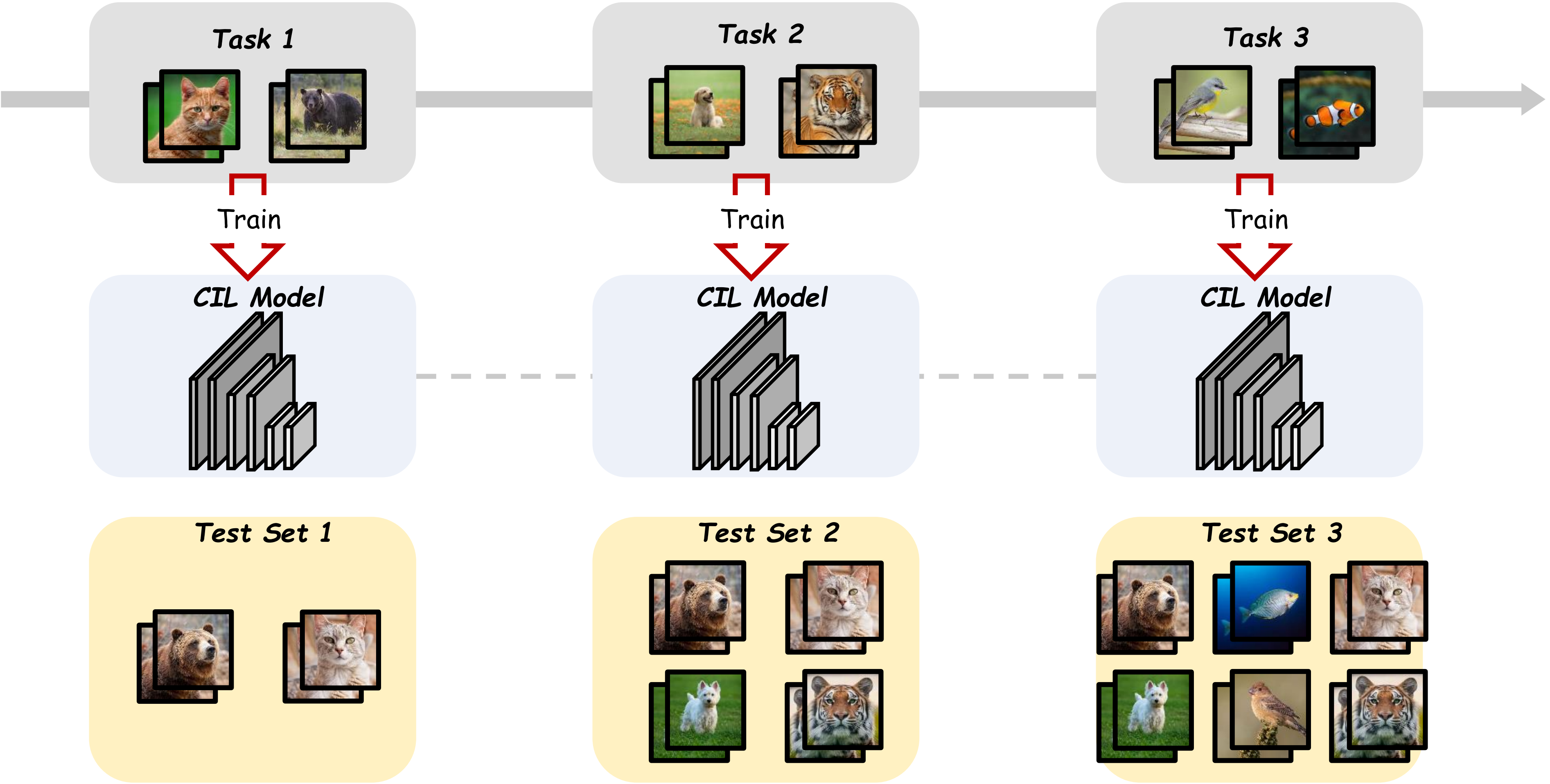
Long-Tail Class Incremental Learning via Independent Sub-prototype Construction

Xi Wang, Xu Yang, Jie Yin, Kun Wei, Cheng Deng*
School of Electronic Engineering, Xidian University, Xi'an 710071, China
{wangxi6317, xuyang.xd, jieyin.xd, weikunsk, chdeng.xd}@gmail.com

