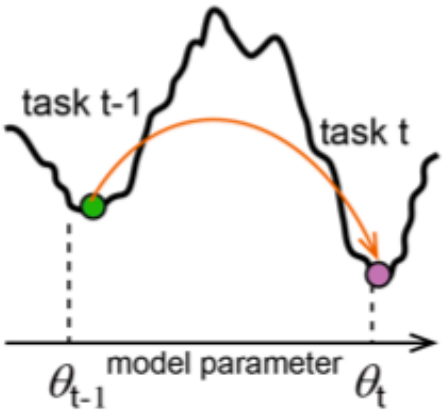
Prototype Augmentation and Self-Supervision for Incremental Learning

1、Motivation

通过存储少量previous task数据的方式放置灾难性遗忘的方法取得了不错的效果，但是其代价是较大的存储开销。

有人提出可以用生成式网络来生成伪数据代替直接存储，但是GAN这样的生成式网络本身就很难训练，而且也存在灾难性遗忘的问题。之前也有exemplar-free的基于正则化的方法，但是这样的方法更适用于task-incremental learning，本文关注的是class-incremental learning这个场景。

同时，也有研究表明train过之前task的网络未必是对当前task的一个好的初始化（甚至不如随机初始化），原因是模型在previous task上过拟合。



因此，作者希望能够保持CIL的分类边界同时减轻模型在previous task上过拟合的现象。对前者，作者提出了prototype augmentation的方法。对于后者，作者提出使用监督的方法。

2、Approach

2.1 Problem statement and analysis

class sets of different task are disjoint. At step t, the goal is to minimize a predefined loss function L on new dataset D_t without interfering with and possibly improving on those that were learned previously [1]:

$$\begin{aligned} \{\theta_t, \phi_t\} &= \underset{\theta_t, \phi_t, \epsilon}{\operatorname{argmin}} L_t(G(F(\mathbf{X}_t; \theta_t); \phi_t), \mathbf{Y}_t) + \sum \epsilon_i \\ \text{s.t. } L_t(\mathbf{X}_i, \mathbf{Y}_i) &- L_i(\mathbf{X}_i, \mathbf{Y}_i) \leqslant \epsilon_i, \epsilon_i \geqslant 0;\ \forall i \in [1, t-1] \end{aligned}, \tag{1}$$

where $L_t(\mathbf{X}_i, \mathbf{Y}_i) = L(G(F(\mathbf{X}_i; \theta_t); \phi_t), \mathbf{Y}_i)$ is the loss of the model at t on old data set D_i and $L_i(\mathbf{X}_i, \mathbf{Y}_i) = L(G(F(\mathbf{X}_i; \theta_i); \phi_i), \mathbf{Y}_i)$ is the loss of the previous model at i on old dataset D_i. The last term $\epsilon = \{\epsilon_i\}$ is a slack variable that tolerates a small increase in old dataset.

