A Experimental Details

A.1 Dataset Details

We follow previous FSCIL works [14, 18] to use CIFAR100 [7] and CUB200-2011 [12] datasets. As the ViT backbone is pre-trained on ImageNet, we follow previous prompt-based CIL works [15, 16] to use ImageNet-R [5] as another benchmark dataset. CIFAR100 contains 100 classes with 500 training images and 100 testing images per class. The CUB200-2011 is a fine-grained dataset with 200 classes of different birds. It contains 5,994 samples for training and 5,794 samples for testing. The ImageNet-R contains 16 different renditions of 200 ImageNet classes and the total number of images is 30, 000. We follow previous works [14, 18] to split the classes as base and new classes. For the new classes under the k-shot setting, k images are randomly chosen from each class for training.

A.2 Baseline Details

For the classical CIL methods iCaRL [10] and Foster [13], we follow the original implementation and equip iCaRL with a rehearsal buffer of size 200. For FSCIL baselines CEC, FACT and TEEN, we replace the original ResNet backbone with the same ViT-B/16-1K [3] we used for `ASP`. The other operations are kept the same as the original implementations. As we find that the original implementation of prompt-based approaches performs badly under the FSCIL setting, we replace the original classifier head with the class mean to construct the Prototypical Network, and denote them as L2P+, DualP+ and CodaP+. For a fair comparison, all methods are tuned 20 epochs on base classes.

A.3 Results for Main Paper

In the main paper, we only provide the Top-1 accuracy of CUB200 in Figure 1, here we provide the detailed comparison between baselines and `ASP` on CUB200, which is shown in Tab. 1. Like CIFAR100 and ImageNet-R, `ASP` achieves the best on A_{avg}, PD and HAcc, surpassing the second best by 0.7%, 3.2% and 3.2% respectively. In addition, `ASP` also achieves the best A_{10} of 83.5% after the last task, outperforms the second best by 2.9%.

A.4 Architecture of Prompt Encoder

We use two small fully connected neural networks with one hidden layer as f_μ and f_Σ, and the hidden unit is set as 256. The parameter of the whole model is around 175M as we load and pass through the ViT backbone twice. We count the backbone twice in case using a different backbone for the prompt encoder. In contrast, the total parameter amount of two fully connected neural networks is around 2M, which is only 1% of the whole model. Thus, `ASP` is parameter efficient.

Table 1: Detailed Top-1 accuracy A_t in each incremental task, average accuracy A_{avg}, performance dropping rate (PD) and Harmonic Accuracy (HAcc) on CUB200 dataset. ↑ means higher is better, and ↓ means lower is better.

Method	Accuracy A_t in each task (%) ↑											A_{avg} ↑	PD↓	HAcc↑
	0	1	2	3	4	5	6	7	8	9	10			
iCaRL	92.4	82.3	71.7	62.9	64.6	62.9	60.8	60.3	58.4	55.7	58.9	66.4	33.5	56.4
Foster	**93.0**	86.4	80.4	74.5	72.9	69.3	68.5	66.0	65.9	65.0	63.4	73.2	29.6	58.8
CEC	84.8	82.5	81.4	78.5	79.3	77.8	77.4	77.6	77.2	76.9	76.8	79.1	7.9	76.2
FACT	87.3	84.2	82.1	78.1	78.4	76.3	75.4	75.5	74.4	74.1	73.9	78.2	13.4	72.0
TEEN	89.0	**86.5**	**85.9**	83.3	83.3	82.2	82.1	80.4	80.5	80.1	80.6	83.1	9.4	80.2
L2P	87.8	81.3	74.2	68.9	63.4	59.2	55.8	51.9	49.0	46.3	44.3	62.0	43.5	3.3
DualP	88.9	82.6	75.3	69.6	64.5	60.4	56.8	53.0	50.0	47.5	45.5	63.1	43.4	5.0
CodaP	89.9	83.1	75.8	70.2	65.1	61.0	57.5	53.7	50.7	48.3	46.2	63.8	43.7	5.7
L2P+	82.4	81.2	79.0	76.8	76.2	74.7	74.1	74.1	72.7	73.0	73.6	76.2	8.7	73.0
DualP+	83.5	82.2	80.9	79.5	78.6	77.0	76.3	77.0	75.7	76.1	76.5	78.5	7.1	76.3
CodaP+	79.6	78.1	76.4	75.6	75.0	73.1	72.5	72.8	72.0	72.4	72.9	74.6	6.8	72.5
Ours	87.1	86.0	84.9	**83.4**	**83.6**	**82.4**	**82.6**	**83.0**	**82.6**	**83.0**	**83.5**	**83.8**	**3.6**	**83.4**

Table 2: Detailed Top-1 accuracy A_t in each incremental task, average accuracy A_{avg} and performance dropping rate (PD) on CIFAR100 dataset. ↑ means higher is better, and ↓ means lower is better.

