Class-Incremental Learning via Dual Augmentation

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep learning systems typically suffer from catastrophic forgetting of past knowledge when acquiring new skills continually. In this paper, we emphasize two dilemmas, *representation bias* and *classifer bias* in class-incremental learning, and present a simple and novel approach which employs explicit *class augmentation (classAug)* and implicit *semantic augmentation (semanAug)* to address the two biases, respectively. On the one hand, we propose to fix the representation bias by learning generalizable and transferable representations. Specifically, we investigate several popular regularization methods empirically, and present a simple technique called *classAug*, to facilitate the representation learning for incremental learning. On the other hand, to overcome the classifier bias, *semanAug* implicitly involves the simultaneous generating of an infinite number of features of old classes, which poses tighter constraints to maintain the decision boundary of previously learned classes. Without storing any old samples, our method can outperform non-exemplar based methods by large margins and perform comparably with representative data replay based approaches. Our code will be publicly available upon the publishing of this paper.

1 Introduction

Deep neural networks (DNNs) have enabled great success in diverse machine learning tasks, based on stationary, large-scale, computationally expensive and memory intensive training data [1, 2, 3]. Yet the need of the ability to acquire sequential experience in dynamic and open environments [4, 5, 6] poses a serious challenge to modern deep learning systems, which only perform well on homogenized, balanced and shuffled data [7]. Typically, DNNs suffer from drastic performance degradation of previously learned tasks after learning new knowledge, which is a well-documented phenomenon, known as catastrophic forgetting [8, 9, 10]. Recently, incremental learning (IL), also referred to as lifelong learning or continual learning, has received extensive attention [11, 12, 13, 14] to enable DNNs to preserve and extend knowledge continually.

In this work, we consider a challenging setting of class-incremental learning (Class-IL), where the model only has access to data of new classes at each stage and needs to learn a unified classifier that can classify all seen classes [13, 15, 16]. Unfortunately, the learning paradigm of Class-IL will lead to two problems: ***representation bias*** and ***classifier bias***, as shown in Figure. 1. Firstly, for representation learning, if the feature extractor is fixed after learning the old task, the model could maintain previously learned knowledge, but suffers from the lack of plasticity for new tasks; on the contrary, if we update the feature extractor on a new task, the old knowledge could be easily forgotten. We denote this as the representation bias in Class-IL. Secondly, to distinguish new classes from old classes, the training loss is typically calculated on all classes. However, the training data

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

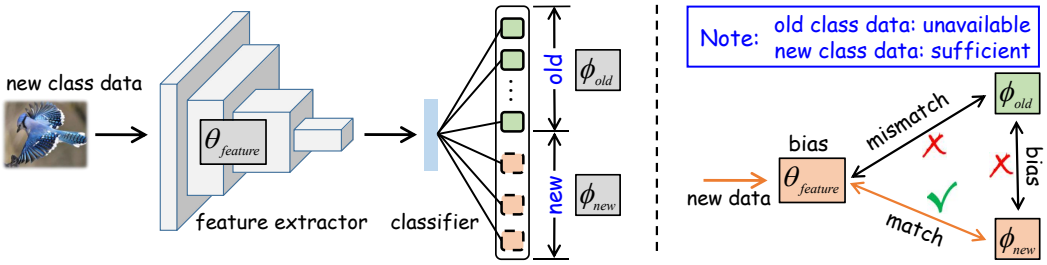

Figure 1: Two inherent problems in Class-IL: representation bias and classifier bias.

for old classes are unavailable and new classes usually have sufficient data. In this case, the classifier weights of old classes would be overly punished and the classifier will be severely biased towards new classes. In this work, we investigate the learning of representation and classifier in incremental learning, and propose a simple and effective dual augmentation framework to overcome these two biases for non-exemplar based (i.e., without storing and replaying data for old classes) Class-IL.

Learning Representation for Incremental Learning. Existing works typically regularize network parameters explicitly [11, 17, 18] or implicitly [12] to reduce the representation shift when learning new classes. However, such regularization would lead to a trade-off between the plasticity and stability [5]. We hypothesize that learning diverse and transferable representation is an important requirement for incremental learning, which has been ignored by previous works. Intuitively, with such representations, it could be easier to find a model to perform well on all tasks and improve both plasticity and stability, since different tasks would be closer in the parameters space. Motivated by the success of several regularization techniques (e.g., Mixup [19], Cutmix [20]) on improving the generalization of DNNs, we investigate whether Class-IL can be benefited from those techniques. We empirically find a surprising pathology: those widely proven regularization methods have negative effect on incremental learning. To understand this phenomenon, a natural question arises:

What properties of learned representations could facilitate incremental learning?

From a spectral analysis viewpoint, we investigate which components of feature representations are more transferable and less forgettable in the incremental learning process. It is found that spectral components with large singular values are less forgettable, which can explain the performance degradation of the above regularizations for incremental learning. Furthermore, we exploit this finding to propose a simple modification of mixup, named *classAug*, which can enlarge the spectral components to introduce more diverse and transferable representations for incremental learning.

