

# Learning Prompt with Distribution-Based Feature Replay for Few-Shot Class-Incremental Learning

Zitong Huang<sup>1</sup>, Ze Chen<sup>2</sup>, Zhixing Chen<sup>1</sup>, Erjin Zhou<sup>2</sup>, Xinxing Xu<sup>3</sup>

Rick Siow Mong Goh<sup>3</sup>, Yong Liu<sup>3</sup>, Wangmeng Zuo<sup>1✉</sup>, Chun-Mei Feng<sup>3✉</sup>

**Abstract**—Few-shot Class-Incremental Learning (FSCIL) aims to learn new classes with few examples while retaining knowledge of previously encountered ones. Existing studies relied on pure visual networks, while in this paper we solved FSCIL by leveraging the pretrained vision-language model and propose a simple yet effective framework, named Learning Prompt with Distribution-based Feature Replay (LP-DiF). We observe that using CLIP for zero-shot evaluation significantly outperforms leading methods. Then, prompt tuning is involved to further improve its adaptation ability, enabling continuous learning of session-specific knowledge. To prevent the learnable prompt from forgetting old knowledge, we propose a pseudo-feature replay approach. Specifically, we preserve old knowledge of each class by maintaining a feature-level Gaussian distribution with a diagonal covariance matrix, which is estimated by the features of training images and synthesized features generated from a VAE. When progressing to a new session, pseudo-features are sampled from old-class distributions combined with training images of the current session to optimize the prompt, thus enabling the model to learn new knowledge while retaining old knowledge. Experiments on prevalent benchmarks, *i.e.*, CIFAR100, mini-ImageNet, CUB-200, and more challenging benchmarks, *i.e.*, SUN-397 and CUB-200\* proposed in this paper showcase the superiority of LP-DiF, achieving new state-of-the-art (SOTA) in FSCIL. Code is publicly available at <https://github.com/1170300714/LP-DiF>.

**Index Terms**—Few-shot class-incremental learning, continual learning, prompt tuning.

## I. INTRODUCTION

CASS-INCREMENTAL LEARNING (CIL) [13], [44], [62] faces challenges in data-scarce real-world applications, *e.g.*, face recognition systems [60] and smart photo albums [40]. This has led to the emergence of Few-Shot CIL (FSCIL) [42], where models adapt to new classes with limited training data, showcasing their relevance and flexibility in data-scarce scenarios.

In FSCIL, with only a few samples for each incremental task, the main challenge is not just avoiding catastrophic forgetting of previous knowledge [39], [40], [42] but also facilitating plasticity from limited data. Existing studies usually address this by first pre-training a classifier on a base

Zitong Huang, Zhixing Chen and Wangmeng Zuo are with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China (e-mail: cswmzuo@gmail.com).

Ze Chen and Erjin Zhou are with the Megvii Research, Megvii Technology Limited, Beijing, China.

Xinxing Xu, Rick Siow Mong Goh, Yong Liu and Chun-Mei Feng are with the Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A\*STAR), Singapore (e-mail: fengcm.ai@gmail.com)

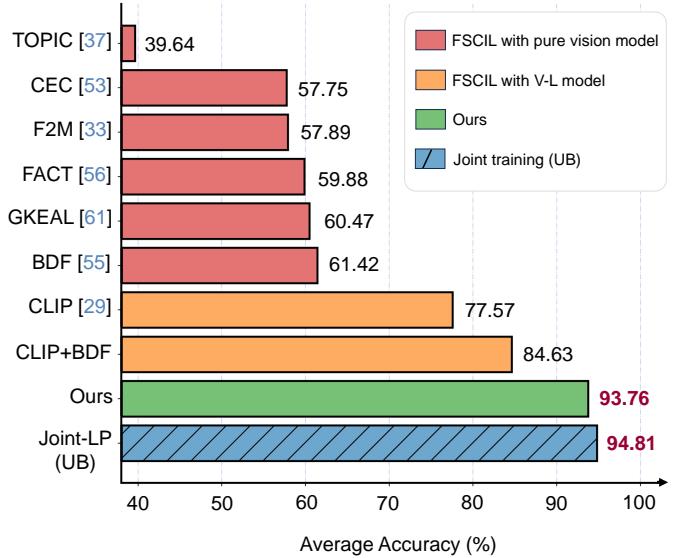


Fig. 1. Comparison of FSCIL methods in terms of Average Accuracy (%) on the test set of *mini-ImageNet* benchmark [35] under 5-shot setting. Red-highlighted bars indicate SOTA vision-based models (*e.g.*, CNN [20]), while orange highlights show V-L pretrained models enhancing FSCIL, significantly outperforming those vision-based counterparts. Our method, marked in green, achieves 93.76%, surpassing CLIP+BDF by 9.13%, and comparable to the theoretical upper bound (UB) that highlights in blue achieved through learning prompt in joint-training manner.

set with numerous images for a robust foundation [24], [36], [40], [56], [60], [61], [63], [65]. Subsequent adaptations, *e.g.*, knowledge distillation [60], class relationship modeling [40], [56], and specific optimization [36], are then applied to the sparse incremental session data to boost performance while maintaining previously acquired knowledge.

This work diverges from approaches that solely rely on visual networks [20], opting instead to leverage the capabilities of a Vision-Language (V-L) pretrained model, *i.e.*, CLIP [32], [64], to develop a few-shot incremental learner. Comparing with existing state-of-the-art techniques (see the Red-highlighted bars in Fig. 1), we observed that by simply crafting the manual prompt “A photo of a [CLS]” as textual input and performing zero-shot evaluation on the widely used FSCIL benchmark, *mini-ImageNet* [35] test set, CLIP (refer to the orange CLIP bar in Fig. 1) substantially outperforms all these SOTA methods, with a notable 16.15% performance boost over BiDistFSCIL (BDF) [60]. This finding indicates that the generalization abilities of V-L pretrained models are

highly beneficial for FSCIL, *e.g.*, naturally mitigating the plasticity issues caused by limited training samples. Further, from Fig. 1, simply replacing the existing backbones of current SOTA methods with a pretrained image encoder, initializing and learning the classifier with the corresponding text encoding of manual prompt can further enhance performance (7.16% gain to CLIP) but still lag behind the UB (9.13% lower than the UB). Therefore, how to derive an efficient and lightweight prompt for FSCIL continues to be a compelling challenge.

Based on the above preliminary results, this paper proposes a simple yet effective FSCIL framework by learning a lightweight prompt built upon the V-L pre-trained models. Unlike CLIP, as well as simply integrating CLIP with existing methods (refer to the orange bar in Fig. 1, we resort to improving prompt tuning [64] for meeting the requirements of FSCIL. Specifically, for session  $t$ , we take the prompt in session  $t-1$  for initialization, combine it with  $[CLS]$  to create the full-text input for each class, and then optimize learnable prompt with training data.

To prevent the learnable prompt from forgetting prior knowledge in a new session, we also propose a pseudo-feature replay technique. Specifically, observing that the image features extracted by the image encoder of CILP for each class seem to follow a Gaussian distribution (refer to Fig. 3), we attempt to estimate its mean vector and diagonal covariance matrix (*i.e.* parameters of Gaussian distribution) to fit the training data of each class. To this end, a VAE [27], [47] comprised of the V-L model and lightweight MLPs are proposed to synthesize features based on the few training samples and text information, permitting the usage of real image features as well as synthesized features to estimate Gaussian distribution parameters more accurately. When the model trains on a new session, pseudo-image features from the old-class distributions are sampled as old-knowledge replay to constrain the optimization direction of the prompt, avoiding learning towards catastrophic forgetting. The results in Fig. 1 showcase that our approach improves zero-shot evaluation for CLIP by 16.19% and for CLIP+BDF by 9.13%. Notably, our method is merely 1.05% lower than the upper bound (Joint-LP, *i.e.*, learning prompt on training data of each session jointly).

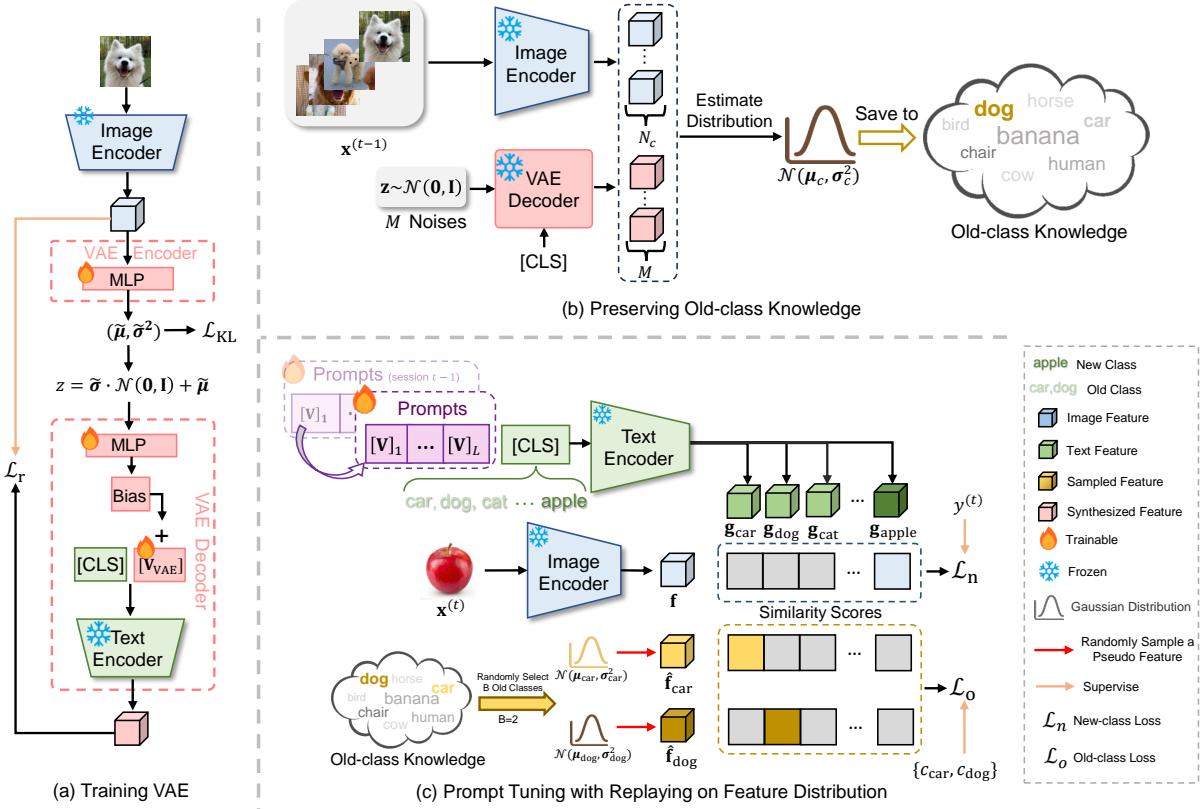
In a nutshell, the main contributions of this paper are summarized as follows:

- 1) We empirically show that pretrained V-L models, *e.g.* CLIP, are beneficial for FSCIL due to its considerable generalization ability, inspiring us to propose a simple yet effective V-L based FSCIL method named LP-DiF.
- 2) We adopt prompt tuning for allowing the model to continually capture specific knowledge of each session, and present a feature replay technique to prevent catastrophic forgetting. By constructing feature-level Gaussian distribution for each class, pseudo feature replay can be combined with training images of current session to learn new knowledge while retaining old knowledge.
- 3) Extensive evaluations and comparisons on three prevalent FSCIL benchmarks (CIFAR-100, CUB-200 and *mini*-ImageNet) and two proposed more challenging benchmarks (SUN-397 and CUB-200\*) show the superiority of our methods in comparison to state-of-the-arts.

