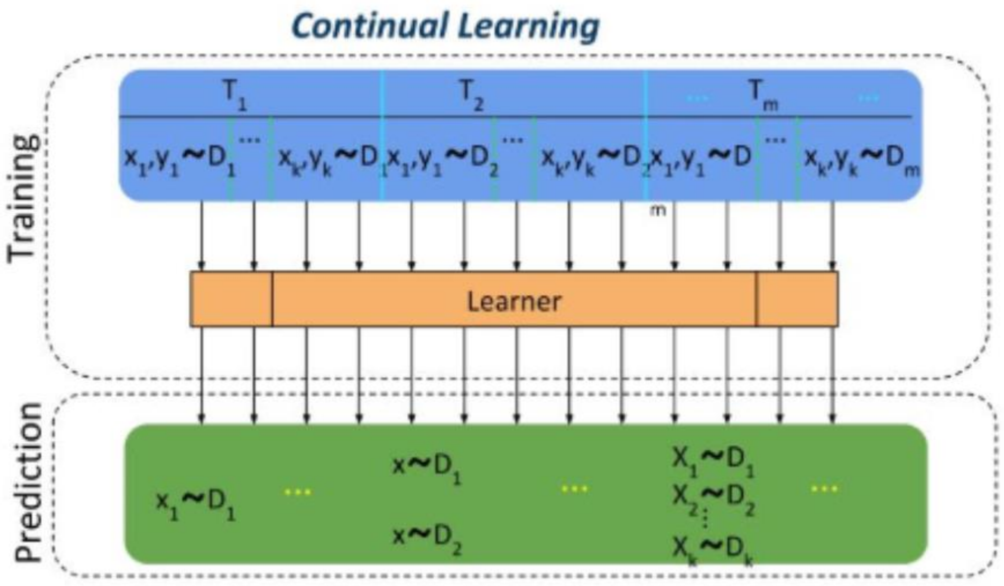
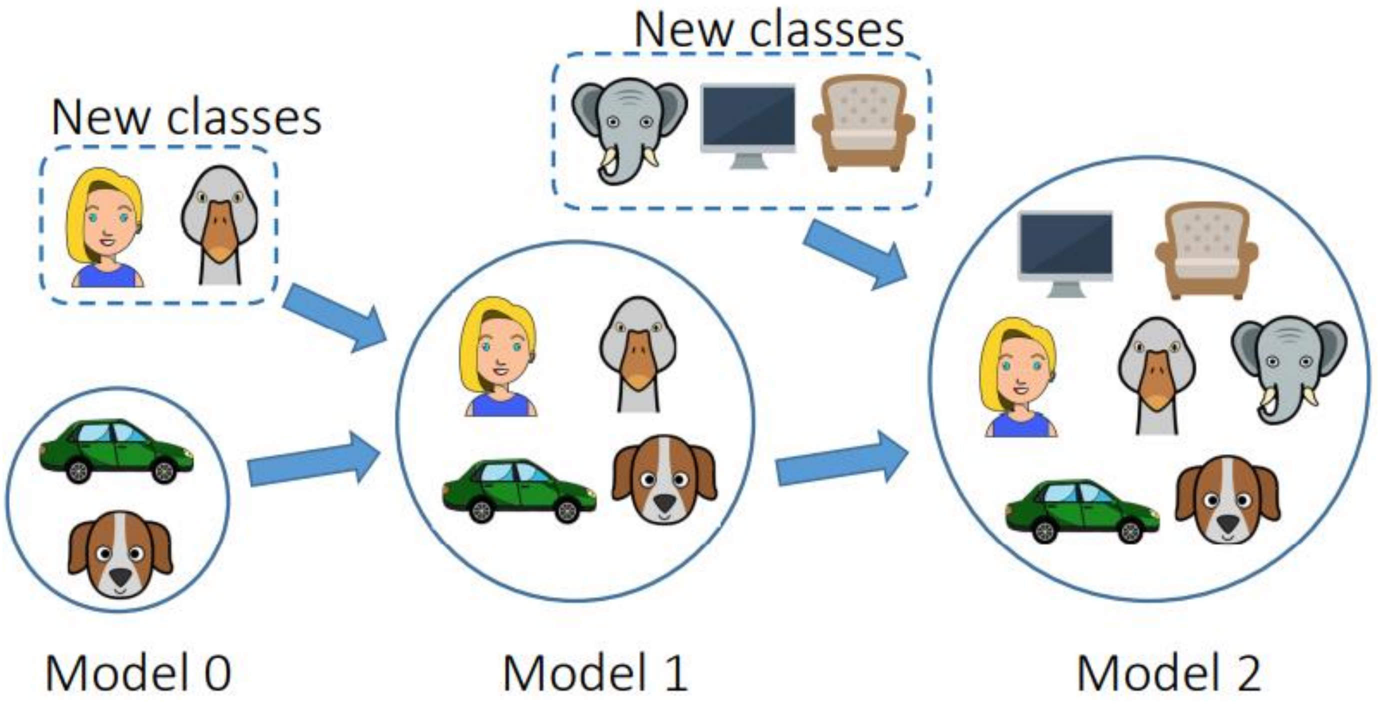
Prototype Augmentation and Self-Supervision for Incremental Learning