Learning Classifier for Incremental Learning. Recently, several works were proposed to alleviate the classifier bias in data replay based methods [16, 21, 22]. However, in non-exemplar based Class-IL setting, the classifier bias is more serious and the above methods can not be directly used. A straightforward way is to storing features of old classes. However, this strategy is undesirable due to the limited memory resource and scalability. This work delves into the classifier learning for Class-IL problems, and proposes an implicit semantic augmentation (*semanAug*) approach to *maintain the decision boundary for previous classes by generating an infinite number of features of old classes*. SemanAug is inspired by MCF [23] and ISDA [24], which have performed semantic augmentation for linear models and DNNs, respectively. However, both our way to leverage semantic augmentation and the motivation fundamentally differ from them [23, 24].

Contributions. **(i)** We provide new insights into the representation learning in incremental learning by analyzing the structural characteristics of the learned embedding space via spectral decomposition, and find that spectral components with large singular values are less forgettable and carry more transferable features. Based on this observation, we propose a simple and effective method of classAug to learning better embedding space for incremental learning. **(ii)** For classifier learning in incremental learning, we propose semanAug which implicitly involves the simultaneous generating an infinite number of features of old classes to maintain the decision boundary of previously learned classes. **(iii)** Extensive experiments on benchmark datasets demonstrate the superior performance of our dual augmentation framework in the setting of class-incremental learning.

2 Related Work

Incremental Learning. Diverse approaches have been proposed for incremental learning of DNNs. They can be roughly divided into three categories: regularization based, data replay based, and architecture based approaches. ***Regularization based methods*** focus on weight regularisation by estimating and preventing the important network weights from changing [11, 17, 18]. The difference among those methods is the way to compute the importance of the parameters. However, it is hard to design a reasonable metric to measure the importance of parameters, and it is known that regularization strategies show poor performance in Class-IL scenario since the classifier bias is still severe [25, 26]. ***Data replay based methods*** address both the representation bias and classifier bias in a straightforward way by storing a fraction of old data to jointly train the model with current data. With stored real samples, some works [15, 13, 27] use a distillation loss to prevent forgetting, while others [28, 29, 30] develop gradient-based regularization to make more efficient use of the rehearsal data. To avoid storing real data, another line of works generates pseudo-samples of all previous classes for replay using deep generative models [31, 32, 33, 34]. Nevertheless, storing real data is undesirable for limited memory resources or privacy and safety concerns, and training big generative models for complex datasets is inefficient. ***Architecture based methods*** dynamically extend the network structure during the course of incremental learning [35, 36, 37, 38].

Data Augmentation. Literature is rich on data augmentation for improving the generalization of DNNs. Classical strategies commonly synthetic "positive" new samples in a way that is consistent with the underlying data distribution of the original dataset [3]. Recent works show that label mixing based methods such as Mixup [19] and Cutmix [20] can greatly improve the generalization of DNNs. In complementary to the ***input space augmentations*** mentioned above, some works have explored ***feature space augmentations*** which augment the learned representations in deep embedding space to enhance classifier performance. The intuition behind those works is that certain directions in the deep feature space correspond to meaningful semantic transformations [39, 40]. For instance, deep feature interpolation leverages simple interpolations in the embedding space to achieve semantic augmentation [40]. A recently proposed ISDA [24] performs semantic augmentation by estimating and leveraging the category-wise distribution of deep representations in an online manner. Despite the simplicity, ISDA shows its effectiveness in semi-supervised learning [24], contrastive learning [41], domain adaptation [42] and long-tailed recognition [43].

3 Dual Augmentation Framework for Class-Incremental Learning

We first introduce the class-incremental learning (Class-IL) problem definition, and then analyze the representation and classifier learning in Class-IL in the following subsection, respectively. With a clearer picture in mind, we design a novel dual augmentation framework for Class-IL.

Problem Definition. Typically, a Class-IL problem involves the sequential learning of $\mathcal{T}$ tasks that consist of disjoint classes sets, and the model has to learn a unified classifier that can classify all seen classes at any given point in training. At incremental step $t \in \{1, ..., \mathcal{T}\}$, training samples x and their corresponding ground truth labels $y \in C_t$ are drawn from an i.i.d. distribution $\mathcal{D}_t$, where C_t is the class set of task t. To facilitate analysis, we represent the DNNs based model with two components: a feature extractor f_θ and a unified classifier g_ϕ. The general objective is to minimize a predefined loss function $\mathcal{L}$ (e.g., cross-extropy) of all observed tasks up to the current one $t \in \{1, ..., t_c\}$:

$$\underset{\theta,\phi}{\operatorname{argmin}} \sum_{t=1}^{t_c} \mathcal{L}_t, \quad \text{where } \mathcal{L}_t \triangleq \mathbb{E}_{(x,y)\backsim\mathcal{D}_t}[l(y, g_\phi(f_\theta(x))]. \tag{1}$$

The key challenge of non-exemplar based Class-IL is that data from previous tasks are assumed to be unavailable, which means that the best configuration of the model for all seen tasks must be sought by minimizing $\mathcal{L}_t$ on current data $\mathcal{D}_t$. An effective and widely used way to preserve old knowledge is knowledge distillation [44], which typically matches the current model with previous model response to current training data using the teacher-student framework.