Presented by: Xi Wang

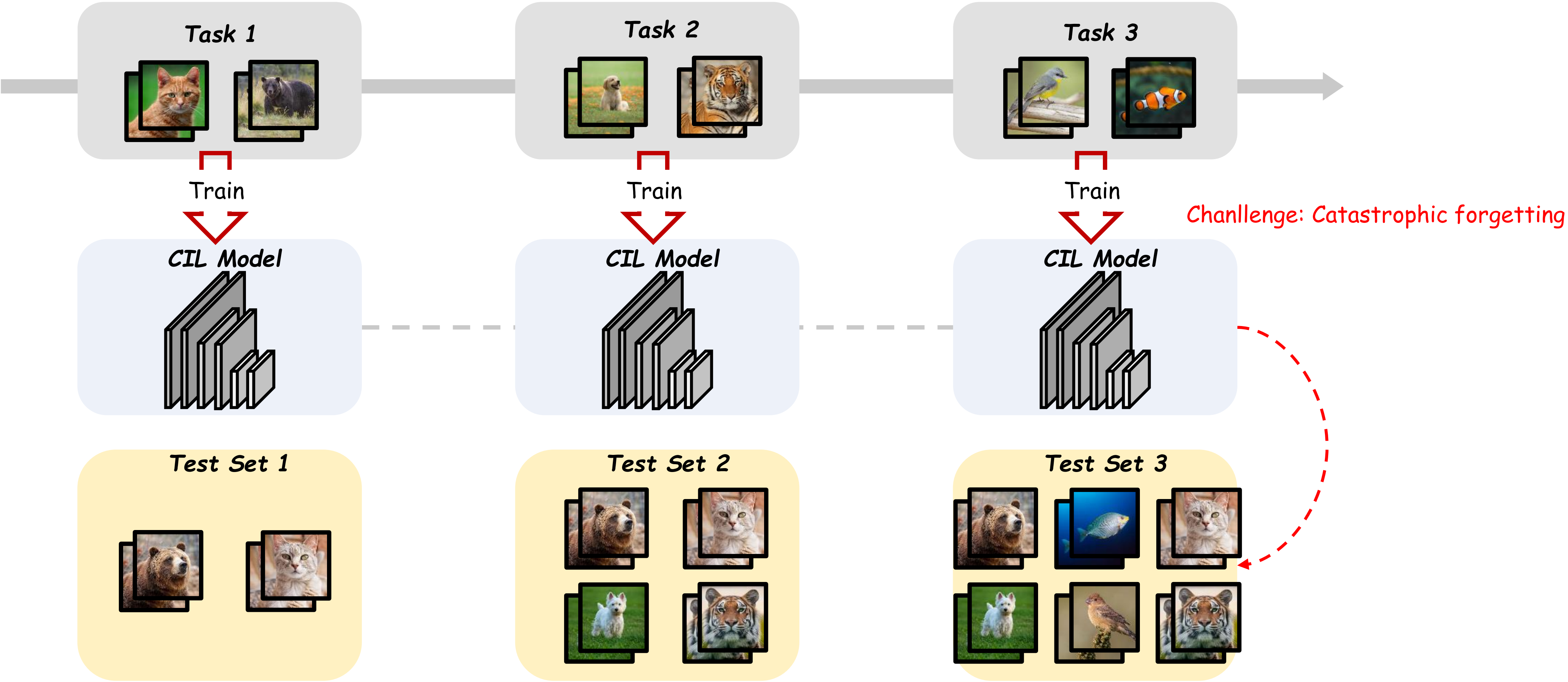
Background

- **Class Incremental Learning (CIL)**



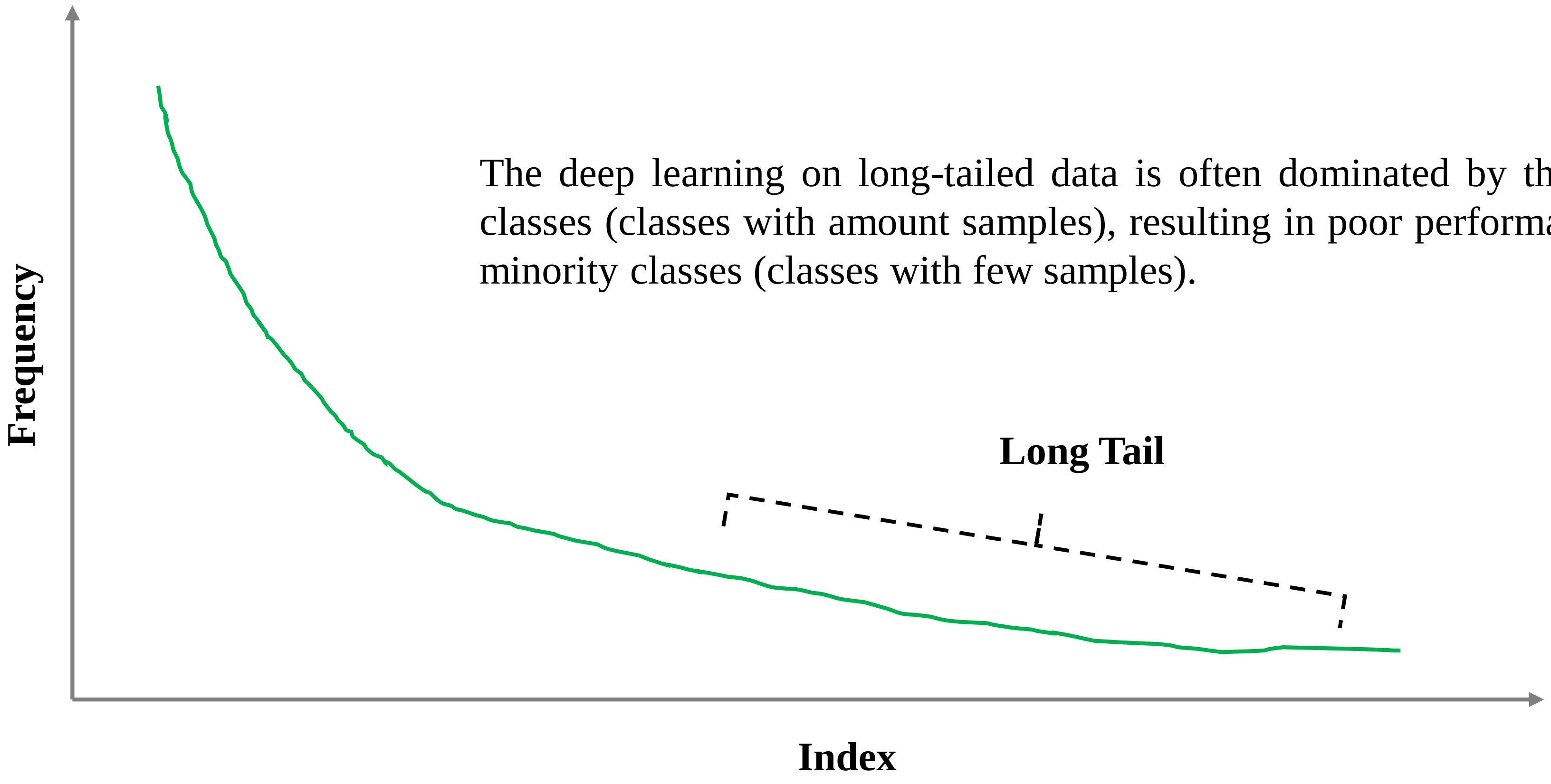
Background

➢ **Class Incremental Learning (CIL)**



Background

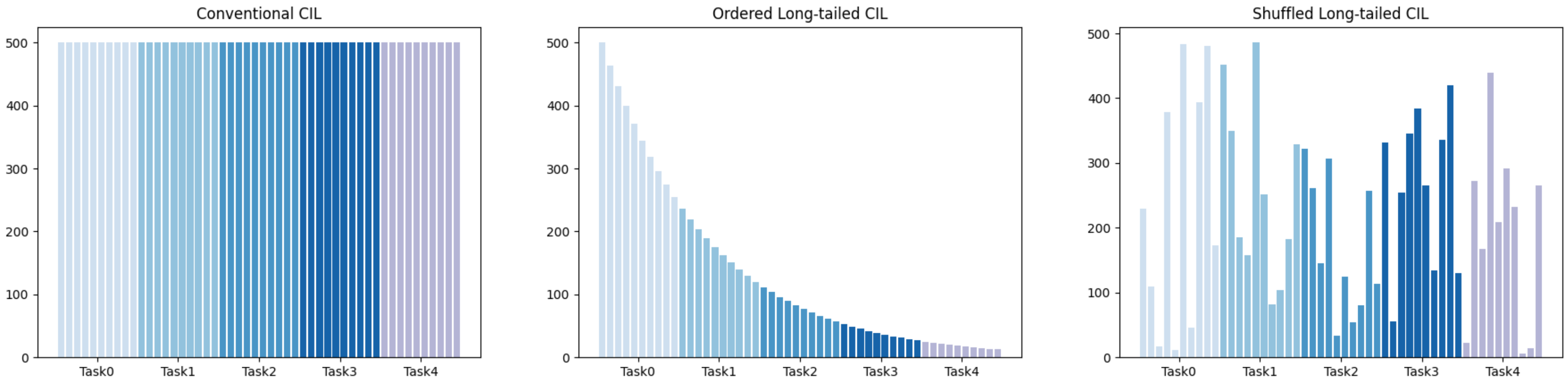
- **Imbalanced data distribution**



The deep learning on long-tailed data is often dominated by the majority classes (classes with amount