作者设计的模型结构如下所示：

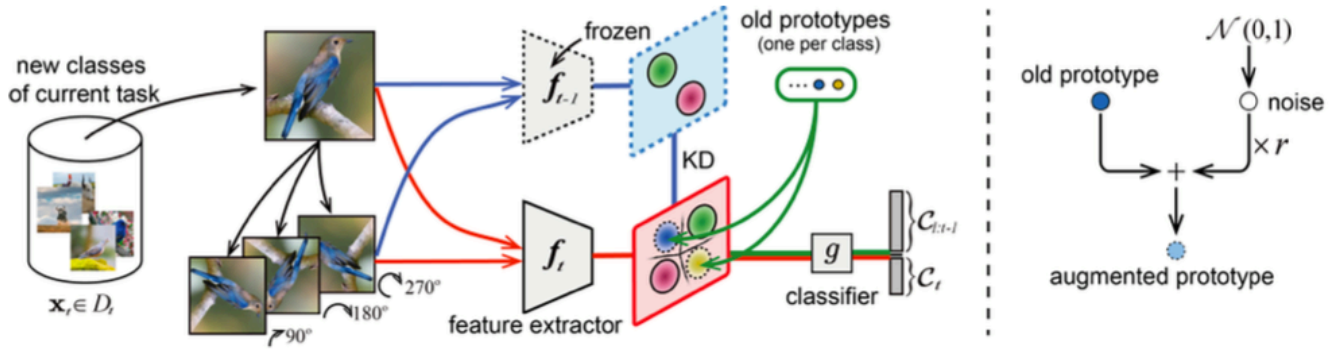


Figure 2: Illustration of PASS for CIL. The classes of current task are augmented by rotation based transformation [32], and the augmented data are fed to the feature extractor. In the deep feature space, we augment the memorized prototypes (one for each classes) via Gaussian noise (right). Our method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

对于旧类别，不存储任何数据，而是存储class-representative prototype。在学习新任务的时候，将这些prototype进行增强后跟新数据的特征一起送入分类器。同时，为了减轻task-level overfitting，预测图片旋转角度对自监督机制也被引入了模型。

2.2 Prototype Augmentation

首先，通过计算旧类别数据特征均值的方式得到旧类别数据的prototype：

$$\mu_{t,k} = \frac{1}{N_{t,k}} \sum_{n=1}^{N_{t,k}} F(X_{t,k}; \theta_t).$$

在学习新任务的时候，对旧类别的每个prototype，通过下面式子进行增强：

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

其中e～N(0,1)，r可以被预先定义， 也可以通过计算类别表现的平均方差得到：

$$r_t^2 = \frac{1}{K_{old}+K_{new}}(K_{old} * r_{t-1}^2 + \sum_{k=1}^{K_{new}} \frac{\mathrm{Tr}(\Sigma_{t,k})}{D})$$

where K_{old} and K_{new} represent the number of old classes and new classes at stage t, respectively. D is the dimension of the deep feature space. $\Sigma_{t,k}$ is the covariance matrix for the features from class k at stage t, and the Tr operation computes the trace of a matrix. We observed that the r_t

最后，将这些增强过的prototype和新的数据得到的特征一起送入分类器进行分类，最后计算损失如下：

$$\{\theta_t, \phi_t\} = \underset{\theta_t,\phi_t,\epsilon}{\mathrm{argmin}}\{L_t(G(F(\mathbf{X}_t;\theta_t);\phi_t), \mathbf{Y}_t) + \sum_{i=1}^{t-1} L(G(F_i;\phi_t), \mathbf{Y}_i)\},$$

个人理解：这样的做法是为了将新类别的数据融合进旧数据的特征空间中。

2.3 SSL based label agumentation

作者将每张新类别的图片分别旋转90，180，270°，最后分类的时候做4分类。

2.4 Integrated Objective of PASS

作者用知识蒸馏的方法，保持模型对新类别数据提取特征的一致：

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

最后模型的训练损失可以写为：

Combining the techniques presented above, we reach a total loss of PASS that comprised of three terms, given as:

$$L_{t,total} = L_{t,ce} + \lambda * L_{t,protoAug} + \gamma * L_{t,kd}. \quad (8)$$

$L_{t,ce} = L_{t,ce}(G(F(\mathbf{X'}_t;\theta_t);\phi_t), \mathbf{Y'}_t)$, and $L_{t,protoAug} = \sum_{i=1}^{t-1} L_{t,ce}(G(F_i;\phi_t), \mathbf{Y}_i)$. λ and γ are loss weights, and we use $\lambda = \gamma = 10$ in our experiments.