Fei Zhu[1,2], Xu-Yao Zhang[1,2*], Chuang Wang[1,2], Fei Yin[1,2], Cheng-Lin Liu[1,2,3]
[1]NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[3]CAS Center for Excellence of Brain Science and Intelligence Technology, Beijing 100190, China
zhufei2018@ia.ac.cn, {xyz, fyin, liucl}@nlpr.ia.ac.cn, wangchuang@ia.ac.cn

Incremental learning



Class Incremental learning

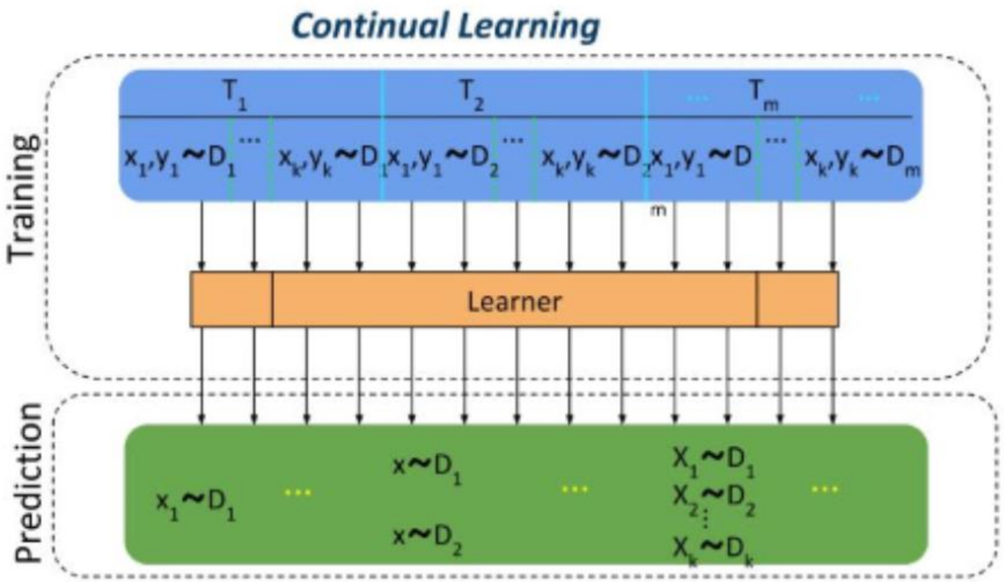


Incremental learning (Continual learning , lifelong learning):

- independent and identically distributed(独立同分布, i.i.d.)
- Stability-plasticity Dilemma

· Learning Step
· predictions with all seen categories

Incremental learning



Methods:

- **Replay**
- **Regularization-based**
- Parameter isolation

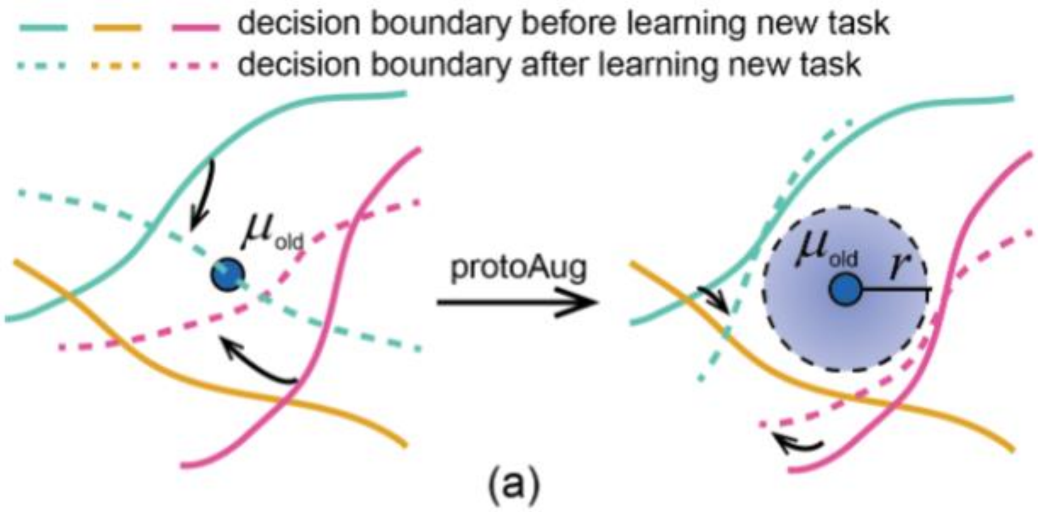
Incremental learning (Continual learning , lifelong learning):

- independent and identically distributed(独立同分布, i.i.d.)
- Stability-plasticity Dilemma