Method	Accuracy A_t in each task (%) ↑									A_{avg} ↑	PD↓
	0	1	2	3	4	5	6	7	8		
CPE-CLIP	87.8	85.9	84.9	82.9	82.6	82.4	82.3	81.4	80.5	83.4	7.3
LP-DiF	80.2	77.8	76.8	74.6	74.0	73.9	73.8	73.0	72.0	75.1	8.2
Ours	**92.2**	**90.7**	**90.0**	**88.7**	**88.7**	**88.2**	**88.2**	**87.8**	**86.7**	**89.0**	**5.5**

A.5 Comparsion with Multi-modality Approach

Instead of using ResNet [4] as the backbone in classical FSCIL methods or using ViT [3] as the backbone in prompt-based CIL methods and `ASP`, some recent works utilize large multi-modality models to address the FSCIL scenario. CPE-CLIP [2], IOS [17] and LP-DiF [6] use OpenAI CLIP [9] as the backbone. Their CLIP backbone uses the ViT-Base model as the image encoder, which is the same size as our experimental setting. Thus, the model backbones have comparable capability, and we compare their performance with `ASP` under the same FSCIL setting. As for now, we only have the detailed results reported by the original paper CPE-CLIP and LP-DiF, we compare them with `ASP` on the CIFAR100 and CUB200 datasets and show the results in Tabs. 2 and 3. `ASP` significantly outperforms CPE-CLIP on all metrics.

A.6 t-SNE Visualization of Prompts

We use t-SNE [8] to visualize the TIP and TSP in the first layer and show it in Fig 1. We can observe that TSP doesn't change a lot between different

Table 3: Detailed Top-1 accuracy A_t in each incremental task, average accuracy A_{avg} and performance dropping rate (PD) on CUB200 dataset. ↑ means higher is better, and ↓ means lower is better.

Method	Accuracy A_t in each task (%) ↑											A_{avg} ↑	PD↓
	0	1	2	3	4	5	6	7	8	9	10		
CPE-CLIP	81.6	78.5	76.7	71.9	71.5	70.2	67.7	66.5	65.1	64.5	64.6	70.8	17.0
LP-DiF	83.9	80.6	79.2	74.3	73.9	73.4	71.6	70.8	69.1	68.7	68.5	74.0	15.4
Ours	**87.1**	**86.0**	**84.9**	**83.4**	**83.6**	**82.4**	**82.6**	**83.0**	**82.6**	**83.0**	**83.5**	**83.8**	**3.6**

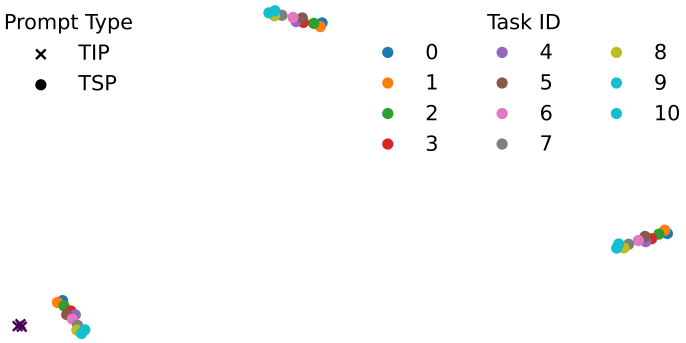

Fig. 1: tSNE visualization of TIP and TSP from the final model. Each point represents a prompt token.

tasks. Thus, it can provide robust and generalized features for new classes. On the other hand, TIP remains the same among different tasks to provide task-invariant information.