3.1 Learning Representation with Class Augmentation

As we focus on non-exemplar based Class-IL, we intentionally avoid storing old data. To maintain the generalizability of the learned representations for old classes, existing methods typically restrain the feature extractor from changing [11, 17, 18, 12]. However, this would lead to a trade-off between the plasticity and stability [5], and it would be hard to perform long-step incremental learning.

Table 1: Effect of several popular regularization techniques for Class-IL.

Method	iCaRL [13]						CCIL [45]					
classes	50	60	70	80	90	100	50	60	70	80	90	100
Baseline	**78.46**	**69.38**	**63.29**	**54.99**	**51.93**	**49.14**	**77.72**	**71.53**	**67.95**	**63.31**	**59.87**	**56.91**
Mixup	80.96	63.40	56.40	49.62	46.04	45.09	78.04	72.28	66.25	58.74	52.60	46.79
Cutmix	80.78	61.77	54.63	46.76	44.91	43.52	78.44	70.88	67.41	61.51	58.31	49.42
LS	79.76	59.58	53.83	46.52	45.49	43.63	78.56	62.06	57.50	51.50	47.43	44.78

Our high idea to learn diverse and transferable representation for incremental learning. Recently, several simple regularization techniques such as Mixup [19], Cutmix [20] and Label-Smoothing (LS) [46] have been verified to be helpful for improving generalization of DNN, we take a further step by investigating their performance on Class-IL. Concretely, we consider two baselines: (1) iCaRL [13] is a representative knowledge distillation based method, and (2) CCIL [45] is a state-of-the-art method that balances intra-task and inter-task learning. However, we find a surprising pathology: although those techniques are quite effective for stand supervised learning, they have negative effect on Class-IL problem, as shown in Table. 1. This simple experiment naturally rises an important and unanswered question: what types of representations could enable incremental learning? To delve into this problem, we conduct analysis to answer the following three question:

- Which part of feature representations tend to be forgotten in incremental learning?
- How the regularization techniques affect the components of feature representations?
- How to facilitate the representation learning for incremental learning?

3.1.1 Analyzing Forgetting via Spectral Decompositions

In what follows, we explore which part of feature representations $f(x)$ are tend to be forgetting and may not be transferable across different tasks in incremental learning. To this end, we propose to quantify the sensitivity of the model to different directions in deep feature space by measuring the similarity of the space before and after learning new tasks.

Formally, given a feature extractor $f_{\theta,old}$ trained on dataset $\mathcal{D}_{old} = \{(x_i, y_i)\}_{i=1}^{n}$. A new dataset $\mathcal{D}_{new}$ that contains disjoint classes with $\mathcal{D}_{old}$ is used to update $f_{\theta,old}$, and the updated feature extractor is donated as $f_{\theta,new}$. For the samples in $\mathcal{D}_{old}$, we can get two groups of deep features mapped by $f_{\theta,old}$ and $f_{\theta,new}$, respectively. Using eigenvalue decomposition, we could respectively decompose *the features mapped by original feature extractor (i.e., $f_{\theta,old}(x_i)$)* as well as *the features mapped by updated feature extractor (i.e., $f_{\theta,new}(x_i)$)* to different directions as following:

$$\frac{1}{n}\sum_{i=1}^{n} f_\theta(x_i) f_\theta(x_i)^{\mathrm{T}} = \sum_{j=1}^{d} u_j \lambda_j u_j^{\mathrm{T}}, \tag{2}$$

where λ_j represents the eigenvalues with index j and u_j is its eigenvector. d is the dimension of the feature space. Through spectral factorization in Eq. (2), we can represent the original and new feature extractors with two groups of eigenvectors: $\{u_{old,1}, ..., u_{old,d}\}$ and $\{u_{new,1}, ..., u_{new,d}\}$.

Next we investigate the forgetting or transferability of eigenvectors in each direction. Shonkwiler [47] introduced the principal angles [48] to measure the similarity of two subspaces. However, it is unreasonable to treate all eigenvectors equally to calculate the principal angles, regardless of their relative eigenvalues. Inspired by [49], we use corresponding angles, denoted by θ, to explore the distance between two subspaces in incremental learning:

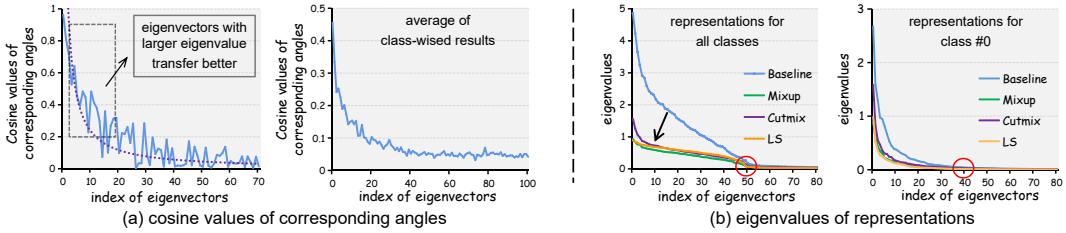

Figure 2: (a) Absolute cosine values of corresponding angles. (b) Distribution of eigenvalues for baseline and regularization based models.