## II. RELATED WORK

**Few-Shot Class-Incremental Learning.** The few-shot class-incremental learning methods (FSCIL) aims to train a model in a class-incremental manner [13], [62] with only a few samples for each new tasks [42]. Existing studies can be categorized into four families, *i.e.*, dynamic network-based methods, meta-learning-based methods, feature space-based methods, and replay-based methods. In specific, dynamic network structure [18], [40], [51], [52] is proposed to adaptive learn the new knowledge by dynamically expanding the network structure, so that the new knowledge is preserved by the new network structure. Meta learning-based methods [12], [21], [31], [55], [58], [65], [67] employ a session sampling scheme, where a sequence of sessions are sampled from the base session, aiming to mimic the incremental learning process during evaluation, to allows the model to learn how to retain old knowledge under the condition of a small number of new data samples. Feature space-based methods [2]–[4], [11], [26], [59], [61], [63], [66], focus on mapping the original image into a condensed feature space while preserving its essential attributes, which ensures that the representations of old category data are not disrupted when the model is trained on new data. Replay-based methods [10], [14], [29] retain or produce significant data from prior tasks to be reintroduced in the ongoing task. These methods are dedicated to selecting the most representative samples of old categories, or utilizing generative models to produce high-quality pseudo-samples of old categories. While these methods have shown commendable performance, all those studies are based on feature extractors and classifiers built from deep networks trained in the base session. Due to the scarcity of incremental class samples, the feature representation ability is limited. In contrast, we propose to construct an incremental learner on a VL pre-trained model [32], [64] that offers inherent merits for FSCIL, *i.e.*, endowing the image encoder with powerful feature representation abilities.

**Replay-based Incremental Learning.** The replay-based approach in incremental learning leverages knowledge from previous tasks to mitigate catastrophic forgetting in models [1], [5]–[9], [19], [22], [23], [34], [37]. A basic data replay approach involves retaining a concise exemplar set, capturing essential samples from prior tasks [6], [7], then, the classification model is trained on the combination of exemplars and the data of the current task. Different from directly storing the real instances, several following works [19], [22], [25], [37] leveraged a generative model [17], [27] for generating data from previous tasks. Compared to methods based on real image replay, pseudo replay reduces storage needs by eliminating the requirement for exemplars and enriches the diversity of samples from previous tasks. Yet, the overhead of training the image generator and dynamically producing pseudo images introduces additional computational demands and prolongs training time. Instead of retaining an image generator, we represent the feature representation for each class using a Gaussian distribution, utilizing it to sample pseudo-features for rehearsing prior knowledge. Moreover, drawing samples from this distribution is computationally



**Fig. 2. Overview** of our proposed LP-DiF. **(a)** In each session, we first train a VAE [27], [47] comprised of the V-L model and lightweight components, *i.e.*, MLPs and learnable prompt, based on few training data and textual information of this session. **(b)** We preserve the knowledge of each class by estimating their feature-level statistical distribution. The mean vector and diagonal covariance matrix of the distribution are estimated by both the features of real images and the synthesized features from trained VAE. **(c)** Prompt is trained jointly with the combination of the real image of the current session and the pseudo-features sampled from old-class distributions.

efficient, offering our method an effective way for handling prevent catastrophic forgetting.

**Incremental Learning via Pre-trained Model.** Recent studies have explored constructing incremental learners using pre-trained models [16], [38], [41], [45], [46], [48], [49], [53], [57]. The core idea of these methods is to leverage a pre-trained backbone, *e.g.*, ViT [15], for robust image feature extraction, while only fine-tuning a selected set of parameters to adapt to new tasks. For example, L2P [49] employs a fixed pre-trained ViT as its backbone and sustains a dynamic prompt pool with various sets of adaptable prompts. Some following works [38], [45] built upon this concept, applying it to the VL pretrained model [32], leveraging linguistic knowledge to bolster classification performance. In addition, Yang. *et al.* [53] built a Bayesian model based on a fixed feature extracted by a pretrained backbone. During the test stage, they use this Bayesian model as the classifier to mitigate the forgetting problem. The above studies underscore the significant advantages of using pretrained models to boost performance in standard CIL scenarios. As for FSCIL, we inherit the advantages of pretrained models in CIL. Inspired by Yang. *et al.* [53], we further explore the potential characteristics of fixed feature, maintain a feature-level Gaussian distribution for each class to preserve the old knowledge, and use it to generate

pseudo features to mitigate the catastrophic forgetting.

### III. PROPOSED METHOD

**Problem Formulation.** The purpose of FSCIL is to continually learn knowledge of new classes from few samples, while simultaneously preventing the model from forgetting knowledge of old classes. Formally, a model is trained by a sequence of training data  $\mathcal{D}_{\text{Train}} = \{D_{\text{Train}}^{(t)}\}_{t=0}^T$  continually, where  $D_{\text{Train}}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=0}^{N^{(t)}}$  denotes the training set of session (task)  $t$ .  $\mathbf{x}_i$  is a training image with corresponding class label  $y_i \in \mathcal{C}^{(t)}$ , where  $\mathcal{C}^{(t)}$  denotes the class space of  $D_{\text{Train}}^{(t)}$ . For different sessions, the class spaces are non-overlapping, *i.e.*  $\forall t_1, t_2 \in \{0, 1, \dots, T\}$  and  $t_1 \neq t_2$ ,  $\mathcal{C}^{(t_1)} \cap \mathcal{C}^{(t_2)} = \emptyset$ . Typically,  $D_{\text{Train}}^{(0)}$  of the first session (*i.e.*  $t = 0$ ), which is usually referred to as the base session, contains a substantial amount of training data. While  $D_{\text{Train}}^{(t)} (t > 0)$  of the incremental sessions only contains few training sample, organized as the  $N$ -Way  $K$ -shot format, *i.e.*,  $N$  classes in each incremental session with each class comprising  $K$  training images. Following the formulation of standard class-incremental learning, in session  $t$ , only  $D_{\text{Train}}^{(t)}$  and an optional memory buffer used to store the old knowledge (*e.g.* exemplar) can be accessed. After finishing training on  $D_{\text{Train}}^{(t)}$ , the model is evaluated on a test set  $D_{\text{Test}}^{(t)}$ ,

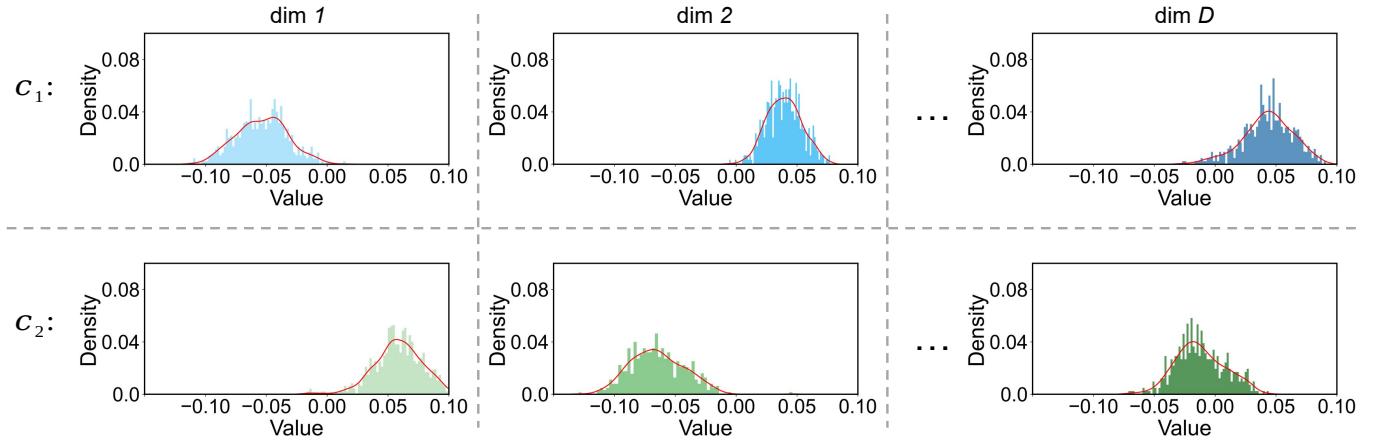


Fig. 3. **Histogram visualization** of the *statistical distribution* of image features. We take the image features with different dimensions (dim) of classes  $c_1$  and  $c_2$  as example selected from the *mini-ImageNet* [35] benchmark by the image encoder of CLIP (ViT-B/16) [32]. Each sub-figure shows the distribution with histogram of corresponding random variable  $Z_{cd}$ , where  $c$  and  $d$  denotes the index of class and feature dimension respectively. Obviously, 1) each dimension of the image features per class approximates Gaussian distribution; 2) distributions of same dimension vary in different classes, e.g.,  $Z_{c11}$  vs.  $Z_{c21}$ .

the class space of which is union of all the classes encountered so far, *i.e.*  $\mathcal{C}^{(0)} \cup \mathcal{C}^{(1)} \dots \cup \mathcal{C}^{(t)}$ .