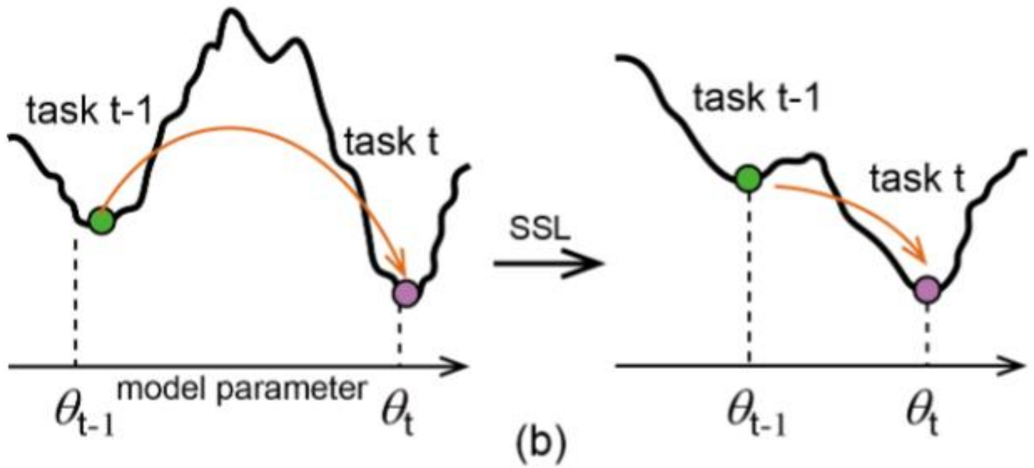
PASS

Motivation

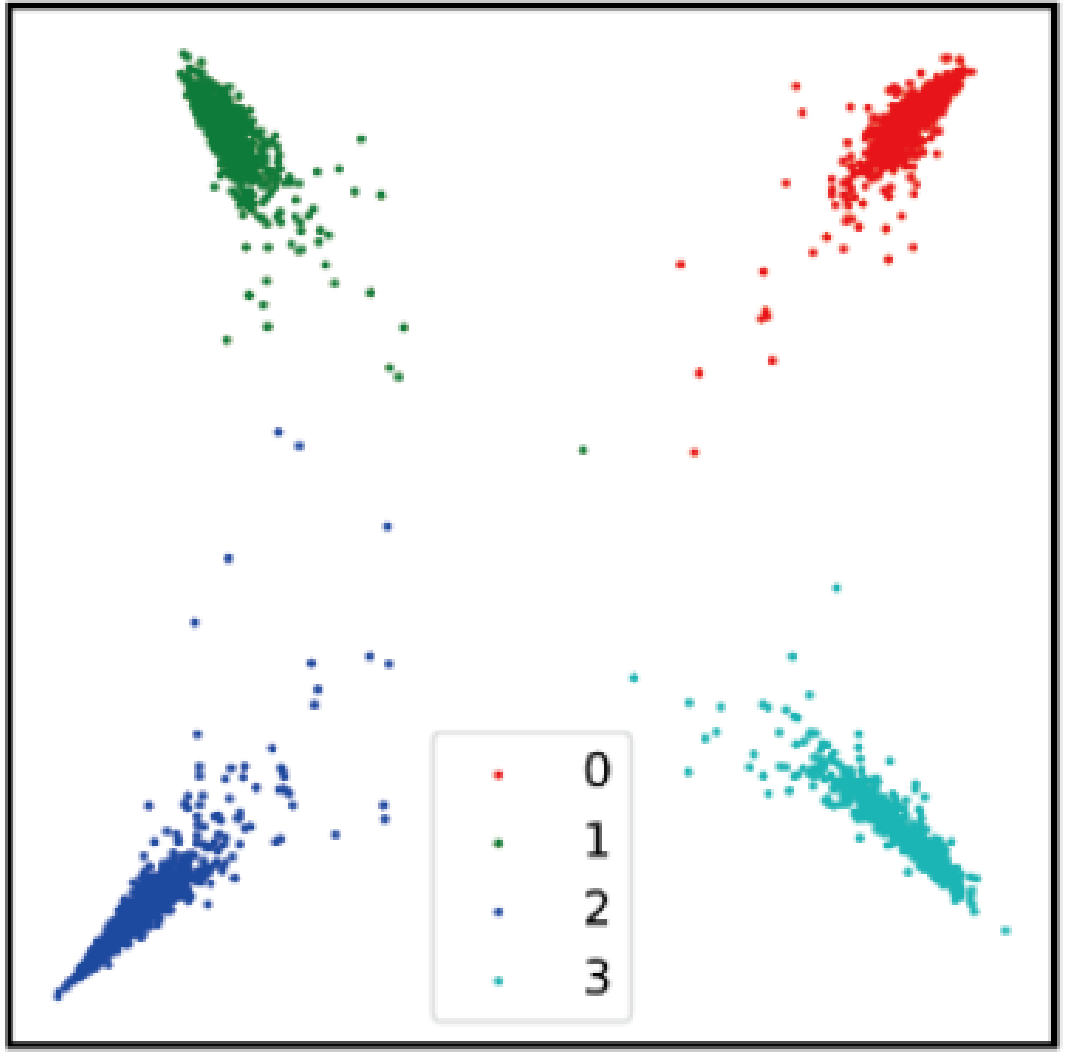
A. Need to maintain the balance between old and new classes