samples), resulting in poor performance of the minority classes (classes with few samples).

Background

➢ **Long-tail class incremental learning (LT-CIL)**

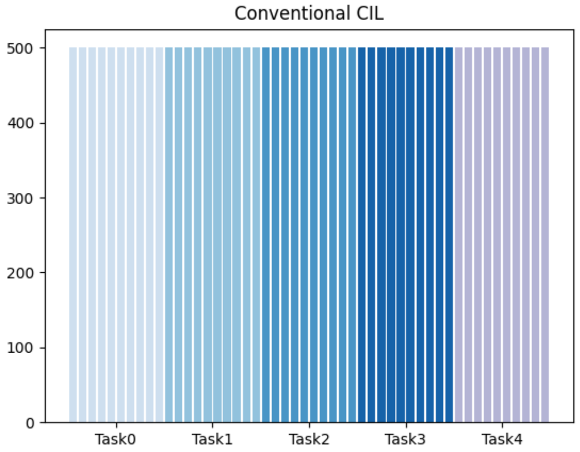


a) **b)** **c)**

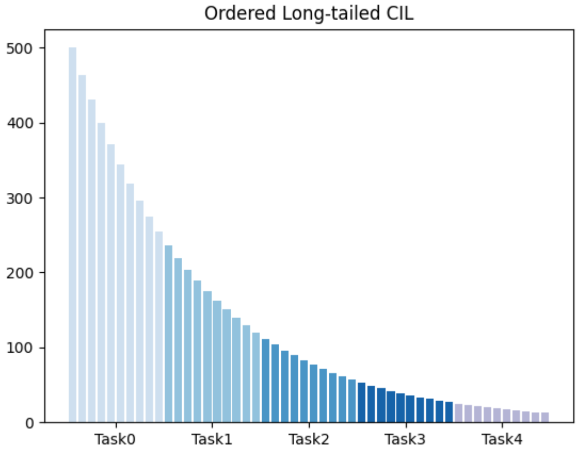
Long-tail class incremental learning (LT-CIL) is designed to perpetually acquire novel knowledge from an imbalanced and perpetually evolving data stream while ensuring the retention of previously acquired knowledge. To tackle this problem, existing work attempts to expand data or change the network structure and so on, both achieved good results. Surprisingly, continual learning on imbalanced data has yet to receive widespread attention.

Background

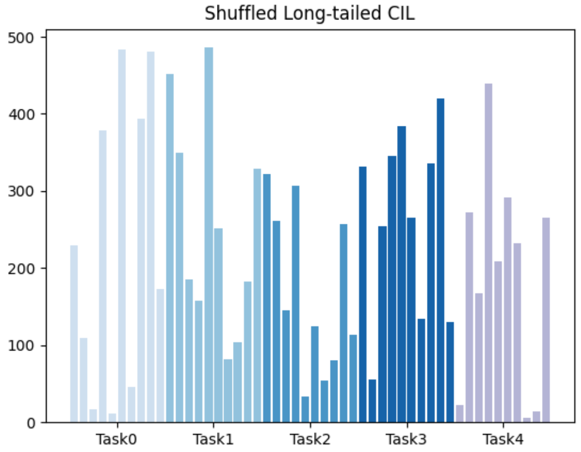
➢ **Long-tail class incremental learning (LT-CIL)**



a)



b)