Definition 1 **(Corresponding Angle)** Given two groups eigenvectors: $\{u_{old,1}, ..., u_{old,d}\}$ and $\{u_{new,1}, ..., u_{new,d}\}$, the angle between two eigenvectors corresponding to the same eigenvalue value index is defined as the cosine similarity between them:

$$\cos(\theta_j) = \frac{\langle u_{old,j}, u_{new,j} \rangle}{\|u_{old,j}\| \cdot \|u_{new,j}\|}, \tag{3}$$

where $u_{old,j}$ is the jth eigenvectors with the jth largest eigenvalue in the old feature space, and similarly for $u_{new,j}$. $\|u_{old,j}\| = 1$ and $\|u_{new,j}\| = 1$. Intuitively, the corresponding angle could capture the representation shift between the old and updated feature extractors during incremental learning, and reflect the forgetting along different directions in feature space.

Based on the metric defined above, we explore the forgetting of different directions in the setting of incremental learning. We use LwF-MC [13] as baseline methods and train ResNet-18 [1] on CIFAR-100 [50] in a 2-step manner. Concretely, the model is firstly trained ($f_{\theta,old}$) on the first 50 classes in CIFAR-100, and then updated ($f_{\theta,new}$) on the other 50 classes. Figure. 2 (a) shows the absolute cosine values of corresponding angles between the old and new eigenvectors. We can observe that eigenvectors with larger eigenvalues produce larger similarity (small corresponding angles), which indicates those directions are more transferable across different tasks. On the contrary, the eigenvectors with small eigenvalues prefer to move after updating the model on the new task, which could be regarded as forgettable directions. Therefore, it is important to enlarge the eigenvalues of eigenvectors to reduce forgetting and enhance the transferability of the representations.

3.1.2 Understanding Regularization Techniques for Incremental Learning

The experiment above indicates that eigenvectors with smaller eigenvalues suffer more forgetting when learning new tasks. With this in mind, we interpret the negative effect of those regularizations by investigating how they affect the components of feature representations. We draw the eigenvalues of covariance in Figure. 2 (b), showing that those regularizations decrease the eigenvalues of the corresponding top eigenvectors compared with standard training (baseline). Consequently, the transferability of the representations would be punished, and the old representations are more like to be forgotten when updating the model on new task. We note that simimlar negative effect of Mixup and LS for incremental learning has been noticed by [45], but without explaining the reason.

Feature Compression Perspective. As shown in Figure. 2 (b), although the output dimension of the model is large, the (intrinsic) dimension of the overall features is only 50 (left), and the (intrinsic) dimension of learned features for each class is around 40 (right), indicating that the variability and structural information in feature space are completely suppressed [51, 52]. Reducing the number of directions with significant variance has been seen as a form of *feature compression* [53], which is linked to generalization by information theory [54, 55]. However, the usual concepts of generalization in common single-task learning scenario may not entirely be appropriate for incremental learning, since standard learning only aims to learn compact representations within training classes without considering new class generalizability. In incremental learning, those less discriminative directions for the current task could capture useful representations for future tasks. For instance, considering a simple example of binary classification between sofa and chair, the classes can be separated by only learning the most discriminative characteristics (e.g., leg). However, the model would be confused about the desk, which is not a chair but also has legs. The intuition behind

this simple example indicates that learning compact or compressed representations for current task might degrade important directions for future tasks, e.g., the direction of eigenvectors with large eigenvalues. Indeed, a very nice recent paper [56] has shown that strong compressed representations can actually hurt the generalization ability in the deep metric learning setting. Our experiments in Table. 1 demonstrate similar phenomenon for incremental learning scenario.

3.1.3 Learning Transferable and Diverse Representations via Class Augmentation

We now exploit our above analysis to propose a simple method for incremental learning by counteracting the compression of the feature space. Our key idea is to learn less compact representations in each task. To this end, we propose ***class augmentation (classAug)*** to augment the original classes at each training stage t. Concretely, classAug randomly interpolates two samples x_a and x_b from two different classes a and b to generate a new sample x_{ab}^{new} representing a new class between these two classes:

$$x_{ab}^{\text{new}} = \lambda x_a + (1-\lambda) x_b, \tag{4}$$

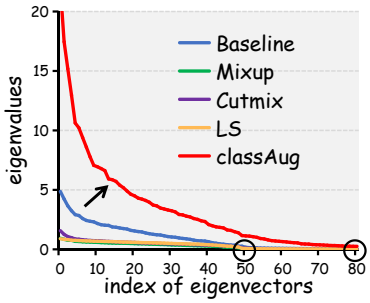

Figure 3: Eigenvalues distribution.

where λ is a *random* number of interpolation coefficient. For a k-class problem, we can generate $k(k-1)/2$ new classes using the above method, which can be further merged to m auxiliary classes. As a result, the original k-class problem in current task is therefore extended to a $(k+m)$-class problem. Moreover, we restrict the λ to be sampled from the interval of $[0.4, 0.6]$, to reduce the overlap between the augmented and original classes. Since the data of augmented classes are mixed from real classes, classAug would push samples apart from its mixed one, which can make the feature less compact. Simultaneously, by forcing the model to classify more classes in each stage t, more diverse representations could be learned. Figure. 3 displays and compares the eigenvalues of features learned with different methods. It is obvious that the proposed classAug can enhance the value of eigenvalues significantly, reducing the forgetting of the representation.