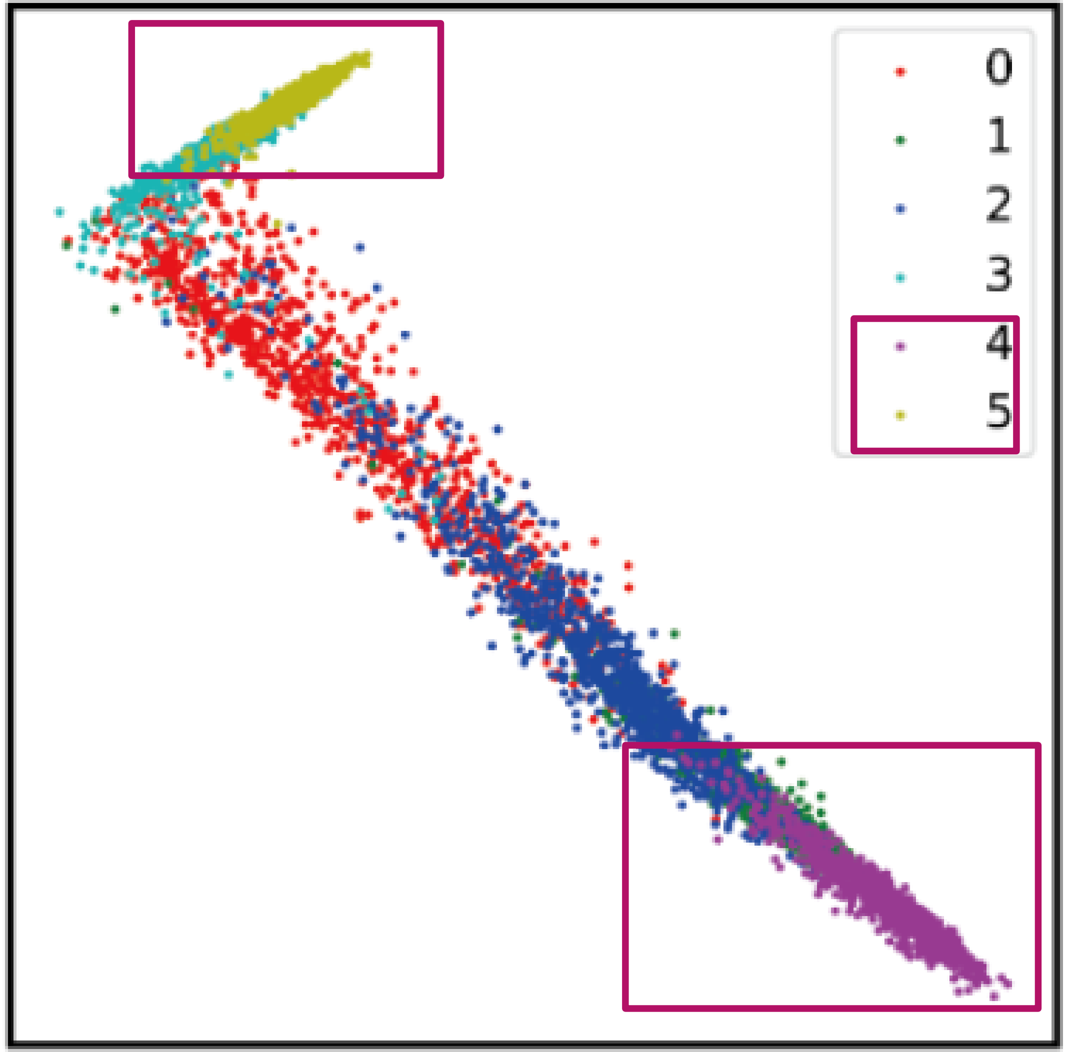


B. Learned features are task-specific in each stage, and maybe a bad initialization for current task



PASS

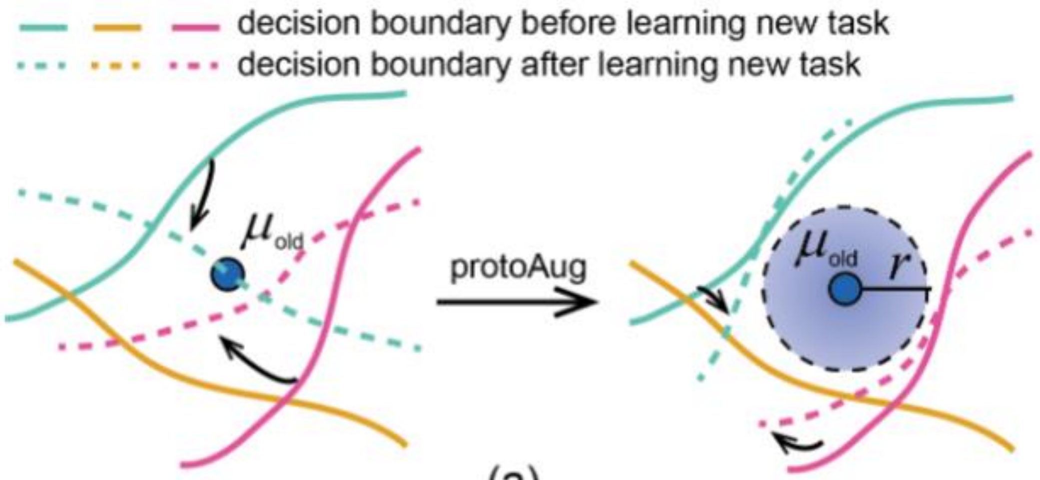




PASS

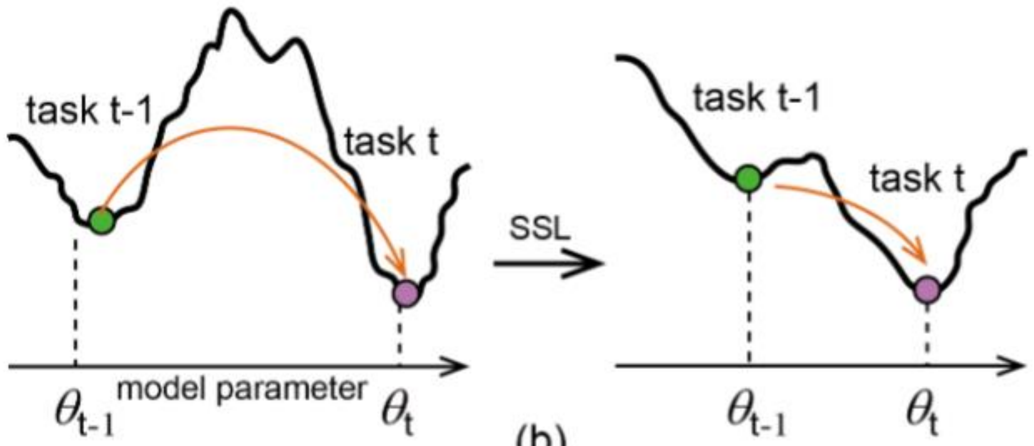
Motivation

A. Need to maintain the balance between old and new classes -> prototype augmentation



(a)

B. Learned features are task-specific in each stage, and maybe a bad initialization for the following tasks



(b)