In this section, we propose a FSCIL method based on the V-L pretrained model, *e.g.* CLIP [32]. We assume that the class names are accessible during the training and testing of each session. Formally, CLIP contains an image encoder  $E_{\text{Img}}(\mathbf{x})$  and a text encoder  $E_{\text{Txt}}(\mathbf{p})$ , which are pretrained jointly with a huge amount of image-text pairs in contrastive learning manner. An image  $\mathbf{x}$  is fed into the image encoder, obtaining the corresponding  $L_2$ -normalized feature  $\mathbf{f}$ .  $\mathbf{p}$  is a text token which is obtained by tokenizing a sentence like “A photo of a [CLS].”, where [CLS] represents a certain class name. We replace [CLS] by each class name respectively and obtain a set of text tokens  $\{\mathbf{p}_c\}_{c=1}^C$ , where  $C$  denotes the total number of classes encountered so far. Then,  $\{\mathbf{p}_c\}_{c=1}^C$  are fed into the text encoder, obtaining the corresponding  $L_2$ -normalized text feature  $\{\mathbf{g}_c\}_{c=1}^C$ . Finally, the prediction score of class  $c$  is computed by:

$$p(y = c | \mathbf{x}) = \frac{\exp(\langle \mathbf{f}, \mathbf{g}_c \rangle / \tau)}{\sum_{j=1}^C \exp(\langle \mathbf{f}, \mathbf{g}_j \rangle / \tau)}, \quad (1)$$

where  $\langle \cdot, \cdot \rangle$  denotes the cosine similarity of the two features and  $\tau$  is the temperature parameter.

#### A. Approach Overview

Although CLIP has demonstrated its superior performance on FSCIL in Fig. 1, using hand-crafted prompt is sub-optimal for transfer the knowledge to each incremental session. So we replace the hand-crafted prompt with a set of learnable vectors  $\mathcal{V} = \{[\mathbf{V}]_l\}_{l=1}^L$  [64], where  $[\mathbf{V}]_l$  ( $l \in \{1, \dots, L\}$ ) denotes one learnable vector, and  $L$  is the number of vectors. Hence, the expression for the text prompt is modified to:

$$\mathbf{p}(\mathcal{V}) = [\mathbf{V}]_1 [\mathbf{V}]_2 \dots [\mathbf{V}]_L [\text{CLS}], \quad (2)$$

To learn  $\mathcal{V}$  on  $\mathcal{D}_{\text{Train}}$ , an intuitive approach is to sequentially tune the prompt using training data from each incremental

session to continually acquire new knowledge. Specifically, at the beginning of session 0, we initialize  $\mathcal{V}$  randomly; while for each following session  $t$  ( $t > 0$ ), we use the  $\mathcal{V}$  trained in the previous session (*e.g.* session  $t-1$ ) to initialize the  $\mathcal{V}$  for current session. In a certain session  $t$ , given a pair of training sample  $(\mathbf{x}_i, y_i)$  from  $\mathcal{D}_{\text{Train}}^{(t)}$ , prompt is optimized on by minimizing  $\mathcal{L}_n$ :

$$\mathcal{L}_n = -\log \frac{\exp(\langle \mathbf{f}_i, E_{\text{Txt}}(\mathbf{p}_{y_i}(\mathcal{V})) \rangle / \tau)}{\sum_{c=1}^{|U_{s=0}^t \mathcal{C}^{(s)}|} \exp(\langle \mathbf{f}_i, E_{\text{Txt}}(\mathbf{p}_c(\mathcal{V})) \rangle / \tau)}, \quad (3)$$

where  $\mathbf{f}_i$  denotes the  $L_2$ -normalized image feature of  $\mathbf{x}_i$ , and  $\mathbf{p}_c(\mathcal{V})$  denotes the prompt corresponding to class  $c$ .

However, using only the  $\mathcal{D}_{\text{Train}}^{(t)}$  to optimize the prompt in session  $t$  will inevitably lead to catastrophic forgetting. Ideally, learning prompt with all training data from both previous and current sessions (*e.g.*  $\bigcup_{s=0}^t \mathcal{D}_{\text{Train}}^{(s)}$ ) can address this issue, but this is not allowed under the protocol of FSCIL. Therefore, this paper adopts a compromise solution, proposing to record old knowledge by maintaining statistical distributions of old classes instead of directly storing origin images. We setup a feature-level Gaussian distribution to represent each old class, which is represented by a mean vector and a diagonal covariance matrix. We name it the **old-class distribution**. The mean vector and diagonal covariance matrix of the old-class distribution are estimated jointly from the features of real images as well as synthetic features generated by a VAE decoder. When learning the prompt in a new session, we randomly sample features based on the statistical distribution of old classes to replay old knowledge. Then, the sampled features of old classes and the real features of new classes will jointly optimize the prompt, thereby learning new knowledge while also replaying old knowledge. In the following, We will introduce how to obtain the old-class distribution in Sec III-B, and how to learn prompt in Sec III-C.

### B. Estimation of Old-Class Distribution

In each session  $t$ , we should estimate the feature-level statistical distribution for each class of  $\mathcal{D}_{\text{Train}}^{(t)}$ . Given a certain class label  $c \in \mathcal{C}^{(t)}$ , the corresponding training images  $\{\mathbf{x}_i\}_{i=1}^{N_c}$  are fed into the image encoder  $E_{\text{Img}}(\mathbf{x})$  to obtain their  $L_2$ -normalized features  $\{\mathbf{f}_i\}_{i=1}^{N_c}$ , where  $N_c$  denotes the number of training images of class  $c$ ,  $\mathbf{f}_i = [f_{i1}, f_{i2}, \dots, f_{iD}]^T$  and  $D$  is the feature dimension (e.g.  $D = 512$  for ViT-B/16). Intuitively, we assume that the features of class  $c$  follow a multivariate distribution  $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ , where  $\boldsymbol{\mu}_c \in \mathbb{R}^D$  denotes the mean vector and  $\boldsymbol{\Sigma}_c \in \mathbb{R}_{\geq 0}^{D \times D}$  denotes the covariance matrix. As shown in Fig. 3, we observe that each dimension of these features of each class approximates a Gaussian distribution, and distributions of same dimension vary in different classes. Thus, each dimension of the feature can be treated as independently distributed, and the covariance matrix  $\boldsymbol{\Sigma}_c$  can be simplified to a diagonal matrix and be represented by a vector  $\boldsymbol{\sigma}_c^2 = [\sigma_{c1}^2, \sigma_{c2}^2, \dots, \sigma_{cD}^2]^T$ , which is diagonal values of  $\boldsymbol{\Sigma}_c$ . We use random variable  $Z_{cd}$  to represent the  $d$ -th dimension of feature, following a specific Gaussian distribution  $\mathcal{N}(\mu_{cd}, \sigma_{cd}^2)$ , where  $\mu_{cd}$  denotes the mean value of the  $d$ -th dimension. Then,  $\mathbf{Z}_c = [Z_{c1}, Z_{c2}, \dots, Z_{cD}]$  represents the random variable of the whole feature following  $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\sigma}_c^2)$ . Our goal is to estimate the  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\sigma}_c^2$  for each class.

For each class, simply using only  $\{\mathbf{f}_i\}_{i=1}^{N_c}$  to estimate the  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\sigma}_c^2$  may be inadequate due to the scarcity of the data. To tackle with this problem, we utilize a VAE [27], [47] comprised of the V-L models and lightweight MLPs, leveraging the few training data and textual information to synthesize more image features, thereby benefiting the estimation of the distribution. As shown in Fig. 2 (a), in VAE Encoder, an image feature  $\mathbf{f}$  is fed into a MLP, encoded to a latent code  $\mathbf{z}$ , of which distribution is assumed to be a prior  $\mathcal{N}(\mathbf{0}, \mathbf{I})$ :

$$\mathcal{L}_{\text{KL}} = KL(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\sigma}}^2) || \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad (4)$$

where  $KL$  represents the Kullback-Leibler divergence. In VAE Decoder,  $\mathbf{z}$  is fed to another MLP and obtain the bias  $\mathbf{r}$ , which is added to a set of learnable prompt  $\mathcal{V}_{\text{VAE}} = \{[\mathbf{V}_{\text{VAE}}]_l\}_{l=1}^L$ :

$$\mathcal{V}_{\text{VAE}}(\mathbf{z}) = \{[\mathbf{V}_{\text{VAE}}]_l + \mathbf{r}\}_{l=1}^L. \quad (5)$$

Then,  $\mathcal{V}_{\text{VAE}}(\mathbf{z})$  concatenating with the class name [CLS] corresponding to  $\mathbf{f}$  is fed into the text encoder, obtaining the reconstruct feature  $\tilde{\mathbf{f}}$  then calculating the reconstruct loss  $\mathcal{L}_r$ :

$$\mathcal{L}_r = \|\mathbf{f} - \tilde{\mathbf{f}}\|_2. \quad (6)$$

Finally, the total loss  $\mathcal{L}_{\text{VAE}}$  of training the VAE is:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{KL}} + \lambda_r \mathcal{L}_r, \quad (7)$$

where  $\lambda_r$  represents the coefficient of  $\mathcal{L}_r$ .

Using both the features synthesized by the VAE and the real image features, we estimate  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\sigma}_c^2$ . As shown in Fig. 2 (b), for a specific class  $c$ ,  $M$  noise vectors  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and corresponding class name are input into the VAE Decoder, obtaining  $M$  synthesized features  $\{\tilde{\mathbf{f}}_j\}_{j=1}^M$ . Then,  $\boldsymbol{\mu}_c$  and  $\boldsymbol{\sigma}_c^2 = [\sigma_{c1}^2, \sigma_{c2}^2, \dots, \sigma_{cD}^2]^T$  are estimated by:

$$\boldsymbol{\mu}_c = \frac{1}{N_c + M} \left( \sum_{i=1}^{N_c} \mathbf{f}_i + \sum_{j=1}^M \tilde{\mathbf{f}}_j \right), \quad (8)$$

$$\sigma_{cd}^2 = \frac{1}{(N_c + M) - 1} \left( \sum_{i=1}^{N_c} (f_{id} - \mu_{cd})^2 + \sum_{j=1}^M (\tilde{f}_{jd} - \mu_{cd})^2 \right). \quad (9)$$

### C. Learning Prompt with Feature Replay

At session  $t$ , we learn prompt with  $\mathcal{D}_{\text{Train}}^{(t)}$  as well as the distributions of old classes preserved in previous sessions.