3、Experiment

3.1 2D visualization of ProtoAug

首先作者在MNIST上可视化ProtoAug的方法（不含SSL）

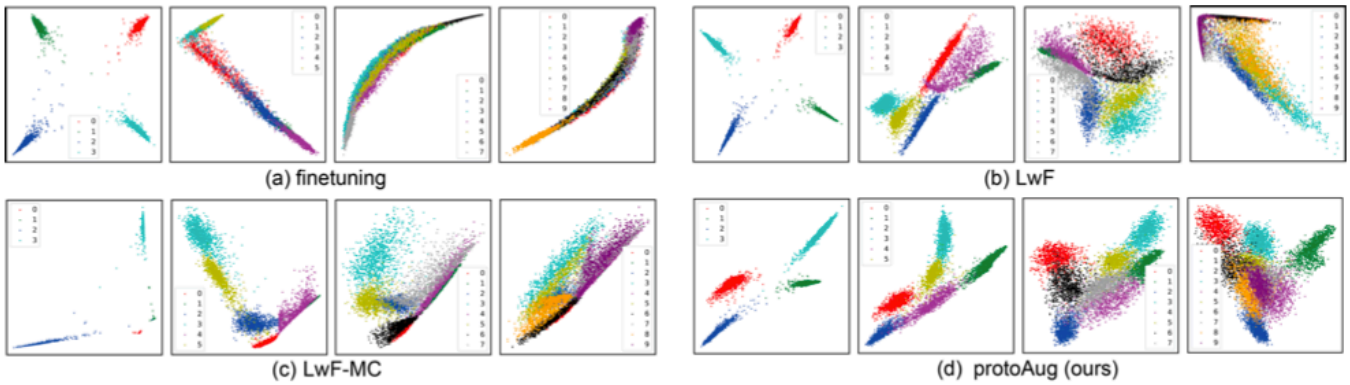


Figure 3: Visualization of class representations in the feature space when learning MNIST [31] incrementally. The outputted features are *2-dimensional* which is suitable for visualization. Best viewed in color.

可以发现，ProtoAug可以有效地保持旧类别的分布，从而尽可能减轻CIL中的灾难性遗忘。

3.2 A closer look at SSL for CIL

Table 1: Results of zero-cost class incremental learning. The model is tested using nearest class mean classifier.

#classes			4 (base)	5	6	7	8	9	Final	Average
CIFAR-10	Novel	Baseline	—	27.20	20.55	17.40	17.23	15.68	14.80	18.81
		+ SSL	—	**76.40**	**61.10**	**46.83**	**40.80**	**40.36**	**37.57**	**50.50** $_{+31.69}$
	All	Baseline	94.55	79.26	68.00	59.65	52.88	48.97	46.46	64.25
		+ SSL	**95.35**	**87.26**	**79.22**	**70.04**	**64.05**	**61.18**	**58.36**	**73.63** $_{+9.38}$
#classes			40 (base)	50	60	70	80	90	Final	Average
CIFAR-100	Novel	Baseline	—	43.50	33.10	30.43	27.45	25.20	23.58	30.54
		+ SSL	—	**55.70**	**44.85**	**42.67**	**38.37**	**34.70**	**32.15**	**41.46** $_{+10.92}$
	All	Baseline	71.83	63.60	55.73	50.64	46.38	42.61	39.37	52.93
		+ SSL	**72.03**	**64.52**	**58.37**	**54.46**	**50.50**	**46.48**	**43.46**	**55.68** $_{+2.74}$