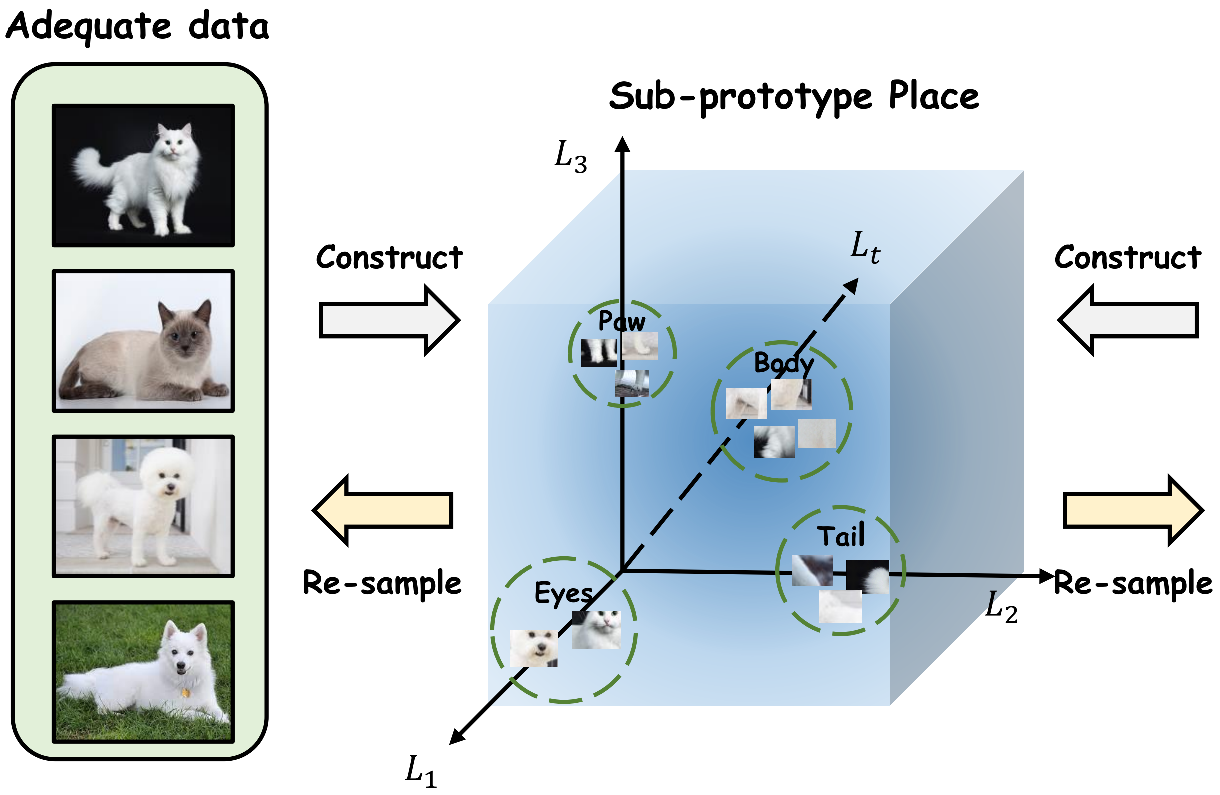
c)

a) Conventional class incremental learning requires the same number of new classes to be learned and the same number of samples of new classes to be learned in each task.
b) The dataset is sequentially divided into different tasks. There is an imbalanced distribution over the complete dataset, and the total number of samples within the tasks is decreasing.
c) The majority and minority classes randomly belong to any task. While there is still an imbalanced distribution in each task, the total number of samples from different tasks shows a random.