- 1) When  $t = 0$ , i.e., the first session, we just follow the approach in Sec.III-A, randomly initializing  $\mathcal{V}$  and learning them with  $\mathcal{D}_{\text{Train}}^{(0)}$  by  $\mathcal{L}_n$  (Eq. (3)).
- 2) When  $t > 0$ ,  $\mathcal{V}$  are initialized from trained weights in session  $t - 1$ . For the new knowledge of  $\mathcal{D}_{\text{Train}}^{(t)}$ , we adopt  $\mathcal{L}_n$ . For the old knowledge of previous sessions, we randomly sample pseudo image features of old classes from their corresponding distributions. As shown in Fig.2 (c), for each selected training image in one batch  $\mathbf{x}_i$ , we first randomly select  $B$  old classes:  $\{c_b\}_{b=1}^B$  and  $c_b \in \cup_{s=0}^{t-1} \mathcal{C}^{(s)}$ . Then, for each selected class  $c_b$ , we randomly sample a pseudo feature  $\hat{\mathbf{f}}_{cb}$  from its feature distribution  $\hat{\mathbf{f}}_{cb} \sim \mathcal{N}(\boldsymbol{\mu}_{cb}, \boldsymbol{\sigma}_{cb}^2)$ . These sampled features with their corresponding class labels are used to calculate the loss  $\mathcal{L}_o$ :

$$\mathcal{L}_o = - \sum_{b=1}^B \log \frac{\exp(\langle \hat{\mathbf{f}}_{cb}, E_{\text{Txt}}(\mathbf{p}_{cb}(\mathcal{V})) \rangle / \tau)}{\sum_{c=1}^{| \cup_{s=0}^{t-1} \mathcal{C}^{(s)} |} \exp(\langle \hat{\mathbf{f}}_{cb}, E_{\text{Txt}}(\mathbf{p}_c(\mathcal{V})) \rangle / \tau)}. \quad (10)$$

Finally, the prompt is optimized by minimizing the loss:

$$\mathcal{L}_{\text{LP}} = \begin{cases} \mathcal{L}_n & \text{if } t = 0, \\ \mathcal{L}_n + \lambda_o \mathcal{L}_o & \text{if } t > 0, \end{cases} \quad (11)$$

where  $\lambda_o$  represents the tradeoff coefficient.

## IV. EXPERIMENTS

### A. Datasets and Metrics

**Datasets.** Following the mainstream benchmark settings [60], we conduct experiments on three datasets, i.e., CIFAR-100 [28], mini-ImageNet [35] and CUB-200 [43], to evaluate our LP-DiF. Tab. II summarizes the details of each selected benchmark.

- CIFAR-100 dataset consists of 100 classes, each of which contains 50,000 training images. Following the previous study [60], there are 60 classes in the base session, and the remaining classes will be divided into 8 incremental sessions, with each incremental session comprising 5 classes.
- CUB-200 is a fine-grained classification dataset containing 200 bird species with about 6,000 training images. Following the previous study [60], there are 100 classes in the base session, and the remaining classes will be divided into 10 incremental sessions, with each incremental session comprising 10 classes.
- mini-ImageNet is a smaller part of ImageNet [35], which has 50,000 training images from 100 chosen classes. Following the previous study [60], there are 60 classes in the base session, and the remaining classes will be divided into 8 incremental sessions, with each incremental session comprising 5 classes.

TABLE I

**COMPARISON ON *mini*-IMAGENET.** “**BACKBONE**.” REPRESENTS THE BACKBONE OF VISUAL MODEL. “**AVG**” REPRESENTS THE AVERAGE ACCURACY OF ALL SESSIONS; THE HIGHER THE VALUE, THE BETTER PERFORMANCE. “**PD**” REPRESENTS THE PERFORMANCE DROP RATE; THE LOWER THE VALUE, THE BETTER PERFORMANCE. “**BS**” IS THE ABBREVIATION OF “**BASELINE**”. “**UB**.” IS THE ABBREVIATION OF “**UPPER BOUND**”.

Methods	Backbone	Accuracy in each session (%) ↑									<b>Avg</b> ↑	<b>PD</b> ↓
		0	1	2	3	4	5	6	7	8		
TOPIC [40]	Res18	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	36.89
CEC [56]	Res18	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.75	24.37
F2M [36]	Res18	72.05	67.47	63.16	59.70	56.71	53.77	51.11	49.21	47.84	57.89	24.21
Replay [30]	Res18	71.84	67.12	63.21	59.77	57.01	53.95	51.55	49.52	48.21	58.02	23.63
MF [12]	Res18	72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19	58.85	22.85
GKEAL [66]	Res18	73.59	68.90	65.33	62.29	59.39	56.70	54.20	52.59	51.31	60.48	22.28
FACT [61]	Res18	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.70	22.07
C-FSCIL [21]	Res12	76.40	71.14	66.46	63.29	60.42	57.46	54.78	53.11	51.41	61.59	14.99
BDF [60]	Res18	74.65	70.70	66.81	63.63	61.36	58.14	55.59	54.23	53.39	62.06	21.26
FCIL [18]	Res18	76.34	71.40	67.10	64.08	61.30	58.51	55.72	54.08	52.76	62.37	23.58
SAVC [39]	Res18	81.12	76.14	72.43	68.92	66.48	62.95	59.92	58.39	57.11	67.05	24.01
NC-FSCIL [54]	Res18	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	67.82	25.71
CLIP ( <i>Bs.</i> ) [32]	ViT-B/16	80.01	79.16	78.89	77.97	77.44	76.83	76.32	76.02	75.45	77.57	<b>4.56</b>
<b>LP-DiF (<i>Ours</i>)</b>	ViT-B/16	<b>96.34</b>	<b>96.14</b>	<b>94.62</b>	<b>94.37</b>	<b>94.06</b>	<b>93.44</b>	<b>92.21</b>	<b>92.29</b>	<b>91.68</b>	<b>93.76</b>	4.66
Joint-LP ( <i>UB. of ours</i> )	ViT-B/16	96.34	96.07	95.75	94.93	94.61	94.26	93.99	93.83	93.56	94.81	2.78

TABLE II

**DETAILS OF SELECTED BENCHMARKS.** THE FIRST THREE LINES ARE COMMONLY USED BENCHMARKS, WHILE THE LAST TWO LINES ARE THE MORE CHALLENGING BENCHMARKS PROPOSED IN THIS PAPER.  $|\mathcal{C}^{\text{All}}|$ ,  $|\mathcal{C}^{\text{Base}}|$  AND  $|\mathcal{C}^{\text{Inc}}|$  DENOTES THE TOTAL NUMBER OF CLASSES, THE NUMBER OF CLASSES IN BASE SESSION, AND THE NUMBER OF CLASSES IN EACH INCREMENTAL SESSION RESPECTIVELY. #**BASE** AND #**INC** DENOTE THE NUMBER OF BASE SESSIONS AND THE INCREMENTAL SESSION RESPECTIVELY. **SHOT** DENOTES THE NUMBER OF TRAINING IMAGES OF EACH INCREMENTAL SESSION. \* REPRESENTS A VARIANT VERSION.

Dataset	$ \mathcal{C}^{\text{All}} $	$ \mathcal{C}^{\text{Base}} $	$ \mathcal{C}^{\text{Inc}} $	#Base	#Inc	Shot
CIFAR-100 [28]	100	60	5	1	8	5
<i>mini</i> -ImageNet [35]	100	60	5	1	8	5
CUB-200 [43]	200	100	10	1	10	5
SUN-397 [50]	397	197	10	1	20	5
CUB-200* [43]	200	0	10	0	20	5

Additionally, this paper also proposes two more challenging benchmarks for FSCIL, *i.e.*, SUN-397 [50] and CUB-200\*.

- SUN-397 is a large-scale scene understanding dataset containing 397 distinct scene classes with about 76,000 training images. We select 197 classes for the base session; the remaining classes will be split into 20 incremental sessions, with each incremental session comprising 10 classes. *We evaluate our method on this benchmark to reveal whether it is effective in scenarios with more classes and more incremental sessions.*
- CUB-200\* is a variant of CUB-200 but excludes the base session. We evenly divide the total 200 classes into 20 incremental sessions, with each session containing 10 categories. Following the previous study [60], there are 100 classes in the base session, and the remaining classes will be divided into 10 incremental sessions, with each incremental session comprising 10 classes. *We use it to*

*evaluate whether our method works in scenarios without the base session.*

**Metrics.** Following existing FSCIL methods [39], [40], [60], we employ the **Avg.**, which is the average accuracy of each session, as primary metric for performance comparison. In addition, we also employ the **performance drop rate** (PD.), which represents the drop of performance of the last session compared to the first session, to reflect the extent of the model’s forgetting of old knowledge.

### B. Implementation Details.

All experiments are conducted with PyTorch on 8× NVIDIA RTX 2080Ti GPUs. We leverage the ViT-B/16 as the image encoder of LP-DiF and adopt SGD with 0.9 momentum to optimize the prompts. The learning rate is initialized by 0.002. For the base session, the batch size is set to 64 and the training epoch is set to 200. As for each incremental session, the batch size and the training epochs are set to 25, 100, respectively. The VAE component is enabled only for incremental sessions. For the hyper-parameters,  $M$  is set to 10;  $B$  and  $\lambda_o$  are set to 8 and 2, respectively;  $L$  is set to 16 following Zhou *et al.* [64];  $\lambda_r$  is set to 1 following Wang *et al.* [47].

### C. Main Results

**Comparison with State-of-The-Arts.** We summarize the results of competing methods on *mini*-ImageNet in Table I. Clearly, employing CLIP (baseline) [32], [41] for zero-shot evaluation alone outperforms all existing FSCIL methods by a large margin in terms of accuracy in each session and Average Accuracy (Avg). Naturally, it achieves a notably lower Performance Drop rate (PD). Our LP-DiF further achieves **16.19%** (77.57% → 93.76%) gains than the CLIP in terms

TABLE III  
**COMPARISON WITH STATE-OF-THE-ART FSCIL METHODS ON CUB-200.** “**BACKBONE**.” REPRESENTS THE BACKBONE OF VISUAL MODEL. “**AVG**” REPRESENTS THE AVERAGE ACCURACY OF ALL SESSIONS; THE HIGHER THE VALUE, THE BETTER PERFORMANCE. “**PD**” REPRESENTS THE PERFORMANCE DROP RATE; THE LOWER THE VALUE, THE BETTER PERFORMANCE. “**BS**” IS THE ABBREVIATION OF “**BASELINE**”. “**UB**.” IS THE ABBREVIATION OF “**UPPER BOUND**”.