加入SSL以后效果有很明显的提升。

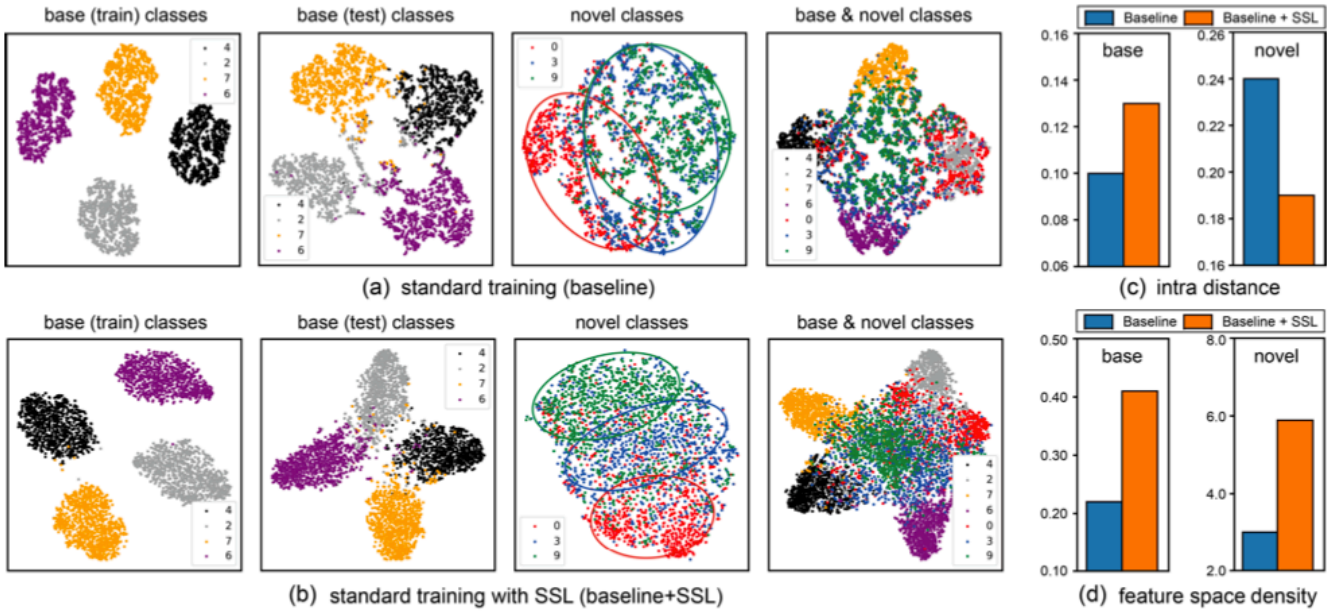


Figure 4: (a-b) SSL improves the separation of the distribution of novel classes, and reducing the the overlap between base and novel classes. (c-b) SSL results in smaller intra distance on novel classes, and high feature space density.

特征的分布也更为分散。

3.3 Comparative results

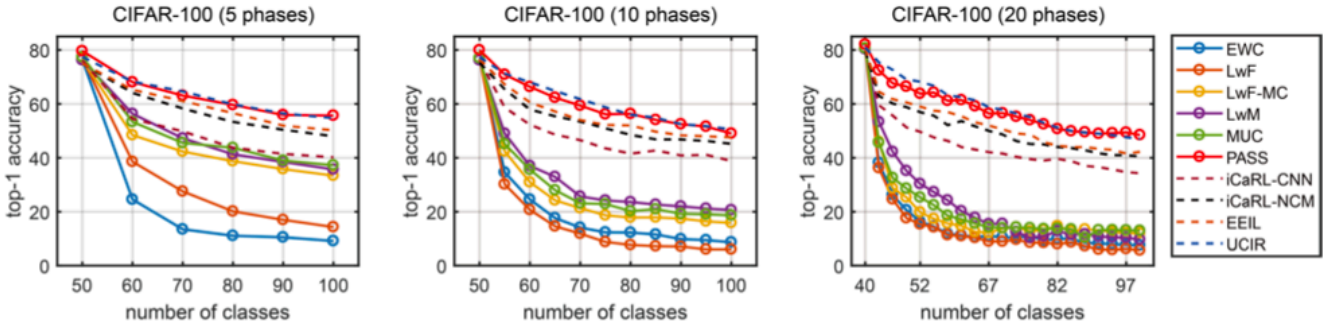


Figure 5: Results of classification accuracy on CIFAR-100, which contains 5, 10 and 20 sequential tasks.