Motivation

➢ **Data relationship**

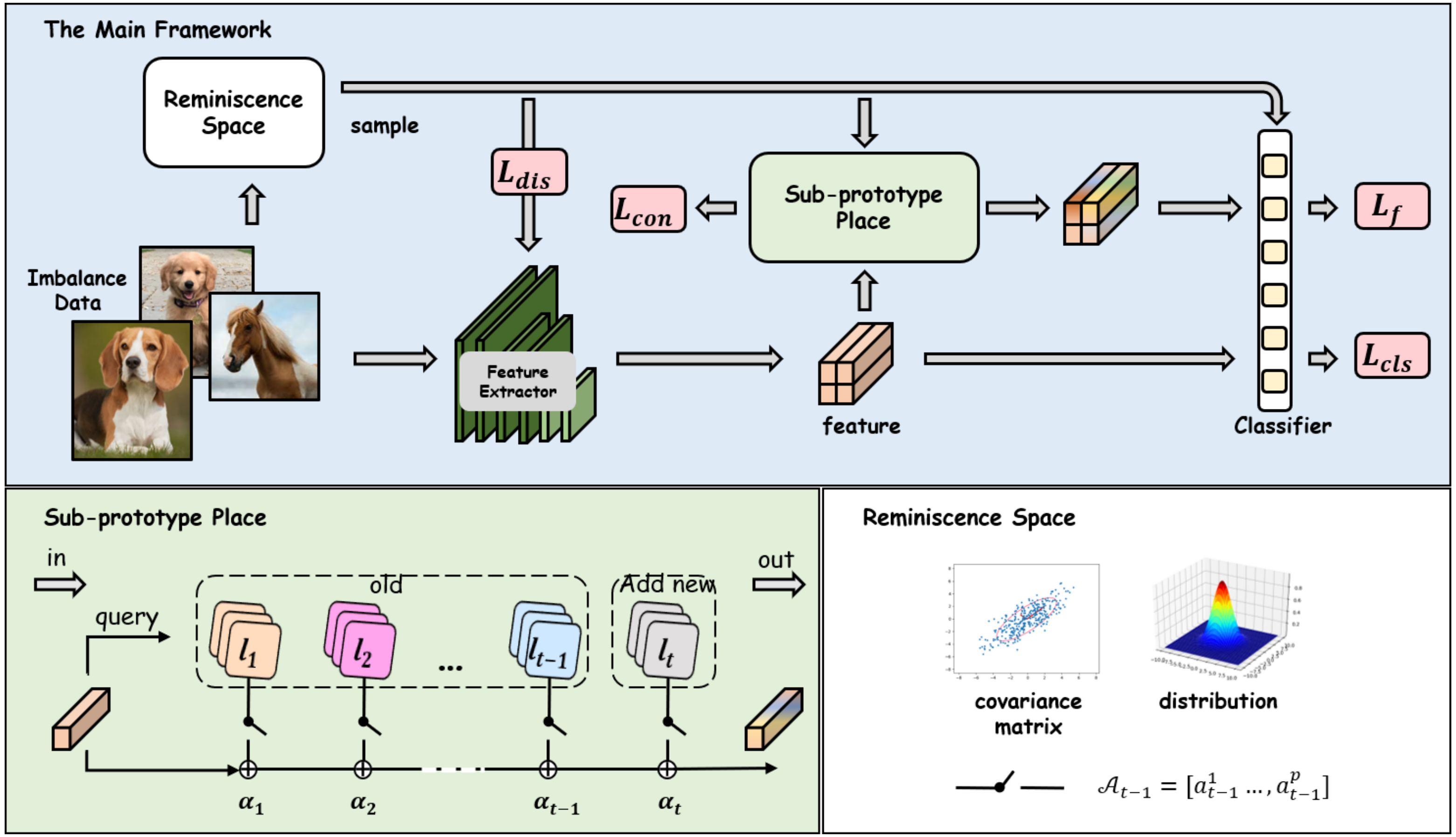
- Leveraging the richness of information from header class data to aid in the learning of tail class data.



Method

➢ Overview

- We propose a **Sub-prototype Space** and a **Reminiscence Space**



Method

➢ Sub-prototype Space

☐ space construction

- When a new set of learned features is input, we project the features into the existing basis vectors, query the correlation between the input features and the basis vectors:

$$A_t = Q(z^t, L_t) = [a_{ij}]_{i \in B, j \in M^t}$$

- Avoiding part of the basis vectors being selected too often, we add the controlling factor:

$$\eta_i = e^{-n_i}$$

$$A_t = Q(z^t, L_t) = [a_{ij}]_{i \in B, j \in M^t}$$

$$a_{ij} = \begin{cases} a_{ij} & \text{if} \quad a_{ij} \in [\mathcal{H} \cdot A[i:]]_{top_k} \\ 0 & \text{else} \end{cases}$$