To learn more general representation

PASS

A. Label Augmentation(Self-Supervision)
B. Prototype Augmentation
C. Knowledge Distillation

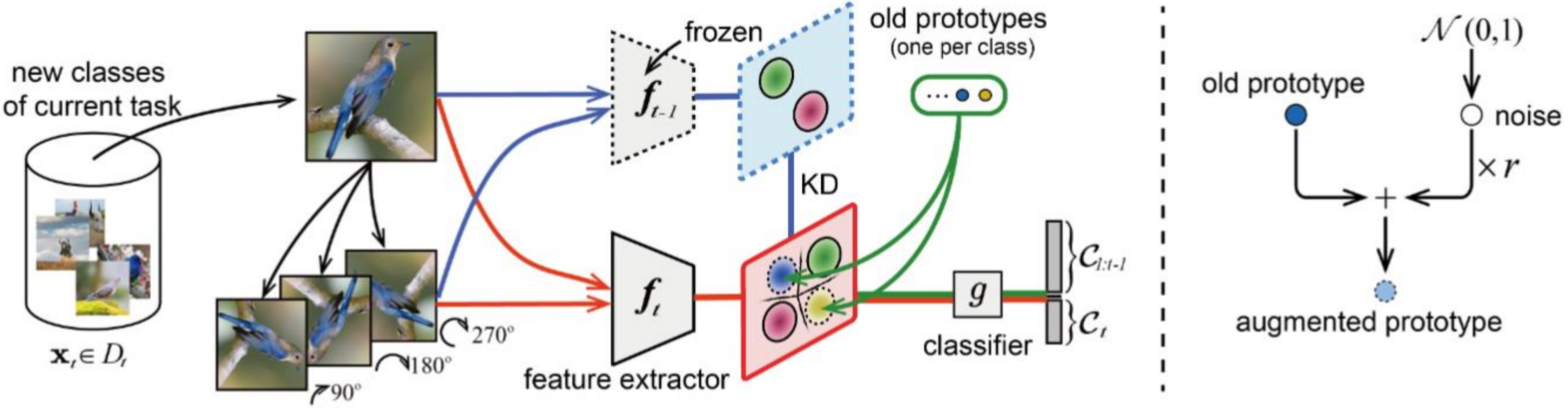
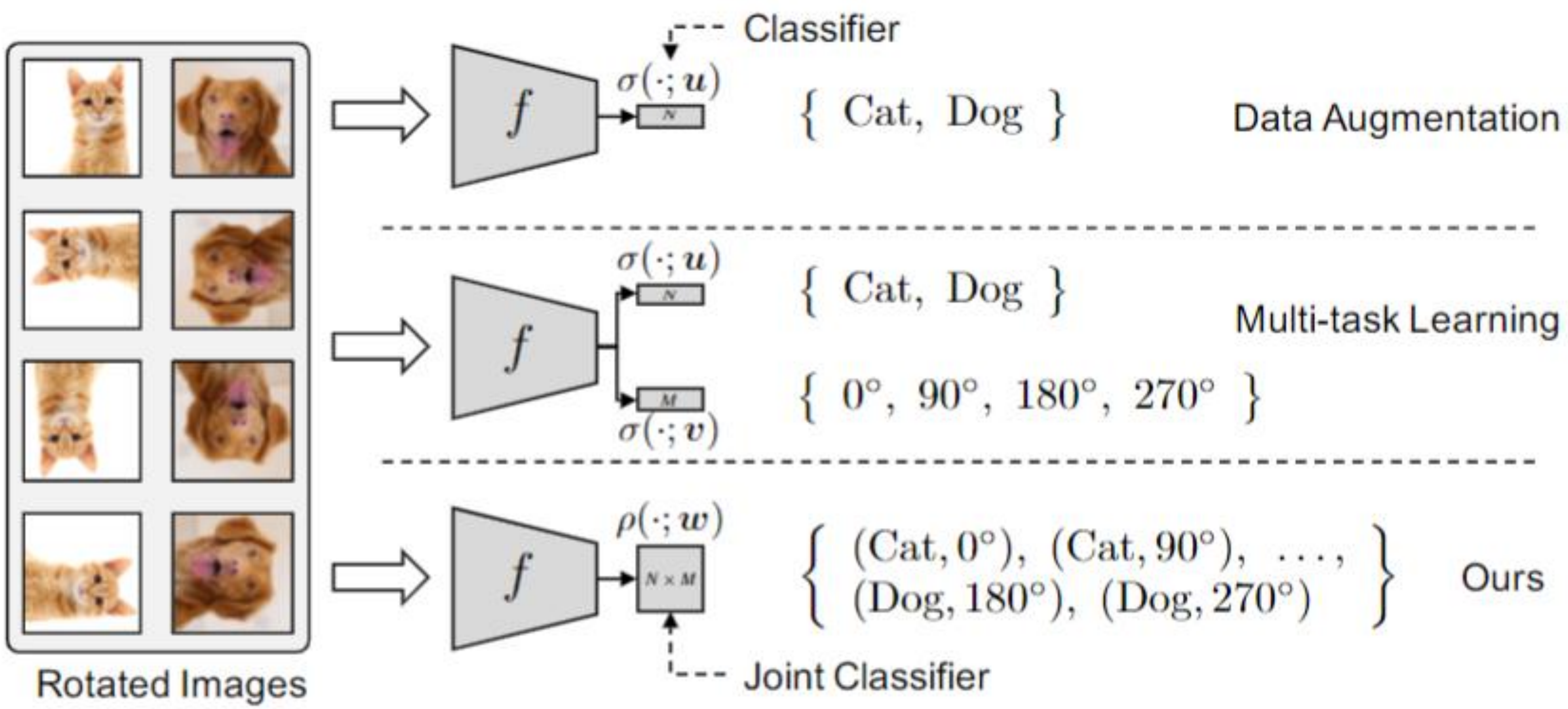


Figure 2: Illustration of PASS for CIL. The classes of current task are augmented by rotation based transformation [32], and the augmented data are fed to the feature extractor. In the deep feature space, we augment the