<b>Methods</b>	<b>Backbone</b>	<b>Accuracy in each session (%) ↑</b>										<b>Avg ↑</b>	<b>PD ↓</b>	
		0	1	2	3	4	5	6	7	8	9	10		
TOPIC [40]	Res18	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.26	43.92	42.42
CEC [56]	Res18	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	23.57
Replay [30]	Res18	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	23.51
MetaFSCIL [12]	Res18	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	61.93	23.26
FACT [61]	Res18	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	18.96
FCIL [18]	Res18	78.70	75.12	70.10	66.26	66.51	64.01	62.69	61.00	60.36	59.45	58.48	65.70	20.22
GKEAL [66]	Res18	78.88	75.62	72.32	68.62	67.23	64.26	62.98	61.89	60.20	59.21	58.67	66.35	20.21
NC-FSCIL [54]	Res18	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	21.01
BiDistFSCIL [60]	Res18	79.12	75.37	72.80	69.05	67.53	65.12	64.00	63.51	61.87	61.47	60.93	67.34	18.19
SAVC [39]	Res18	81.85	77.92	74.95	70.21	69.96	67.02	66.16	65.30	63.84	63.15	62.50	69.35	19.35
F2M [36]	Res18	81.07	78.16	75.57	72.89	70.86	68.17	67.01	65.26	63.36	61.76	60.26	69.49	20.81
CLIP ( <i>Bs.</i> ) [32]	ViT-B/16	65.54	62.91	61.54	57.75	57.88	57.89	56.62	55.40	54.20	54.23	55.06	58.09	<b>10.48</b>
<b>LP-DiF (Ours)</b>	ViT-B/16	<b>83.94</b>	<b>80.59</b>	<b>79.17</b>	<b>74.30</b>	<b>73.89</b>	<b>73.44</b>	<b>71.60</b>	<b>70.81</b>	<b>69.08</b>	<b>68.74</b>	<b>68.53</b>	<b>74.00</b>	15.41
Joint-LP ( <i>UB. of ours</i> )	ViT-B/16	83.94	80.83	79.43	77.06	76.35	74.89	73.66	72.79	71.84	72.06	71.88	75.88	12.06

TABLE IV  
**COMPARISON BETWEEN OUR LP-DiF AND OTHER REPLAY-BASED FSCIL SOTA METHODS ON mini-IMAGENET.** “**EXEMPLAR / CLS**” REPRESENTS THE NUMBER OF EXEMPLAR OF EACH CLASS. NOTE THAT THESE EXISTING FSCIL SOTA METHODS USE THE UNITS OF IMAGE AS EXEMPLAR. “**DISK SPACE / CLS**” REPRESENTS THE DISK SPACE CONSUMED BY THE EXEMPLAR OF EACH CLASS.

<b>Methods</b>	<b>Exemplar / cls</b>	<b>Disk Space / cls</b>	<b>Avg</b>
Replay [30]	1 image	51.19 KB	58.02
F2M [36]	5 images	255.95 KB	57.89
BDF [60]	1 image	51.19 KB	61.42
<b>LP-DiF</b>	<b>2 vectors</b>	<b>1.15 KB</b>	<b>93.76</b>

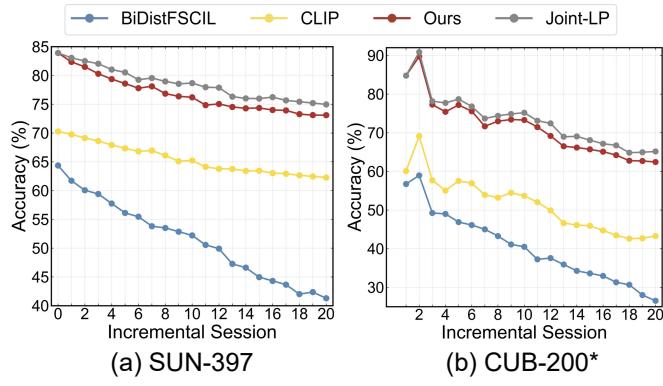


Fig. 4. **Accuracy curves** of our LP-DiF and comparison with counterparts on (a) SUN-397 and (b) CUB200\*. our LP-DiF method significantly surpasses both CLIP and BiDistFSCIL, and attains performance levels that are very close to the respective upper bounds.

of Avg, and shows comparable PD performance, *i.e.*, 4.66% vs. 4.56%. As for the existing SOTA methods, *e.g.*, NC-FSCIL [54], which presents the best Avg among all the SOTA methods, LP-DiF gains **25.94%** improvements, *i.e.*, 67.82% →

**93.76%**. Comparing with C-FSCIL [21], which presents the best PD. among the competing methods, LP-DiF gains **10.33%** improvements, *i.e.*, 14.99% → **4.66%**. Tab. IV highlights the comparison with the replay-based FSCIL method. Note that existing replay-based methods directly store old-class images, while our method only requires storing one mean vector and one variance vector (diagonal elements of the covariance matrix) for each old class. Generally, LP-DiF significantly outperforms others in terms of performance while costing the least amount of storage space. Tab. III and Tab. V shows the comparison results on CUB-200 and CIFAR-100. Overall, the performance of our LP-DiF can be summarized in two points. **1)** There are significant improvements compared to CLIP (*baseline*) in terms of Avg (*i.e.*, **15.91%** and **4.26%** improvements on CUB-200 and CIFAR-100 respectively). **2)** Compared to existing SOTA methods, LP-DiF achieves a higher Avg and lower PD. Moreover, considering that our method use CLIP as backbone, which is more stronger than those existing FSCIL methods whose backbone are ResNet, we replaced the backbone of these methods with CLIP to make a fairer comparison. Specifically, we replace the original backbone of these methods with CLIP’s image encoder (ViT-B/16), and we use the text encoding generated by CLIP’s text encoder for each category as initialization for the classification layer parameters of these methods. Tab. VII shows that the performance of the existing SOTA FSCIL methods combined with CLIP are still lower than our LP-DiF. These above results clearly illustrate the superiority of our LP-DiF.

**Comparison with Upper Bound.** Assuming that the training set from each previous session is available, we can jointly train the prompts using these sets, thereby avoiding the issue of forgetting old information. In class-incremental learning, the above-mentioned setting can be considered as an upper bound, and serve as a reference for evaluating FSCIL method. Thus,

TABLE V

**COMPARISON WITH STATE-OF-THE-ART FSCIL METHODS ON CIFAR-100.** “**BACKBONE.**” REPRESENTS THE BACKBONE OF VISUAL MODEL. “**AVG**” REPRESENTS THE AVERAGE ACCURACY OF ALL SESSIONS; THE HIGHER THE VALUE, THE BETTER PERFORMANCE. “**PD**” REPRESENTS THE PERFORMANCE DROP RATE; THE LOWER THE VALUE, THE BETTER PERFORMANCE. “**BS**” IS THE ABBREVIATION OF “**BASELINE**”. “**UB.**” IS THE ABBREVIATION OF “**UPPER BOUND**”.

Methods	Backbone	Accuracy in each session (%) ↑								Avg ↑	PD ↓
		0	1	2	3	4	5	6	7		
TOPIC [40]	Res18	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62
F2M [36]	Res18	64.71	62.05	59.01	55.58	52.55	49.96	48.08	46.28	44.67	53.65
CEC [56]	Res18	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53
Replay [30]	Res18	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.78
MetaFSCIL [12]	Res18	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79
GKEAL [66]	Res18	74.01	70.45	67.01	63.08	60.01	57.30	55.50	53.39	51.40	61.35
C-FSCIL [21]	Res12	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47	61.64
FCIL [18]	Res18	77.12	72.42	68.31	64.47	61.18	58.17	56.06	54.19	52.02	62.66
FACT [61]	Res18	78.22	72.40	68.57	64.73	61.40	58.57	56.30	53.83	51.72	62.86
SAVC [39]	Res18	78.77	73.31	69.31	64.93	61.70	59.25	57.13	55.19	53.12	63.63
BiDistFSCIL [60]	Res18	79.45	75.38	71.84	67.95	64.96	61.95	60.16	57.67	55.88	66.14
NC-FSCIL [54]	Res18	<b>82.52</b>	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50
CLIP ( <i>BS.</i> ) [32]	ViT-B/16	74.44	72.96	72.21	70.49	70.18	70.00	69.81	69.23	68.37	70.86
<b>LP-DiF (<i>Ours</i>)</b>	ViT-B/16	80.23	<b>77.75</b>	<b>76.78</b>	<b>74.62</b>	<b>74.03</b>	<b>73.87</b>	<b>73.84</b>	<b>72.96</b>	<b>72.02</b>	<b>75.12</b>
Joint-LP ( <i>UB. of ours</i> )	ViT-B/16	80.23	79.85	78.63	76.13	75.31	74.67	74.24	73.58	73.35	76.22
											6.88

TABLE VI

**COMPARISON WITH STANDARD CIL METHODS BASED ON PRE-TRAINED MODELS ON THE THREE COMMON BENCHMARKS IN TERMS OF AVG.** <sup>‡</sup> INDICATES OUR REPRODUCTION ON FSCIL PROTOCOL.

Methods	CIFAR-100	mini-ImageNet	CUB-200
L2P <sup>‡</sup> [49]	61.77	75.68	56.95
DualPrompt <sup>‡</sup> [48]	63.50	76.61	62.32
AttriCLIP <sup>‡</sup> [45]	59.24	81.74	47.81
<b>LP-DiF (<i>Ours</i>)</b>	<b>75.12</b>	<b>93.76</b>	<b>74.00</b>

TABLE VII

**COMPARISON WITH SOTA FSCIL METHODS COMBINED WITH CLIP ON THREE COMMON BENCHMARKS IN TERMS OF AVG.** WE REPLACE THE ORIGINAL BACKBONE OF THESE METHODS WITH CLIP’S IMAGE ENCODER (ViT-B/16), AND WE USE THE TEXT ENCODING GENERATED BY CLIP’S TEXT ENCODER FOR EACH CATEGORY AS INITIALIZATION FOR THE CLASSIFICATION LAYER PARAMETERS OF THESE METHODS.

Methods	CIFAR-100	mini-ImageNet	CUB-200
SAVC [39] + CLIP	68.46	86.81	71.66
BiDistFSCIL [60] + CLIP	69.40	84.63	70.95
<b>LP-DiF (<i>Ours</i>)</b>	<b>75.12</b>	<b>93.76</b>	<b>74.00</b>

we compare our LP-DiF with its upper bound (*i.e.* **Joint-LP**). As shown in the last row of Tab. I, Tab. III and Tab. V across the three benchmarks, the performances of our method are very close to the upper bounds in terms of Avg, with the largest gap being only 1.05%, 1.42% and 1.10% on mini-ImageNet, CUB-200 and CIFAR-100, respectively. The results indicate that our LP-DiF is highly effective in preventing catastrophic forgetting. It is noted that NC-FSCIL achieves higher accuracy than both ours and Joint-LP. The architectures of NC-FSCIL, which is ResNet-based method, and Joint-LP, which is CLIP-

based method, are different; NC-FSCIL trains all layers of the model during the base session, whereas Joint-LP only train the prompt. Therefore, it is acceptable that NC-FSCIL outperforms Joint-LP in session 0.

#### Comparison with Pre-trained Models-based Standard CIL Methods.

To further demonstrate the superiority of our method, we compare it with several recent standard CIL [62] methods which also utilize pre-trained models: L2P [49], DualPrompt [48] and AttriCLIP [45]. The L2P and Dual-Prompt are based on a pretrained ViT and learn the visual prompts to solve CIL problems, while AttriCLIP builds on CLIP and training different text prompts to encode different knowledge. We reproduce these three approaches on CIFAR-100, CUB-200, and mini-ImageNet and evaluate them under FSCIL protocol respectively. As shown in Tab. VI, our LP-DiF outperforms these methods by a large margin in terms of Avg across all three benchmarks. We also find that these methods based on pre-trained models underperform BiDistFSCIL [60] on CIFAR-100 and CUB-200. This indicates that these methods are not advantageous for the FSCIL setting, further underscoring the effectiveness and significance of our method.