B Detailed Derivation of IB Loss

The approach of *Information Bottleneck* was first proposed by work [11], where the goal is to learn an encoding feature P that is maximally expressive about the label Y while being maximally compressive about the input X:

$$\mathcal{L}_{IB} = I(\mathcal{P}; \mathcal{X}) - \gamma I(\mathcal{P}; \mathcal{Y}) \tag{1}$$

In general, it is computationally challenging to calculate the mutual information $I(\cdot)$. We follow the Variational Information Bottleneck method [1] to construct a lower bound on the IB objective in Eq. (1).

Following standard practice in the IB literature, we assume that the joint distribution $P(\mathcal{X}, \mathcal{Y}, \mathcal{P})$ factors as follows:

$$P(\mathcal{X}, \mathcal{Y}, \mathcal{P}) = P(\mathcal{P}|\mathcal{X}, \mathcal{Y})P(\mathcal{Y}|\mathcal{X})P(\mathcal{X}) = P(\mathcal{P}|\mathcal{X})P(\mathcal{Y}|\mathcal{X})P(\mathcal{X}) \tag{2}$$

where we assume $P(\mathcal{P}|\mathcal{X},\mathcal{Y}) = P(\mathcal{P}|\mathcal{X})$ based on the Markov chain. We then write $I(\mathcal{P};\mathcal{Y})$ out in full:

$$I(\mathcal{P},\mathcal{Y}) = \int d\boldsymbol{y}\, d\boldsymbol{p}\, P(\boldsymbol{y},\boldsymbol{p}) \log \frac{P(\boldsymbol{y},\boldsymbol{p})}{P(\boldsymbol{y})P(\boldsymbol{p})} = \int d\boldsymbol{y}\, d\boldsymbol{z}\, P(\boldsymbol{y},\boldsymbol{p}) \log \frac{P(\boldsymbol{y}|\boldsymbol{p})}{P(\boldsymbol{y})} \quad (3)$$

where $P(\boldsymbol{y}|\boldsymbol{p})$ is defined as follows:

$$P(\boldsymbol{y}|\boldsymbol{p}) = \int d\boldsymbol{x}\, P(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{p}) = \int d\boldsymbol{x}\, P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{x}|\boldsymbol{p}) = \int d\boldsymbol{x}\, \frac{P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x})P(\boldsymbol{x})}{P(\boldsymbol{p})} \quad (4)$$

Let $q(\boldsymbol{y}|\boldsymbol{p})$ be a variational approximation to $P(\boldsymbol{y}|\boldsymbol{p})$. As the Kullback Leibler divergence is always positive, we have

$$\mathbb{KL}[P(\mathcal{Y}|\mathcal{P}), q(\mathcal{Y}|\mathcal{P})] \geq 0 \implies \int d\boldsymbol{y}\, P(\boldsymbol{y}|\boldsymbol{p}) \log P(\boldsymbol{y}|\boldsymbol{p}) \geq \int d\boldsymbol{y}\, P(\boldsymbol{y}|\boldsymbol{p}) \log q(\boldsymbol{y}|\boldsymbol{p}) \quad (5)$$

and hence

$$I(\mathcal{P},\mathcal{Y}) \geq \int d\boldsymbol{y}\, d\boldsymbol{z}\, P(\boldsymbol{y},\boldsymbol{p}) \log \frac{q(\boldsymbol{y}|\boldsymbol{p})}{P(\boldsymbol{y})} \quad (6)$$

$$= \int d\boldsymbol{y}\, d\boldsymbol{p}\, P(\boldsymbol{y},\boldsymbol{p}) \log q(\boldsymbol{y}|\boldsymbol{p}) - \int d\boldsymbol{y}\, P(\boldsymbol{y}) \log P(\boldsymbol{y}) \quad (7)$$

$$= \int d\boldsymbol{y}\, d\boldsymbol{p}\, P(\boldsymbol{y},\boldsymbol{p}) \log q(\boldsymbol{y}|\boldsymbol{p}) + H(\mathcal{Y}) \quad (8)$$

The entropy of labels $H(\mathcal{Y})$ is independent of our optimization procedure, and thus can be ignored.

