25**

C. Extended Related Work

Few-shot class-incremental learning (FSCIL) can be roughly grouped into adaptation-based (Hou et al., 2019; Rebuffi et al., 2017; Castro et al., 2018; Tao et al., 2020) and metric-based methods (Zhang et al., 2021; Zou et al., 2022). The first group adapts the model during novel-class training, but the backbone network may be frozen to avoid catastrophic forgetting (Zhou et al., 2022). For example, CEC (Zhang et al., 2021) meta-trains the the graph network for propagating the classifier information according to contexts on base classes, and then transfers the propagation mechanism to novel classes for generating novel-class classifiers. FACT (Zhou et al., 2022) reserves feature space for novel classes to avoid the conflicts between novel classes and base classes, so as to alleviate the catastrophic forgetting brought by the novel-class finetuning. The second group represents each class through prototypes averaged from samples (Zou et al., 2022), which also freezes network parameters to avoid catastrophic forgetting. For example, CLOM (Zou et al., 2022) learns a margin-based feature extractor to improve the representations, and recognizes novel classes by the distance between prototypes and each sample's representation. However, most of current works learn a holistic feature for each input sample, and seldom works studied the compositional structure of the FSCIL models. To the best of our knowledge, we are the first to discover the compositional components of the learned knowledge, and build a compositional model with both higher performance and better interpretability.

Compositional learning aims to learn through primitives (components) of knowledge, which has been well studied in cognitive science (Biederman, 1987; Hoffman & Richards, 1984; Fodor, 1975). Some works applied this concept in other domains. For example, CompCos (Zou et al., 2020) decomposes classes into channels for few-shot learning, which views the cosine similarity between prototypes and input samples as the element-wise comparison between primitive sets. CORL (He et al., 2021) decomposes knowledge into pre-defined visual prototypes learned on base classes, and utilizes pre-defined activation maps for novel-class composition. (Purushwalkam et al., 2019) decomposes visual features to attributes for zero-shot learning, which encourages the visual features to be close to the combination of attribute features. (Kato et al., 2018) decomposes human-object interactions into actions and objects. However, seldom effort has been made for the FSCIL task so far, and most of the current works (Purushwalkam et al., 2019; Kato et al., 2018) rely on the extra attribute or part annotations. Compared with them, our decomposition is from the spatial dimension and does not require additional annotations for primitives.