**More Challenging Benchmarks.** On the three widely used benchmarks, the performance of our LP-DiF closely approaches the upper bound. To further assess our LP-DiF, we provide two more challenging benchmarks: SUN-397 and CUB-200\*. For each challenging benchmark, we compare our LP-DiF with three distinct approaches: zero-shot evaluation using CLIP (baseline), Joint-LP (upper bound), and BiDistFSCIL [60] (SOTA open-source method). The corresponding performance curves are depicted in Fig. 4. Overall, on both SUN-397 and CUB-200\*, our LP-DiF method 1) significantly surpasses both CLIP and BiDistFSCIL, and 2) attains perfor-

TABLE VIII

**ABLATION STUDIES OF OUR LP-DIF ON mini-IMAGENET.** **LP** AND **OCD** DENOTE LEARNING PROMPTS AND OLD-CLASS DISTRIBUTION, RESPECTIVELY. **RF** AND **SF** DENOTE THE REAL FEATURES OF TRAINING IMAGES AND SYNTHESIZED FEATURES GENERATED BY VAE, RESPECTIVELY.

CLIP	LP	OCD		Accuracy in each session (%) ↑								Avg ↑	PD ↓	
		RF	SF	0	1	2	3	4	5	6	7	8		
✓				80.01	79.16	78.89	77.97	77.44	76.83	76.32	76.02	75.45	77.57	<b>4.56</b>
✓	✓			<b>96.34</b>	94.28	92.83	89.93	88.39	86.10	85.49	85.70	84.76	89.31	11.58
✓	✓	✓		<b>96.34</b>	<b>96.14</b>	94.01	94.27	93.23	93.07	91.34	91.17	90.76	93.37	5.46
✓	✓		✓	<b>96.34</b>	<b>96.14</b>	93.79	92.48	91.25	90.94	90.15	89.41	89.27	92.23	7.07
✓	✓	✓	✓	<b>96.34</b>	<b>96.14</b>	<b>94.62</b>	<b>94.37</b>	<b>94.06</b>	<b>93.44</b>	<b>92.21</b>	<b>92.29</b>	<b>91.68</b>	<b>93.76</b>	4.66

TABLE IX

**ABLATION STUDIES OF OUR LP-DiF ON CUB-200.** **LP** AND **OCD** DENOTE LEARNING PROMPTS AND OLD-CLASS DISTRIBUTION, RESPECTIVELY. **RF** AND **SF** DENOTE THE REAL FEATURES OF TRAINING IMAGES AND SYNTHESIZED FEATURES GENERATED BY VAE, RESPECTIVELY.

CLIP	LP	OCD		Accuracy in each session (%) ↑								Avg ↑	PD ↓			
		RF	SF	0	1	2	3	4	5	6	7	8				
✓				65.54	62.91	61.54	57.75	57.88	57.89	56.62	55.40	54.20	54.23	55.06	58.09	<b>10.48</b>
✓	✓			<b>83.94</b>	78.32	75.10	70.62	70.75	68.09	65.69	64.55	62.47	61.94	61.96	70.71	21.98
✓	✓	✓		<b>83.94</b>	<b>80.59</b>	78.83	73.66	73.24	72.54	70.57	69.72	68.88	67.86	67.90	73.43	16.04
✓	✓		✓	<b>83.94</b>	<b>80.59</b>	78.41	72.65	72.76	71.25	69.86	67.99	67.20	66.73	66.88	72.56	17.06
✓	✓	✓	✓	<b>83.94</b>	<b>80.59</b>	<b>79.17</b>	<b>74.30</b>	<b>73.89</b>	<b>73.44</b>	<b>71.60</b>	<b>70.81</b>	<b>69.08</b>	<b>68.74</b>	<b>68.53</b>	<b>74.00</b>	15.41

TABLE X

**ABLATION STUDIES OF OUR LP-DiF ON CIFAR-100.** **LP** AND **OCD** DENOTE LEARNING PROMPTS AND OLD-CLASS DISTRIBUTION, RESPECTIVELY. **RF** AND **SF** DENOTE THE REAL FEATURES OF TRAINING IMAGES AND SYNTHESIZED FEATURES GENERATED BY VAE, RESPECTIVELY.

CLIP	LP	OCD		Accuracy in each session (%) ↑								Avg ↑	PD ↓	
		RF	SF	0	1	2	3	4	5	6	7	8		
✓				74.44	72.96	72.21	70.49	70.18	70.00	69.81	69.23	68.37	70.86	<b>6.07</b>
✓	✓			<b>80.23</b>	75.81	75.03	71.65	71.67	70.94	70.48	70.01	69.54	72.81	10.69
✓	✓	✓		<b>80.23</b>	<b>77.75</b>	<b>76.84</b>	74.40	73.81	73.24	73.69	72.52	71.60	74.89	8.63
✓	✓		✓	<b>80.23</b>	<b>77.75</b>	75.63	73.75	73.09	72.36	72.31	71.84	70.76	74.19	9.47
✓	✓	✓	✓	<b>80.23</b>	<b>77.75</b>	76.78	<b>74.62</b>	<b>74.03</b>	<b>73.87</b>	<b>73.84</b>	<b>72.96</b>	<b>72.02</b>	<b>75.12</b>	8.21

mance levels that are very close to the respective upper bounds. The results show that our LP-DiF remains very effective on these challenging situations, including those with a larger number of classes and extended session lengths, e.g., 397 classes across 21 sessions in SUN-397, as well as in those without a base session, exemplified by CUB-200\*.

#### D. Ablation Studies and Analysis

**Analysis of Key Components.** Our proposed method involves prompt tuning on CLIP to adapt the knowledge from each incremental session. It also constructs feature-level distributions to preserve old knowledge, thereby achieving resistance to catastrophic forgetting. To investigate the effect of the key components in our method, i.e., CLIP, prompt learning (LP), the distribution estimated by real features (RF) of training images, and the synthesized features (SF), we summarized the performance of each component on three common FSCIL benchmark in Tab. VIII, Tab. IX and Tab. X. Take the results on mini-ImageNet as an example, as illustrated in Tab. VIII, employing the LP technique noticeably improves performance across each session, ultimately resulting in a superior 11.74%

performance, i.e., from 77.57% → 89.31% in terms of average performance (refer to the second row of Table VIII). However, solely implementing LP causes higher PD than CLIP, e.g., 4.56% → 11.58%, due to the forgetting of old knowledge during learning in new sessions. Additionally, as mentioned in Sec.III-B, the old-class distribution (OCD) is effective in tackling the forgetting problem. Note that the distribution of each incremental session is estimated by real features (RF) and the synthesized features (SF) generated by VAE. So we conducted separate evaluations to assess the effect of these two types of “features”. Concretely, using only RF to estimate the old-class distribution can improve the Avg by 4.11%, i.e., 89.31% → **93.42%**, and reduce the PD. by **6.12%**, i.e., 11.58% → **5.46%**, (see the third row of Table VIII). Using only SF for each incremental session can also improve Avg and reduce PD, however, its effectiveness is marginally inferior to using only RF (refer to the fourth row of Table VIII). Finally, using both types of “features” can further improve the performance, which surpasses the outcomes achieved by using either RF or SF alone (see the last row of Table VIII). Thus, although LP can enable the model to effectively capture the

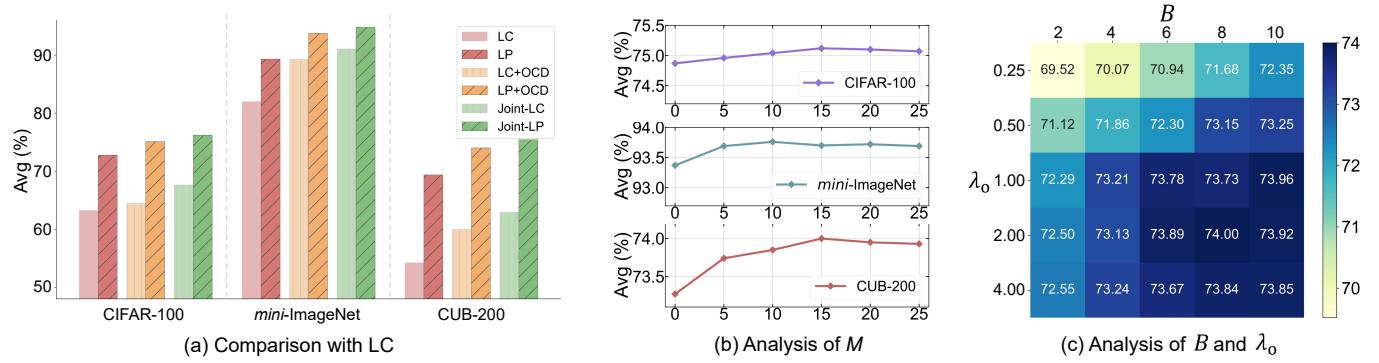


Fig. 5. **Ablation studies** of our LP-DiF. (a) Comparison with the method of incorporating a Linear Classifier (LC) into a pre-trained image encoder for training on three common benchmarks. (b) Analysis of  $M$  on three common benchmarks. (c) Analysis of  $B$  and  $\lambda_o$  in terms of Avg on **CUB-200**.

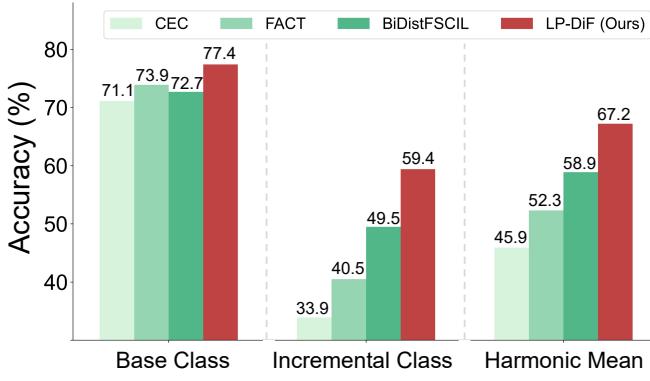


Fig. 6. **Decomposing** the performance of the **base class** and the **incremental class**. Their harmonic mean are also reported. The performance is evaluated by the model from the last session on **CUB-200**.

knowledge of each session and improve performance, it still leads to catastrophic forgetting, while OCD can effectively prevent this issue.