memorized prototypes (one for each classes) via Gaussian noise (right). Our method is non-exemplar based, simple and effective.

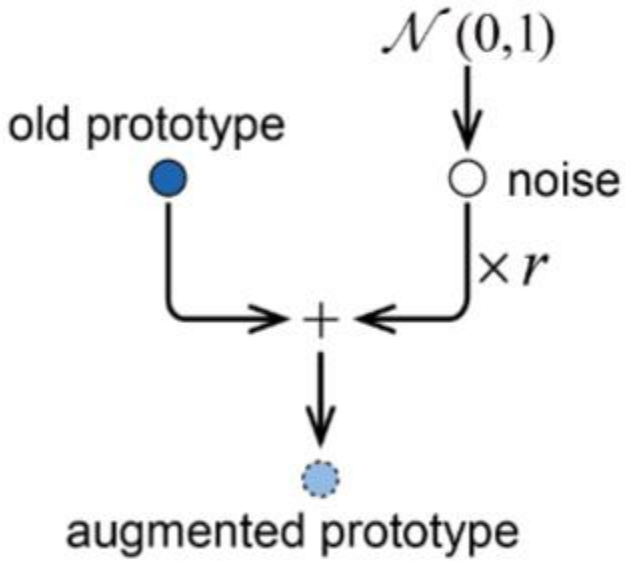
Label augmentation



Extend the label set of K-classes problem
Calculate CELoss $L_{t,CE}$ with **4K**-classes

Lee H, Hwang S J, Shin J. Self-supervised label augmentation via input transformations[C]//International Conference on Machine Learning. PMLR, 2020: 5714-5724.