Leveraging our Markov assumption, we can rewrite $P(\boldsymbol{y},\boldsymbol{p}) = \int d\boldsymbol{x}\, P(\boldsymbol{x},\boldsymbol{y},\boldsymbol{p}) = \int d\boldsymbol{x}\, P(\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x})$ in 8, which gives us a new lower bound on the first term of our objective:

$$I(Z,Y) \geq \int d\boldsymbol{x}\, d\boldsymbol{y}\, d\boldsymbol{z}\, P(\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x}) \log q(\boldsymbol{y}|\boldsymbol{p})\,. \quad (9)$$

We now consider the term $\gamma I(\mathcal{P},\mathcal{X})$:

$$I(\mathcal{P},\mathcal{X}) = \int d\boldsymbol{p}\, d\boldsymbol{x}\, P(\boldsymbol{x},\boldsymbol{p}) \log \frac{P(\boldsymbol{p}|\boldsymbol{x})}{P(\boldsymbol{p})} \quad (10)$$

$$= \int d\boldsymbol{p}\, d\boldsymbol{x}\, P(\boldsymbol{x},\boldsymbol{p}) \log P(\boldsymbol{p}|\boldsymbol{x}) - \int d\boldsymbol{p}\, P(\boldsymbol{p}) \log P(\boldsymbol{p})\,. \quad (11)$$

In general, computing the marginal distribution of $\mathcal{P}$, $P(\boldsymbol{p}) = \int d\boldsymbol{x}\, P(\boldsymbol{p}|\boldsymbol{x})P(\boldsymbol{x})$, might be difficult. We let $r(\boldsymbol{p})$ be a variational approximation to this marginal. Since $\mathbb{KL}[P(\mathcal{P}), r(\mathcal{P})] \geq 0 \implies \int d\boldsymbol{p}\, P(\boldsymbol{p}) \log P(\boldsymbol{p}) \geq \int d\boldsymbol{p}\, P(\boldsymbol{p}) \log r(\boldsymbol{p})$, we have the following upper bound:

$$I(\mathcal{P},\mathcal{X}) \leq \int d\boldsymbol{x}\, d\boldsymbol{p}\, P(\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x}) \log \frac{P(\boldsymbol{p}|\boldsymbol{x})}{r(\boldsymbol{p})} \quad (12)$$

Combining both of these bounds we have that

$$I(\mathcal{P},\mathcal{Y}) - \gamma I(\mathcal{P},\mathcal{X}) \geq \int d\boldsymbol{x}\, d\boldsymbol{y}\, d\boldsymbol{p}\, P(\boldsymbol{x})P(\boldsymbol{y}|\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x}) \log q(\boldsymbol{y}|\boldsymbol{p}) \\ - \gamma \int d\boldsymbol{x}\, d\boldsymbol{p}\, P(\boldsymbol{x})P(\boldsymbol{p}|\boldsymbol{x}) \log \frac{P(\boldsymbol{p}|\boldsymbol{x})}{r(\boldsymbol{p})} \tag{13}$$

References

1. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. arXiv preprint arXiv:1612.00410 (2016)
2. D'Alessandro, M., Alonso, A., Calabrés, E., Galar, M.: Multimodal parameter-efficient few-shot class incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 3393–3403 (October 2023)
3. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
5. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8340–8349 (2021)
6. Huang, Z., Chen, Z., Chen, Z., Zhou, E., Xu, X., Goh, R.S.M., Liu, Y., Feng, C., Zuo, W.: Learning prompt with distribution-based feature replay for few-shot class-incremental learning. arXiv preprint arXiv:2401.01598 (2024)
7. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
8. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
9. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
10. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
11. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
12. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
13. Wang, F.Y., Zhou, D.W., Ye, H.J., Zhan, D.C.: Foster: Feature boosting and compression for class-incremental learning. In: European conference on computer vision. pp. 398–414. Springer (2022)
14. Wang, Q.W., Zhou, D.W., Zhang, Y.K., Zhan, D.C., Ye, H.J.: Few-shot class-incremental learning via training-free prototype calibration. arXiv preprint arXiv:2312.05229 (2023)
15. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European Conference on Computer Vision. pp. 631–648. Springer (2022)
16. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 139–149 (2022)

17. Yoon, I.U., Choi, T.M., Lee, S.K., Kim, Y.M., Kim, J.H.: Image-object-specific prompt learning for few-shot class-incremental learning. arXiv preprint arXiv:2309.02833 (2023)
18. Zhou, D.W., Wang, F.Y., Ye, H.J., Ma, L., Pu, S., Zhan, D.C.: Forward compatible few-shot class-incremental learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9046–9056 (2022)