The proposed classAug is related to the *Mixup* [19] which applies random interpolation on a pair of training samples and the respective one-hot labels. However, the interpolated samples in Mixup are near original data, and the number of classes is not changed, but in our method, it is increased. Particularly, it has been found that the (intrinsic) dimension of features learned by standard cross-entropy loss has a strong correlation with the number of training classes [51]. Figure. 3 confirms this conclusion by showing that classAug can extend the (intrinsic) dimension of the overall features from 50 to 80, improving the diversity of the representations. As shown in Section 5, this simple technique can improve the performance of Class-IL significantly.

3.2 Learning Classifier with Semantic Augmentation

As shown in Figure. 1 and demonstrated in introduciton Section, ***classifier bias*** is another problem in Class-IL. When learning new classes, the previously learned decision boundary would suffer from catastrophic distortion, thus the features of old tast samples could be easily mapped to wrong class. To overcome this issue, our high level idea to leverage the distribution information (i.e., class mean and covariance) of old classes to regularize the learning of the classifier. Formally, for each old class $k \in \{1, ..., C_{old}\}$, we can generate M deep features from its distribution, i.e., $f_k^m \backsim \mathcal{N}(\mu_k, \Sigma_k)$. Then the generated features of old classes and real features of new classes can be jointly fed to the classifier for minimizing cross-entropy loss:

$$\mathcal{L}_t = \underbrace{\frac{1}{n_t}\sum_{i=1}^{n_t} -\log\left(\frac{e^{w_{y_i}^{\mathrm{T}} f_i + b_{y_i}}}{\sum_{c=1}^{C_{all}} e^{w_c^{\mathrm{T}} f_i + b_c}}\right)}_{\mathcal{L}_{t,new}\text{: loss on real features of new classes}} + \underbrace{\frac{1}{C_{old}}\sum_{k=1}^{C_{old}}\frac{1}{M}\sum_{m=1}^{M} -\log\left(\frac{e^{w_k^{\mathrm{T}} f_k^m + b_k}}{\sum_{c=1}^{C_{all}} e^{w_c^{\mathrm{T}} f_k^m + b_c}}\right)}_{\mathcal{L}_{t,old}\text{: loss on generated features of old classes}}, \tag{5}$$

where n_t is the number of training samples in current task dataset $\mathcal{D}_t$, C_{old} is the number of total old classes upon stage t, and $C_{all} = C_{old} + C_t$ is the number of all seen classes at stage t. $W =$

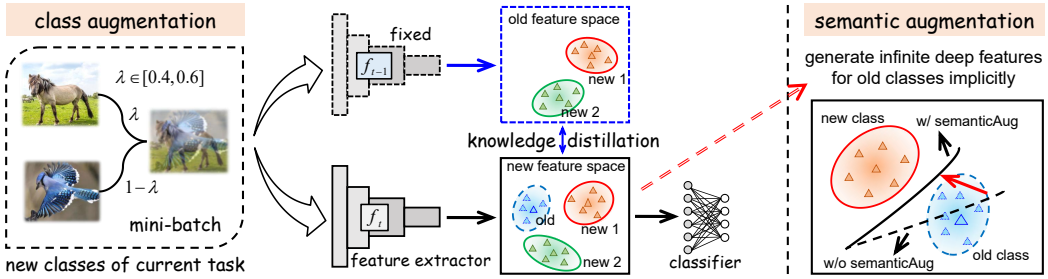

Figure 4: Illustration of our dual augmentation framework (IL2A) for Class-IL. On the one hand, the training samples of new classes at current task are augmented via the proposed classAug. On the other hans, the distribution of old classes are retained by semanAug in the deep feature space.

$[w_1, ..., w_{C_{all}}]^{\mathrm{T}} \in \mathcal{R}^{C_{all} \times d}$ and $b = [b_1, ..., b_{C_{all}}]^{\mathrm{T}} \in \mathcal{R}^{C_{all}}$ are the weight matrix and bias vector of the last fully connected layer, respectively.

In Class-IL, the sencond term in eq. (5), $\mathcal{L}_{t,old}$, is computationally inefficient when M and C_{old} are large. In the following, we intend to find an easy-to-compute way to implicitly generate infinite features for old classes. Concretely, in the case of $M \to \infty$, the sencond term in eq. (5):