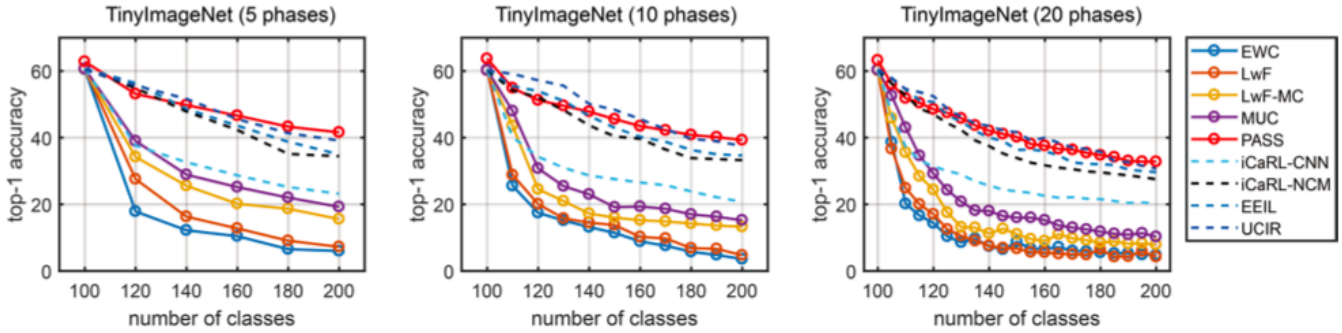


Figure 6: Results of classification accuracy on TinyImageNet, which contains 5, 10 and 20 sequential tasks.

方法的效果远超无回放的方法， 并且在有回放的方法中也取得了很棒的效果。

Table 2: Results of average forgetting on CIFAR-100 and TinyImageNet.

	CIFAR-100			TinyImageNet		
Method	5 phases	10 phases	20 phases	5 phases	10 phases	20 phases
LwF_MC	44.23	50.47	55.46	54.26	54.37	63.54
MUC	40.28	47.56	52.65	51.46	50.21	58.00
PASS	25.20	30.25	30.61	**18.04**	23.11	**30.55**
iCaRL-CNN	42.13	45.69	43.54	36.89	36.70	45.12
iCaRL-NCM	24.90	28.32	35.53	27.15	28.89	37.40
EEIL	23.36	26.65	32.40	25.56	25.91	35.04
UCIR	**21.00**	**25.12**	**28.65**	20.61	**22.25**	33.74

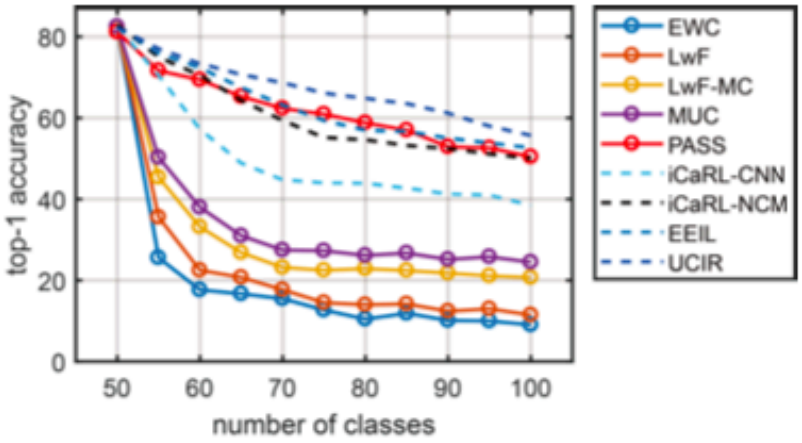


Figure 7: Results of classification accuracy on ImageNet-Subset, which contains 10 sequential tasks.

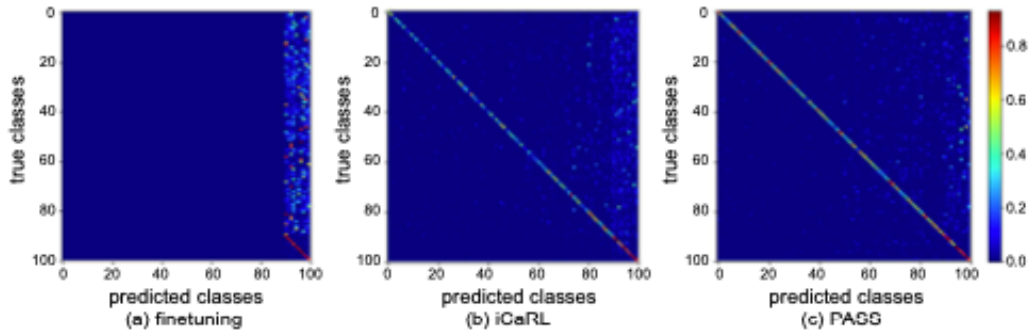


Figure 8: The comparison of confusion matrix of finetuning, iCaRL and PASS.

3.4 Ablation study

Table 3: The effectiveness of each component in our method.

#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**