**Analysis of  $M$ .** As mentioned in Sec. III-B,  $M$  is the number of synthetic features which participate in estimating  $\mu_c$  and  $\sigma_c$  of old-class distribution. The value of  $M$  will affect the accuracy of the estimated distribution, thus influencing the performance of the model. Therefore, we test the effect of  $M$  on the performance. Fig. 5 (b) shows the results on the three widely used benchmarks. Clearly, when  $M$  increases from 0, the Avg gradually improves. Nonetheless, when  $M$  continues to increase, the Avg decreases slightly, possibly ascribing to that too many synthesized features can cause the estimated distribution to overly skew towards the distribution of the synthesized features. In summary,  $M = 10$  (on *mini-ImageNet*) or  $M = 15$  (on CIFAR-100 and CUB-200) are the best choices for performance.

**Analysis of  $B$  and  $\lambda_o$ .** Here we investigate the effect of the two hyper-parameters, *i.e.*,  $B$ , the number of selected old classes involved in  $\mathcal{L}_o$ , and  $\lambda_o$ , the tradeoff coefficient in  $\mathcal{L}_o$ . The results on **CUB-200** are shown with a mixed matrix of these two hyper-parameters in Fig 5 (c). Obviously, keeping  $\lambda_o$  fixed, as  $B$  increases, the Avg improves gradually. Keeping  $B$

fixed and tuning  $\lambda_o$  shows a similar tendency with the above setting. It achieves the best Avg when  $B = 8$  and  $\lambda_o = 2$  on CUB-200. Then, when the values of  $B$  and  $\lambda_o$  are too large, *e.g.*  $B = 10$  and  $\lambda_o = 4$ , there is a slight performance drop. These may be because too small values of  $B$  and  $\lambda_o$  lead to insufficient representation of old knowledge, while too large values may cause the model to overly emphasize old knowledge.

**Learning Prompt *vs.* Linear Classifier.** This study utilizes prompt tuning to tailor CLIP to the specific knowledge of each session. Another straightforward and intuitive strategy involves incorporating a linear classifier with the image encoder, which is initialized using the text encoding of handcrafted prompts. So we conduct additional experiments: **1)** Refining the linear classifier (LC) solely with the training set accessible in the current session; **2)** Extending the first approach by integrating the old-class distribution for feature replay (LC + OCD); **3)** Jointly training the linear classifier with the complete training set from each session (Joint-LC). As shown in Fig 5 (a), the Avg of LC is notably lower than that of LP across three wide benchmarks in terms of Avg. The incorporation of OCD with LC (denoted as LC + OCD) enhances performance beyond LC alone, highlighting OCD's effectiveness in mitigating catastrophic forgetting. Nevertheless, the combined LC + OCD is still inferior to LP + OCD. In a joint training scenario, the performance of Joint-LC continues to be inferior to Joint-LP. The results suggest that the strategy of learning prompts offers more merits for FSCIL than that of learning a linear classifier.

**Old-Class Distribution *vs.* Image Exemplar.** To further validate its efficacy in avoiding catastrophic forgetting, we compare our method with other replay-based approaches tailored for learning prompts, *i.e.*, **1)** randomly selecting  $N_e$  images of per old class as exemplars; **2)** adopting the replay strategy in iCaRL [33], specifically choosing  $N_e$  images for each old class based on the proximity to the mean feature. In addition, we execute the random selection approach five times, each with a different random seed, to reduce the uncertainty. The average results with necessary storage space for replay on CUB-200 are shown in Table XI, where CLIP + LP indicates learning prompts sequentially across each incremental session without replay of old classes. (*e.g.* the second row in Tab. IX).

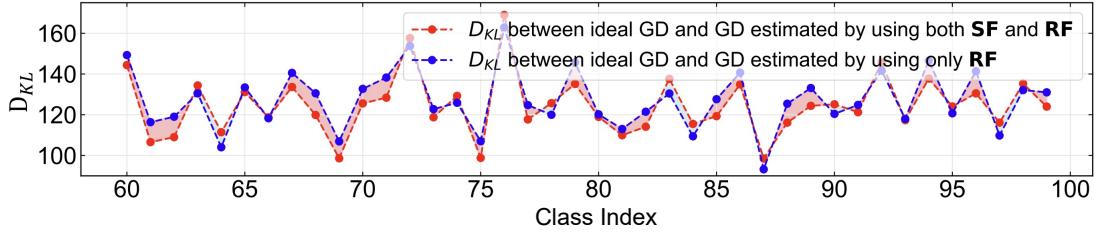


Fig. 7. Analysis the effectiveness of VAE on *mini*-ImageNet. We calculate the difference between the estimated GD of each incremental class and corresponding reference GD (ideal GD, *i.e.*, computed using the full training images of that class) by KL divergence. The red dots represents KL divergence between ideal GD and GD estimated by using both **SF** and **RF**. The blue dots represents KL divergence between ideal GD and GD estimated by using only **RF**.

TABLE XI

COMPARISON WITH OTHER *replay approaches* ON CUB-200 IN TERMS OF **AVG**. THE RESULTS OF “RANDOMLY SELECTION” ARE REPORTS OVER 5 RUNS WITH MEAN AND STANDARD DEVIATIONS. iCaRL<sup>†</sup> MEANS APPLYING THE REPLAY TECHNIQUE PROPOSED IN iCaRL TO CLIP + LP.

Methods	$N_e$	Disk Space	Avg
CLIP + LP	-	-	69.36
Randomly selection	1	$18.32 \pm 0.37$ MB	$69.95 \pm 0.56$
	2	$37.31 \pm 0.99$ MB	$71.16 \pm 0.24$
	3	$55.95 \pm 1.01$ MB	$72.44 \pm 0.09$
	4	$74.33 \pm 0.70$ MB	$73.64 \pm 0.16$
iCaRL <sup>†</sup> [33]	1	18.54 MB	70.81
	2	38.39 MB	71.68
	3	55.40 MB	72.86
	4	74.68 MB	73.95
<b>LP-DiF (Ours)</b>	-	<b>0.22 MB</b>	<b>74.00</b>

Obviously, our method exhibits the best performance and lowest storage space in comparison to the two counterparts under various  $N_e$ . Especially, compared with iCaRL<sup>†</sup> under  $N_e = 4$ , LP-DiF shows a comparable performance while only requiring about **0.002%** storage space (thanks to the fact that we only store two vectors for each old class). This underscores that our pseudo-feature replay technique can effectively combat catastrophic forgetting under conditions of light storage overhead.

**Decomposing the Performance of Base and Incremental Classes.** Following previous studies [56], [60], [61], in this section, we decompose the accuracy, respectively analyzing the effectiveness of our LP-DiF for the classes in the base session (*i.e.*, base class) and for the classes in incremental sessions (*i.e.*, incremental class), to evaluate if our method performs well on both base and incremental classes. We report the comparison results in terms of individual accuracy of base and novel classes, as well as their harmonic mean, in the last session on CUB-200. Fig. 6 shows that our LP-DiF outperforms the second best method on base class (*i.e.*, FACT) by 3.5%, while outperforms the second best method on incremental class (*i.e.*, BiDistFSCIL) by 9.9%. Finally, the superior harmonic mean demonstrates our achievement of an enhanced balance between base and novel classes.

**Analysis on Shot Numbers.** To further demonstrate the superiority of our approach, we conducted experiments under

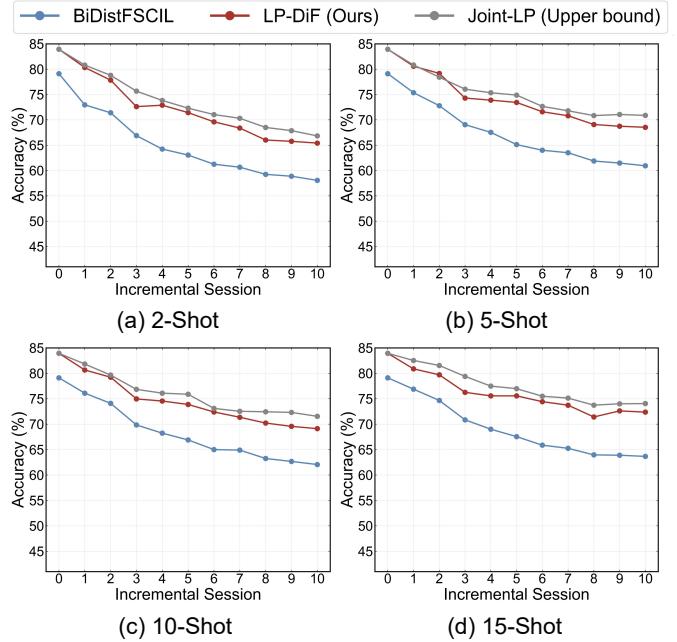


Fig. 8. Comparison with BiDistFSCIL (SOTA FSCIL method) and Joint-LP (Upper bound) under various **shot numbers** of incremental classes on CUB-200.

various shot numbers of incremental classes. Fig. 8 show the comparison results with BiDistFSCIL [60] and Joint-LP on CUB-200 under (a) 2-shot, (b) 5-shot, (c) 10-shot and (d) 15-shot. Obviously, across all the shot number settings, our LP-DiF consistently outperforms BiDistFSCIL significantly, and its performance is very close to the upper bound. This result demonstrates that, regardless of the shot numbers of incremental classes, our LP-DiF presents satisfactory performance and the ability to resist catastrophic forgetting.

**Analysis on Effect of VAE.** From the results presented in Tab. VIII, one can observe that using both synthesizing features (**SF**) and real features (**RF**) for estimating the Gaussian distribution (GD), as compared to using only real features, can achieve higher performance. To elucidate the quality of the features synthesized by the VAE, we conducted the following analysis on *mini*-ImageNet: we calculate the difference between the estimated GD of each incremental class and corresponding reference GD (ideal GD, *i.e.*, computed using

the full training images of that class) by *KL divergence*  $D_{\text{KL}}$

$$D_{\text{KL}}(P|Q) = \frac{1}{2} \sum_{i=1}^n \left( \frac{\sigma_{p_i}^2}{\sigma_{q_i}^2} + \frac{(\mu_{q_i} - \mu_{p_i})^2}{\sigma_{q_i}^2} - 1 + \ln \left( \frac{\sigma_{q_i}^2}{\sigma_{p_i}^2} \right) \right), \quad (12)$$

where  $P$  and  $Q$  represent the ideal GD and the estimated GD respectively, and  $n$  represents the dimension of GD. Statistically, lower  $D_{\text{KL}}$  indicates that the estimated GD is closer to reference GD. Fig 7 shows the results on *mini-ImageNet*. Note that for most classes, the GD estimated using both **SF** and **RF** is closer to the reference distribution, indicating that **SF** can enrich more class-relevant information.