$$\begin{aligned}
\mathcal{L}_{t,old} &= \frac{1}{C_{old}} \sum_{k=1}^{C_{old}} \mathbb{E}_{f_k} \left[-\log \left(\frac{e^{w_k^{\mathrm{T}} f_k^m + b_k}}{\sum_{c=1}^{C_{all}} e^{w_c^{\mathrm{T}} f_k^m + b_c}} \right) \right] = \frac{1}{C_{old}} \sum_{k=1}^{C_{old}} \mathbb{E}_{f_k} \left[\log \left(\sum_{c=1}^{C_{all}} e^{(w_c^{\mathrm{T}} - w_k^{\mathrm{T}}) f_k + (b_c - b_k)} \right) \right] \\
&\leqslant \frac{1}{C_{old}} \sum_{k=1}^{C_{old}} \log \left(\mathbb{E}_{f_k} \left[\sum_{c=1}^{C_{all}} e^{(w_c^{\mathrm{T}} - w_k^{\mathrm{T}}) f_k + (b_c - b_k)} \right] \right) \\
&= \frac{1}{C_{old}} \sum_{k=1}^{C_{old}} \log \left(\sum_{c=1}^{C_{all}} e^{v_{c,k}^{\mathrm{T}} f_k + (b_c - b_k) + \frac{1}{2} v_{c,k}^{\mathrm{T}} \Sigma_k v_{c,k}} \right) \triangleq \mathcal{L}_{t,semanAug}.
\end{aligned} \tag{6}$$

In above eqution, $v_{c,k} = w_c - w_k$. The inequality is based on Jensen's inequality $\mathbb{E}[\log(X)] \leqslant \log\mathbb{E}[X]$, and the last equality is obtained by using the monment-generating function $\mathbb{E}[e^{aX}] = e^{t\mu + \frac{1}{2}\sigma^2 a^2}$, $X \backsim \mathcal{N}(\mu, \sigma^2)$, due to the fact that $(w_c - w_k) f_k + (b_c - b_k)$ is a Gaussian random variable. As can be seen, eq. (6) is an upper bound of original $\mathcal{L}_{t,old}$, which provides an elegant and much efficient way to implicitly generate infinite features for old classes. In practice, we view μ_k as f_k, that is, $\mathcal{L}_{t,semanAug}$ implicitly performs semantic transformations for μ_k based on Σ_k.

Discussion. Although the derivation of the upper bound in eq. (6) is similar with ISDA [24], our motivation is to maintain the decision boundary of previously learned classes in Class-IL. When learning new classes, we only *apply semanAug for old classes* based on the memorized distribution information of old classes. While ISDA completely focuses on single-task learning setting. In addition, and a crucial step in ISDA is to estimate the mean and covariance of each class in an online manner. Differently, Class-IL is naturally suitable for applying of semanAug, since the distribution of old classes can be estimated with all training samples at the end of each learning stage.

3.3 The Dual Augmentation Learning Framwork

With classAug for representation bias and semanAug for classifier bias, Figure. 4 discribes the learning process of the dual augmentation framework (**IL2A**). Note that we also used the well-known knowledge distillation (KD) [21] in our method for two reasons. Firstly, classAug and KD are complementary and foucs on differnt aspect of learning representation. Secondly, KD can reduce the change of feature extractor, which is crucial for semanAug because it implicitly generate features from old distribution. The total learning objective at each stage t is as following:

$$\mathcal{L}_t = \mathcal{L}_{t,new} + \alpha \mathcal{L}_{t,semanAug} + \beta \mathcal{L}_{t,kd}, \tag{7}$$

where α and β are two hyper-parameters balancing the trade-off of each terms. $\mathcal{L}_{t,new}$ and $\mathcal{L}_{t,semanAug}$ are shown in eq. (5) and eq. (6), respectively. $\mathcal{L}_{t,kd} = \frac{1}{n_t} \sum_{i=1}^{n_t} \| f_{\theta_{t-1}}(x_i) - f_{\theta_t}(x_i) \|$.

4 Experiments

Datasets. We perform our experiments on challenging incremental learning datasets: CIFAR-100 [50] and Tiny-ImageNet [57]. A common setting is to train the model on half of classes for first task, and equal classes in the rest incremental step. Based on this, we split the CIFAR-100 dataset in different settings: $50 + \boldsymbol{5} \times 10$, $50 + \boldsymbol{10} \times 5$, $40 + \boldsymbol{20} \times 3$. For instance, $50 + \boldsymbol{10} \times 5$ that the first task contains 50 classes and there are 5 classes for the following 10 tasks. Similarly, the settings for Tiny-ImageNet are $100 + \boldsymbol{5} \times 20$, $100 + \boldsymbol{10} \times 10$ and $100 + \boldsymbol{20} \times 5$. Intuitively, more classes in each tasks requires the model to learn a harder problem for each task, while increasing the length of the task sequence challenges the model's retention.

Implementation Details. In our experiments, we follow [58] to utilize the ResNet-18 [1] as our base architecture, and train it from scratch in each experiment. All models are trained using Adam [59] optimizer with an initial learning rate of 0.001 for 100 epochs with the mini-batch size of 64. The learning rate is reduced by a factor of 10 at 45 and 90 epochs. We use the same hyperparameter value for all experiments. Specifically, we set $\alpha = 10$ and $\beta = 10$ in eq. (7). At the end of each incremental stage, we first remove the weights of the m augmented classes in the last fully connected layer, and then test the model on all seen classes. All source code will be publicly available.