Prototype augmentation



Old prototype $\mu_{t_{old},k_{old}}$ is *calculated* by the mean of feature embeddings of the category

$$F_{t_{old},k_{old}} = \mu_{t_{old},k_{old}} + e * r,$$

$$e \sim \mathcal{N}(0,1)$$

r: Hyper Parameters

$$L_{protoaug} = CELoss(G_t(F_{t_{old},k_{old}}, \theta), Y_{old})$$

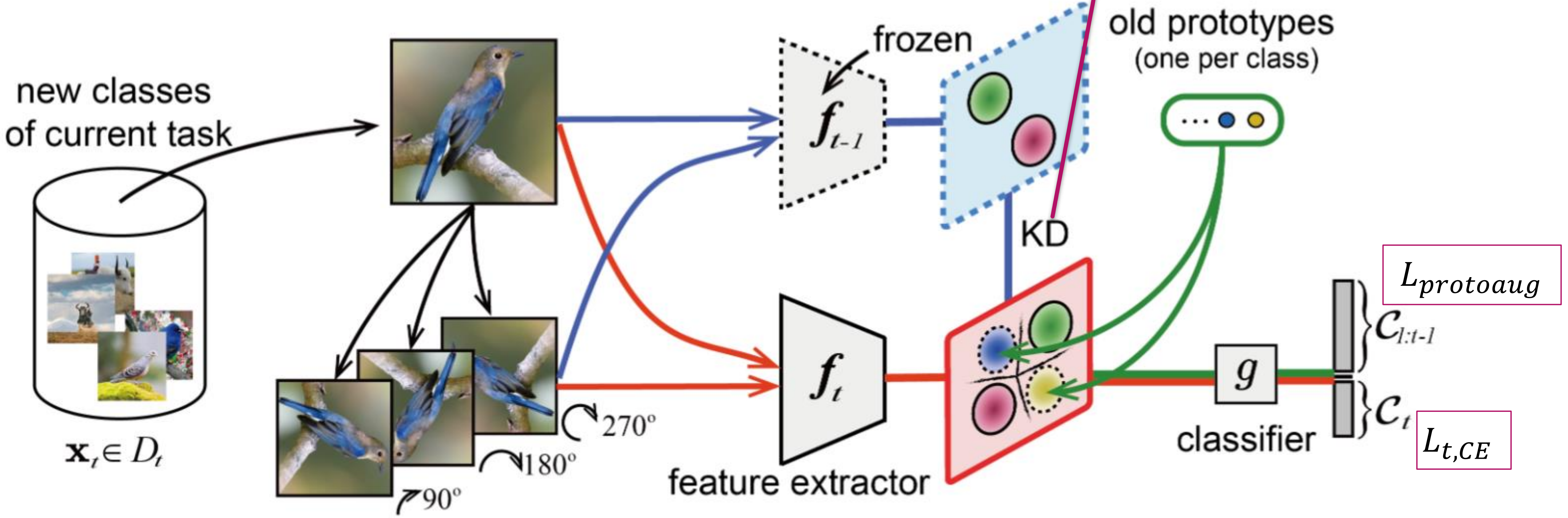
```
soft_feat_aug = self.model.fc(proto_aug)
loss_protoAug = nn.CrossEntropyLoss()(soft_feat_aug/self.args.temp, proto_aug_label)
```

Loss && Overview

$$L_{t,total} = L_{t,ce} + \lambda * L_{t,protoAug} + \gamma * L_{t,kd}$$

$$\lambda = \gamma = 10$$

$$\boxed{L_{t,kd} = \|F_t(\mathbf{X}'_t;\theta_t) - F_{t-1}(\mathbf{X}'_t;\theta_{t-1})\|}$$



Inference: Orange Path

Experiments

#dataset & classes				CIFAR-100			TinyImageNet		
	Method	protoAug	SSL	5 phases	10 phases	20 phases	5 phases	10 phases	20 phases
Accuracy	KD	✗	✗	14.33	6.04	5.67	7.23	4.70	4.23
	KD+SSL	✗	✓	17.15	8.46	8.57	9.71	6.53	6.60
	KD+protoAug	✓	✗	50.19	39.80	38.61	33.11	26.52	20.97
	KD+protoAug+SSL	✓	✓	**55.67**	**49.03**	**48.48**	**41.58**	**39.28**	**32.78**
Forgetting	KD+protoAug	✓	✗	28.72	35.70	40.59	25.62	35.33	43.91
	KD+protoAug+SSL	✓	✓	**25.20**	**30.25**	**30.61**	**18.04**	**23.12**	**30.55**