**Training Time and Model Size.** Compared to existing ResNet-based FSCIL methods, our LP-DiF is based on the heavier model (*i.e.*, CLIP), which may raise concerns about model training efficiency and memory overhead. However, since LP-DiF only trains lightweight prompt vectors and a few layers of MLP in the VAE, it does not incur excessive computational costs. Here, we offer some quantitative results for reference: For the training time, with  $8 \times 2080ti$  GPUs, for each incremental session, training LP-DiF takes about 4.5 minutes (1.8 minutes for training VAE and 2.7 minutes for training prompts). In comparison, BiDistFSCIL [5] takes about 3.3 minutes for training for the same epochs. For the volume of trainable parameters, existed FSCIL methods relied on ResNet require training about 11.3M parameters for ResNet-18, respectively. However, LP-DiF only needs to train about 7.4M parameters for prompts and MLPs. Thus, our LP-DiF achieved significant performance gain with acceptable addition on training time and lower volume of trainable parameters.

## V. CONCLUSION

In this paper, we studied the FSCIL problem by introducing V-L pretrained model, and proposed **L**earning **P**rompt with **D**istribution-based **F**eature replay (LP-DiF). Specifically, prompt tuning is involved to adaptively capture the knowledge of each session. To alleviate catastrophic forgetting, we established a feature-level distribution for each class, which is estimated by both real features of training images and synthesized features generated by a VAE decoder. Then, pseudo features are sampled from old-class distributions, and combined with the training set of current session to train the prompts jointly. Extensive experiments show that our LP-DiF achieves the new state-of-the-art in the FSCIL task.

## REFERENCES

- [1] Aishwarya Agarwal, Biplab Banerjee, Fabio Cuzzolin, and Subhasis Chaudhuri. Semantics-driven generative replay for few-shot class incremental learning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5246–5254, 2022.
- [2] Touqueer Ahmad, Akshay Raj Dhamija, Steve Cruz, Ryan Rabinowitz, Chunchun Li, Mohsen Jafarzadeh, and Terrance E Boult. Few-shot class incremental learning leveraging self-supervised features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3900–3910, 2022.
- [3] Touqueer Ahmad, Akshay Raj Dhamija, Mohsen Jafarzadeh, Steve Cruz, Ryan Rabinowitz, Chunchun Li, and Terrance E Boult. Variable few shot class incremental and open world learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3688–3699, 2022.
- [4] 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.
- [5] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.
- [6] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with memory of diverse samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8218–8227, 2021.
- [7] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European conference on computer vision (ECCV)*, pages 532–547, 2018.
- [8] Arslan Chaudhry, Albert Gordo, Puneet Dokania, Philip Torr, and David Lopez-Paz. Using hindsight to anchor past knowledge in continual learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6993–7001, 2021.
- [9] Jimpeng Chen, Runmin Cong, Yuxuan Luo, Horace Ip, and Sam Kwong. Saving 100x storage: Prototype replay for reconstructing training sample distribution in class-incremental semantic segmentation. *Advances in Neural Information Processing Systems*, 36, 2024.
- [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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2534–2543, 2021.
- [11] Ali Cheraghian, Shafin Rahman, Sameera Ramasinghe, Pengfei Fang, Christian Simon, Lars Petersson, and Mehrtash Harandi. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8661–8670, 2021.
- [12] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metascil: A meta-learning approach for few-shot class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14166–14175, 2022.
- [13] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7):3366–3385, 2021.
- [14] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1255–1263, 2021.
- [15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.
- [16] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9285–9295, 2022.
- [17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [18] Ziqi Gu, Chunyan Xu, Jian Yang, and Zhen Cui. Few-shot continual infomax learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19224–19233, 2023.
- [19] Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *BMVC*, page 98, 2018.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [21] Michael Hersche, Geethan Karunaratne, Giovanni Cherubini, Luca Benini, Abu Sebastian, and Abbas Rahimi. Constrained few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9057–9067, 2022.
- [22] Wpeng Hu, Zhou Lin, Bing Liu, Chongyang Tao, Zhengwei Tao, Jinwen Ma, Dongyan Zhao, and Rui Yan. Overcoming catastrophic forgetting for continual learning via model adaptation. In *International conference on learning representations*, 2018.
- [23] David Isele and Akansel Cosgun. Selective experience replay for lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

- [24] Zhong Ji, Zhishen Hou, Xiyao Liu, Yanwei Pang, and Xuelong Li. Memorizing complementation network for few-shot class-incremental learning. *IEEE Transactions on Image Processing*, 32:937–948, 2023.
- [25] Jian Jiang, Edoardo Cetin, and Oya Celiktutan. Ib-drr-incremental learning with information-back discrete representation replay. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3533–3542, 2021.
- [26] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *The Eleventh International Conference on Learning Representations*, 2022.
- [27] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [28] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [29] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9020–9029, 2021.
- [30] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022.
- [31] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 2337–2345, 2021.
- [32] 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*, pages 8748–8763. PMLR, 2021.
- [33] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.
- [34] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [35] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.
- [36] 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. *Advances in neural information processing systems*, 34:6747–6761, 2021.
- [37] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.
- [38] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.
- [39] Zeyin Song, Yifan Zhao, Yujun Shi, Peixi Peng, Li Yuan, and Yonghong Tian. 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*, pages 24183–24192, 2023.
- [40] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192, 2020.
- [41] Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.
- [42] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. A survey on few-shot class-incremental learning. *arXiv preprint arXiv:2304.08130*, 2023.
- [43] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.
- [44] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *arXiv preprint arXiv:2302.00487*, 2023.
- [45] Runqi Wang, Xiaoyue Duan, Guoliang Kang, Jianzhuang Liu, Shaohui Lin, Songcen Xu, Jinhua Lü, and Baochang Zhang. Attriclip: A non-incremental learner for incremental knowledge learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3654–3663, 2023.
- [46] Yabin Wang, Zhiwu Huang, and Xiaopeng Hong. S-prompts learning with pre-trained transformers: An occam’s razor for domain incremental learning. *Advances in Neural Information Processing Systems*, 35:5682–5695, 2022.
- [47] Zhengbo Wang, Jian Liang, Ran He, Nan Xu, Zilei Wang, and Tieniu Tan. Improving zero-shot generalization for clip with synthesized prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3032–3042, 2023.
- [48] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.
- [49] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.
- [50] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.
- [51] 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.
- [52] 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 Transactions on Pattern Analysis and Machine Intelligence*, 45(3):2945–2951, 2022.
- [53] Yang Yang, Zhiying Cui, Junjie Xu, Changhong Zhong, Wei-Shi Zheng, and Ruixuan Wang. Continual learning with bayesian model based on a fixed pre-trained feature extractor. *Visual Intelligence*, 1(1):5, 2023.
- [54] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning. In *The Eleventh International Conference on Learning Representations*, 2022.
- [55] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. *arXiv preprint arXiv:2302.03004*, 2023.
- [56] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464, 2021.
- [57] Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19148–19158, 2023.
- [58] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, XIAOPENG ZHANG, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. In *The Twelfth International Conference on Learning Representations*.
- [59] Hanbin Zhao, Yongjian Fu, Mintong Kang, Qi Tian, Fei Wu, and Xi Li. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.
- [60] Linglan Zhao, Jing Lu, Yunlu Xu, Zhanzhan Cheng, Dashan Guo, Yi Niu, and Xiangzhong Fang. Few-shot class-incremental learning via class-aware bilateral distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11838–11847, 2023.
- [61] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9046–9056, 2022.
- [62] Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Deep class-incremental learning: A survey. *arXiv preprint arXiv:2302.03648*, 2023.
- [63] Da-Wei Zhou, Han-Jia Ye, Liang Ma, Di Xie, Shiliang Pu, and De-Chuan Zhan. Few-shot class-incremental learning by sampling multi-phase tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [64] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.
- [65] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning.

- In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6801–6810, 2021.
- [66] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. 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*, pages 7746–7755, 2023.
- [67] Yixiong Zou, Shanghang Zhang, Yuhua Li, and Ruixuan Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. *Advances in neural information processing systems*, 35:27267–27279, 2022.



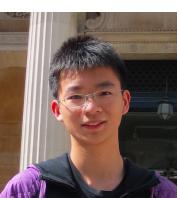
**Zitong Huang** is currently pursuing the Ph.D. degree with Harbin Institute of Technology, Harbin, China. His research interests include computer vision, deep learning, continual learning and object detection.



**Ze Chen** is currently employed at Megvii Technology Limited, having graduated with a master's degree from Shanghai Jiao Tong University. His research interests encompass general object detection, the practical applications of deep learning models, and generative modeling.



**Zhixing Chen** is currently pursuing a Bachelor's degree in Robotics Engineering at Harbin Institute of Technology, China. His research interests include computer vision, artificial intelligence, and robotics.



**Erjin Zhou** is the Director of the research group at Megvii Research Institute, responsible for building and leading the team to conduct research on facial and human body detection and recognition, portrait generation, general object detection, and action recognition technologies. Erjin Zhou also leads the team in the research and development of algorithm production tools. His research achievements have been applied to Megvii's cloud-based identity verification solution, as well as industry solutions for intelligent building access, smart phone AI solutions, and financial industry identity verification.



**Xinxing Xu** received his Ph.D. degrees from the Nanyang Technological University (NTU). He currently is a scientist with Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A\*STAR), Singapore. His research interests include computer vision, deep learning, and digital healthcare.



**Rick Siow Mong Goh** received his Ph.D. degree in electrical and computer engineering from the National University of Singapore in 2006. He is currently the Director of the Computing and Intelligence (CI) Department at A\*STAR's Institute of High Performance Computing (IHPC). He leads a team of over 80 scientists in performing world-leading scientific research, developing technology to commercialisation, and engaging and collaborating with industry. His research interests include artificial intelligence (AI), high performance computing, blockchain and federated learning.



**Yong Liu** is the Deputy Department Director of Computing & Intelligence Department at Institute of High Performance Computing (IHPC), A\*STAR, Singapore. He is also Adjunct Associate Professor at Duke-NUS Medical School, NUS and Adjunct Principal Investigator at Singapore Eye Research Institute (SERI). His research interests include computer vision and deep learning.



**Wangmeng Zuo** received the Ph.D. degree from the Harbin Institute of Technology in 2007. He is currently a Professor in the School of Computer Science and Technology, Harbin Institute of Technology. His research interests include image enhancement and restoration, image and face editing, object detection, visual tracking, and image classification. He has published over 100 papers in top tier journals and conferences. His publications have been cited more than 55,000 times. He also serves as Associate Editors for IEEE T-PAMI and IEEE T-IP.



**Chun-Mei Feng** received the Ph.D. degree from Harbin Institute of Technology, Shenzhen. She is currently a research scientist in Institute of High Performance Computing (IHPC), Agency for Science, Technology and Research (A\*STAR), Singapore. Her research interests include Federated learning, Medical image analysis, and Multi-modal foundation learning.