Comparison Methods. Our method (IL2A) is non-exemplar based, and does not store any old samples for replay when learning new classes. Therefore, we first compare IL2A with several popular non-exemplar based approaches: MAS [18], LwF-MC [13], MUC [60], LwM [61]. In addition, we also compare with several exemplar based methods such as iCaRL [13], EEIL [16] and LUCIR [21]. Specifically, for the data replay based methods, we follow [13, 21] to store 20 samples for each class using 'herd' selection technique. We report the average top-1 accuracy of all previously seen tasks up to each incremental step t. For iCaRL, we respectively report its results of CNN predictions and nearest-mean-of-exemplars classification, denoted as iCaRL-CNN and iCaRL-NME.

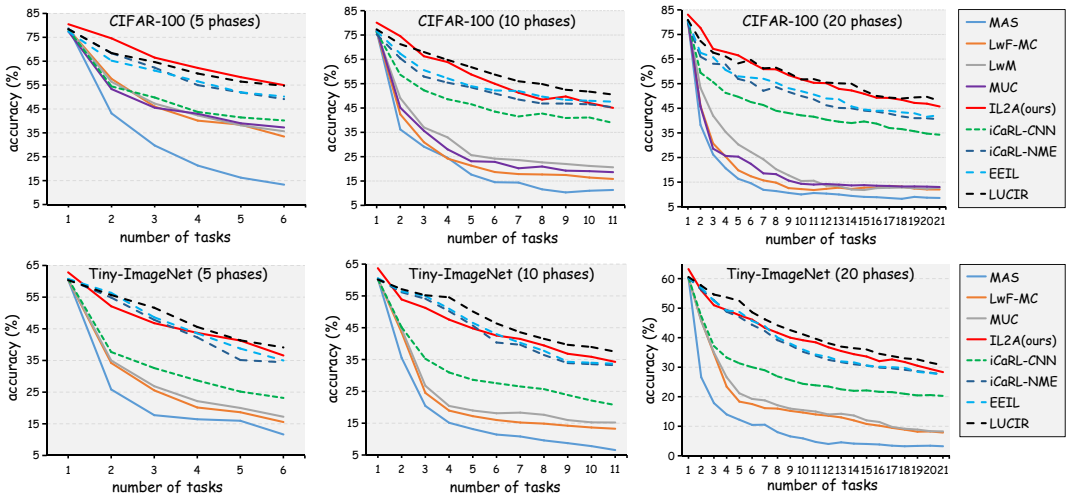

Figure 5: Results of top-1 accuracy on CIFAR-100 and Tiny-ImageNet under different settings.

Performance Comparison. Comparative results are shown in Figure. 5. Firstly, we observe that our method performs much better than non-exemplar based methods such as MAS, LwF-MC and MUC in the trend of accuracy curve under different settings. Particularly, the gap appears unbridgeable in long-step Class-IL setting, e.g., 10 incremental and 20 incremental steps. This suggesting that only regularization of old sets of parameters does not suffice to prevent forgetting. We argue that this is due to the unaddressed classifier bais. When compared to representative data replay based methods such as iCaRL, EEIL and LUCIR, our method remarkably shows comparable performance without storing old samples. The success of our method can contribute to the proposed classAug and semanAug. Specifically, classAug is applied to new classes of current task, which enables the

model to learn more diverse and transferable representations for further classes and in turn, reduces the forgetting of old parameters when learning new classes. While semanAug is applied to old classes of previous task, which leverage the valuable distribution information of old classes to learn a unified classifier that capable of relating the classes from different tasks to each other.

Ablation Study. To evaluate the effect of each component in IL2A, we perform the ablation study, and show the results of 10 incremental-tasks setting (CIFAR-100) in Table. 2. Specifically, the baseline denotes the method that using knowledge distillation to regularize the feature extractor and use the class mean for each old classes in the penultimate layer to regularize the classifier. Intuitively, the deep feature class mean is the primary information of its distribution, and is also the base of the semanAug which further enriches the old knowledge by leveraging the covariance information. In summary, we can observe that: (1) baseline improves the performance of KD significantly. (2) SemanAug improves the performance of baseline from 34.71% to 42.09%. Those results indicate the effect of the distribution information for maintaining old knowledge in Class-IL. (3) ClassAug also has remarkably effect on baseline, and (4) the performance can be further improved by combing with semanAug, which indicates that those two modules are complementary. Similar results are observed in other settings of CIFAR-100 and Tiny-ImageNet dataset.

Table 2: The effect of each component in IL2A.

Method\Incremental stage	1	2	3	4	5	6	7	8	9	10	Final
Knowledge Distillation	78.78	30.18	20.71	14.61	11.87	8.80	7.70	7.23	7.10	6.05	6.04
Baseline	78.86	62.85	56.96	54.66	51.72	47.33	43.61	40.12	40.76	36.55	34.71
+ semanAug	79.16	69.14	60.68	58.18	54.77	50.89	48.45	46.29	46.97	44.38	42.09
+ classAug	79.72	68.30	64.15	60.15	56.21	52.61	51.48	46.48	46.36	43.63	41.56
+ classAug + semanAug	**81.08**	**74.54**	**66.28**	**63.89**	**58.80**	**54.97**	**51.32**	**48.64**	**49.74**	**47.05**	**45.07**

Further Analysis. To analyze the effectiveness of classAug more concretely, we explore how it affects the new tasks accuracy (↑) and average forgetting (↓). Intuitively, new task accuracy can be viewed as the plasticity of the incremental learner and the average forgetting can be viewed as the stability of the incremental learner. Figure. 6 reports the results, from which we clearly see that classAug simultaneously improving the new task accuracy and reducing the average forgetting. Specifically, the significant improvement on new task accuracy implies that the model training with classAug is a good initialization for the next task. Consequently, classAug is effective to improve the trade-off between plasticity and stability of the model.