Method

➢ Sub-prototype Space

◻ space construction

- The projection of the input features can be expressed as:

$$\widetilde{Z} = A_t \, L_t^{\top} = [\ \widetilde{z}_1, \quad \cdots \quad , \widetilde{z_B}\ ]^{\top}$$

- We use the L2 Norm to constrain the updating and the entire training loss is:

$$\mathcal{L}_1 = \mathcal{L}_{cls} + \lambda_1 \mathcal{L}_{con}$$

$$\mathcal{L}_{con} = \|\widetilde{Z} - Z\|_2,$$

Method

➢ Sub-prototype Space

☐ Feature re-sampling

- We resample feature from the sub-prototype space

$$\hat{z} = l_{top_1} + l_{top_2} + \mathbb{I}\left[\mathrm{C}_{M^t-2}^{h} \cdot \sum_{i=1}^{h} l_i\right]$$

- We use resample feature to calculate cross-entropy loss

$$\mathcal{L}_f(\phi; x, y) = \mathcal{L}_{ce}(\mathcal{Z}_{new}, Y_{new})$$

Method

➢ Reminiscence Space

☐ Storing data distributions

- Class feature centroid and corresponding covariance matrix:

$$\mu_k = \frac{1}{n_k} \sum_{i=1}^{n_k} f_\theta(x_i)$$

$$\mathcal{N}_k = \mathcal{N}(\mu_k, \Sigma_k)$$

- mean of the correlation matrix

$$\mathcal{A}^k = \frac{1}{n} \sum_{\{\triangle_k | y \in \text{ class } k\}} \triangle_k$$

Method

➢ Reminiscence Space

❑ Distillation loss

- We sample features and compute the cross-entropy loss

$$\widetilde{v}^k \sim \mathcal{S}_k$$

$$\mathcal{L}_{ce}(\widetilde{v}^k, k)$$

- We perform the distillation loss on the sub-prototype basis vectors

$$\mathcal{L}_{dis1} = \sum_{i=1}^{N_t} \left\| \mathcal{A}^i - Q\left(\mu_i, L_t\right)[: I_i] \right\|_2$$

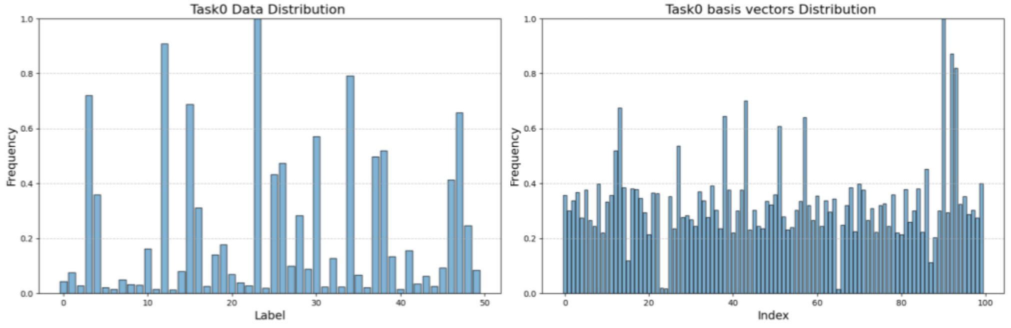
Experiments

➢ Results on Shuffled LT-CIL.

Table 1. **Results on Shuffled LT-CIL**. We compare our method with Baselines and previous methods.