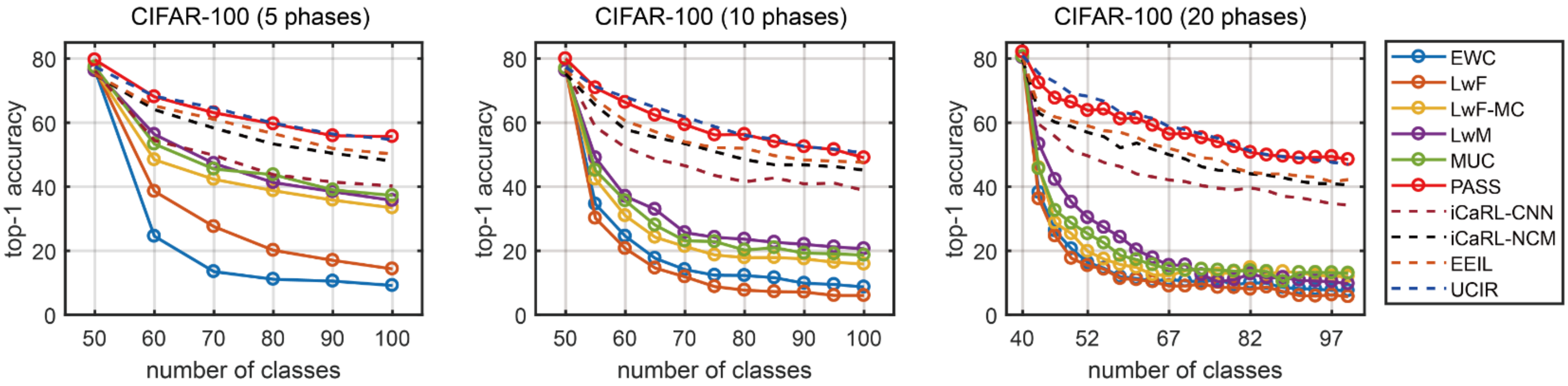
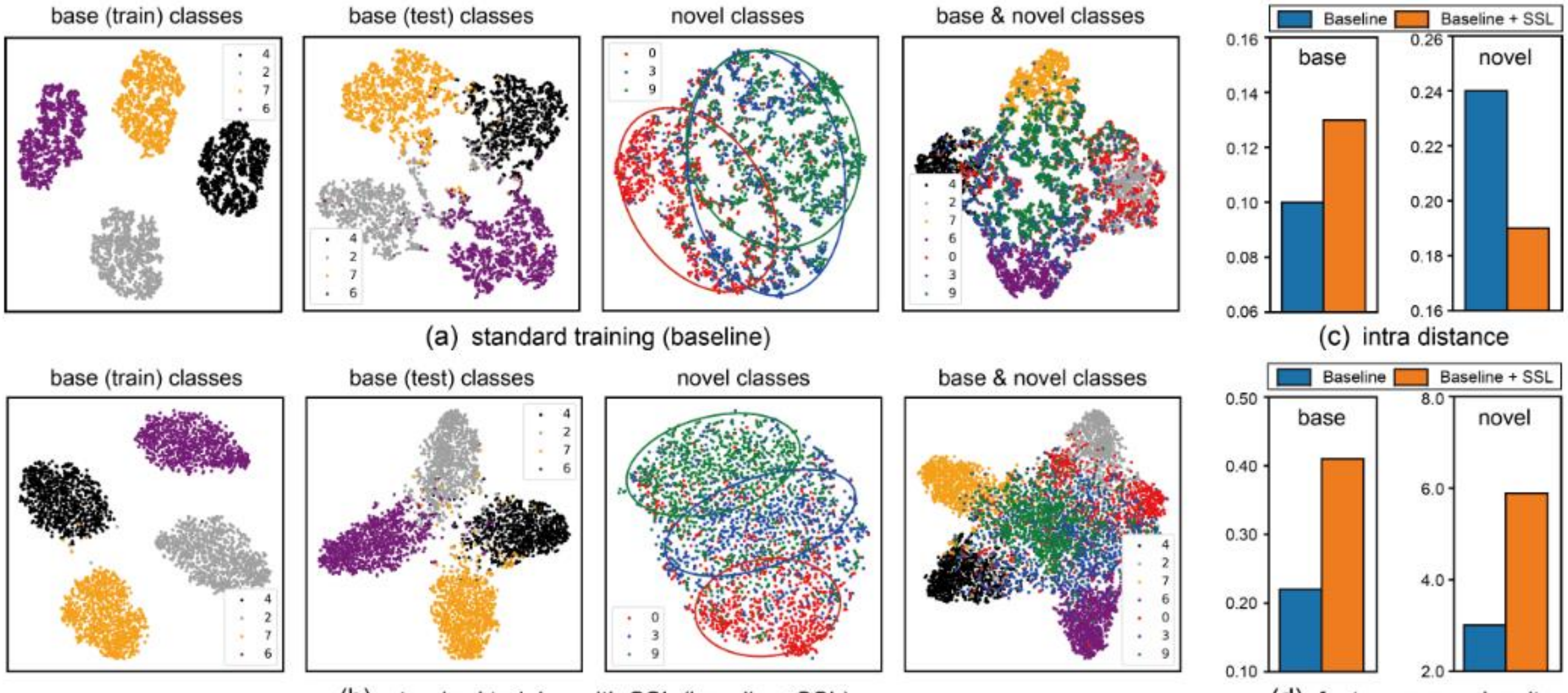


Figure 5: Results of classification accuracy on CIFAR-100, which contains 5, 10 and 20 sequential tasks.

Experiments



(a) standard training (baseline)

(b) standard training with SSL (baseline+SSL)

(c) intra distance

(d) feature space density