Methods	Memory Size	Shuffled LT-CIL CIFAR100 $\rho = 0.01$ 5 tasks	CIFAR100 $\rho = 0.01$ 10 tasks	CIFAR100 $\rho = 0.05$ 5 tasks	CIFAR100 $\rho = 0.05$ 10 tasks	CIFAR100 $\rho = 0.1$ 5 tasks	CIFAR100 $\rho = 0.1$ 10 tasks	ImageNet-Subset $\rho = 0.01$ 5 tasks	ImageNet-Subset $\rho = 0.01$ 10 tasks	ImageNet-Subset $\rho = 0.05$ 5 tasks	ImageNet-Subset $\rho = 0.05$ 10 tasks	ImageNet-Subset $\rho = 0.1$ 5 tasks	ImageNet-Subset $\rho = 0.1$ 10 tasks
Baseline	0	11.3	7.3	13.6	8.1	13.9	8.2	12.9	11.2	15.3	13.1	16.7	12.7
LDAM[4]	0	11.5	11.4	15.2	14.3	19.8	18.2	15.4	15.2	19.4	19.0	22.6	22.1
BalPoE[1]		18.9	17.6	20.5	20.0	26.1	25.4	17.8	17.3	22.6	21.4	25.3	24.3
MDCS[47]		18.2	16.3	19.5	19.1	24.3	23.7	16.9	16.2	22.4	21.4	25.6	24.1
EWC[18]	0	28.7	25.3	33.1	31.9	40.6	39.7	30.8	30.4	35.6	34.7	43.8	43.6
LwF[21]		29.3	25.1	34.3	33.5	41.2	41.0	31.6	31.0	36.1	35.4	44.4	43.9
SDC[44]		32.7	29.6	35.2	34.1	42.9	42.3	33.9	33.4	39.4	38.1	45.9	45.1
PASS[49]		33.6	31.8	37.9	35.5	43.2	42.1	34.2	33.8	39.9	38.5	46.2	45.7
IL2A[48]		35.1	36.2	43.9	39.4	50.2	49.3	40.5	39.2	44.2	43.7	53.4	52.7
SAVC[34]		34.4	32.3	38.3	35.9	43.1	42.0	35.3	34.9	40.1	39.6	48.3	47.6
iCaRL[29]	1000	31.5	30.5	40.2	39.1	46.5	45.9	35.4	34.6	42	41.3	48.2	47.4
TwF[3]		34.2	33.8	42.3	42.1	49.3	48.7	38.6	38.1	43.6	43.3	52.2	51.7
SCoMMER[31]		35	35.2	43.4	42.3	49.9	49.1	39.3	38.9	44.9	44.1	52.9	52.6
LUCIR+LWS[22]		37.2	36.9	45.2	45.0	51.9	51.2	43.1	42.3	47.3	47.1	54.7	54.1
Ours	0	**40.2**	**39.4**	**47.3**	**47.0**	**53.6**	**53.1**	**45.3**	**44.8**	**49.2**	**48.9**	**56.2**	**55.4**

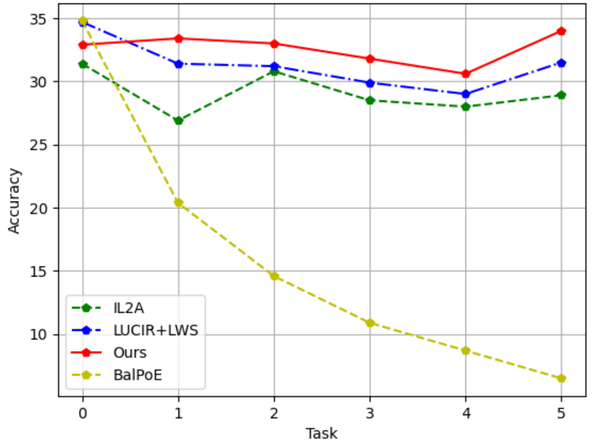
➢ Basis vectors selection frequency .



➢ Ablation Study.

	Components $\mathcal{L}_{cls}$	$\mathcal{L}_{con}$	$\mathcal{L}_{f}$	$\mathcal{L}_{dis1}$	$\mathcal{L}_{dis2}$	Accuracy
a)	√					11.3
b)	√	√				10.4
c)	√	√	√			34.7
d)	√	√	√		√	36.3
e)	√	√	√	√		38.9
f)	√	√	√	√	√	40.2

➢ Minority classes accuracy.



Conclusion

- In this work, we address two fundamental challenges in long-tail class incremental learning: intra-task imbalance due to data imbalanced distribution and inter-task imbalance due to forgetting what was learned before during incremental learning.

- We propose a novel and effective learnable sub-prototype space that simultaneously mitigates intra-task and inter task imbalances in long-tail class incremental learning.

- We propose a reminiscence space to store data distribution, which prevents the model from collapsing under the influence of new knowledge and forgetting the learned old knowledge during training.

- We perform extensive experiments to demonstrate the effectiveness of our method, all achieving state-of-the-art in diverse settings.

Thanks!