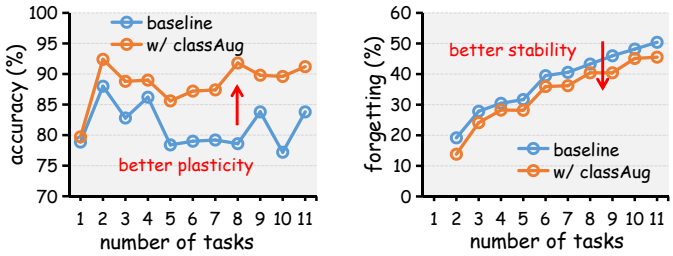

Figure 6: ClassAug can simultaneously improve the new task accuracy and reduce the average forgetting.

5 Conclusion

In this paper, we propose a simple and effective dual augmentation framework to address the representation bias and classifier bias in class-incremental learning. We first investigate the transferability (or forgetting) of features in incremental learning via spectral decomposition, which motivates us to propose classAug that can learn transferable, diverse and less compact representations for incremental learning. Furthermore, we propose to use semanAug for leveraging the distribution of old classes to avoid forgetting in an elegant way that can implicitly generate infinite features of old classes during jointly learning of the unified classifier. Experiments show that our method could achieve remarkable performance gains compared with the state-of-the-art class-incremental learning methods. Future works will consider the dual augmentation framework for more tasks like few-shot learning, domain generalization, and so on.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 5, 8

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012. 1, 3

[4] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015. 1

[5] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. 1, 2, 4

[6] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 1

[7] Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 2020. 1

[8] Ian J. Goodfellow, M. Mirza, Xia Da, Aaron C. Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. *CoRR*, 2014. 1

[9] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, pages 109–165, 1989. 1

[10] Robert M French. Interactive tandem networks and the sequential learning problem. Citeseer. 1

[11] J. Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, J. Veness, G. Desjardins, Andrei A. Rusu, K. Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, C. Clopath, D. Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, pages 3521 – 3526, 2017. 1, 2, 3, 4

[12] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, pages 2935–2947, 2018. 1, 2, 4

[13] Sylvestre-Alvise Rebuffi, A. Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017. 1, 3, 4, 5, 8

[14] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *ECCV*, pages 524–540, 2020. 1

[15] Y. Wu, Yan-Jia Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019. 1, 3

[16] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, pages 233–248, 2018. 1, 2, 8

[17] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *ICML*, pages 3987–3995, 2017. 2, 3, 4

[18] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 2, 3, 4, 8

[19] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 4, 6

[20] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 2, 3, 4

[21] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and D. Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, pages 831–839, 2019. 2, 7, 8

[22] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13205–13214, 2020. 2

[23] Laurens Maaten, Minmin Chen, Stephen Tyree, and Kilian Weinberger. Learning with marginalized corrupted features. In *ICML*, pages 410–418, 2013. 2

[24] Yulin Wang, Gao Huang, Shiji Song, Xuran Pan, Yitong Xia, and Cheng Wu. Regularizing deep networks with semantic data augmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2021. 2, 3, 7

[25] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 3

[26] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 3

[27] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, pages 86–102, 2020. 3

[28] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. In *ICLR*, 2018. 3

[29] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *NeurIPS*, 2017. 3

[30] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019. 3

[31] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, pages 2994–3003, 2017. 3

[32] Chenshen Wu, L. Herranz, X. Liu, Y. Wang, Joost van de Weijer, and B. Raducanu. Memory replay gans: Learning to generate new categories without forgetting. In *NeurIPS*, pages 5962–5972, 2018. 3

[33] Ye Xiang, Ying Fu, Pan Ji, and Hua Huang. Incremental learning using conditional adversarial networks. In *ICCV*, pages 6618–6627, 2019. 3

[34] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. In *ICLR*, 2018. 3

[35] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 3

[36] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 3

[37] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, pages 4548–4557, 2018. 3

[38] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *ICLR*, 2018. 3

[39] Yoshua Bengio, Grégoire Mesnil, Yann Dauphin, and Salah Rifai. Better mixing via deep representations. In *ICML*, pages 552–560, 2013. 3

[40] Paul Upchurch, Jacob Gardner, Geoff Pleiss, Robert Pless, Noah Snavely, Kavita Bala, and Kilian Weinberger. Deep feature interpolation for image content changes. In *CVPR*, pages 7064–7073, 2017. 3

[41] Qi Cai, Yu Wang, Yingwei Pan, Ting Yao, and Tao Mei. Joint contrastive learning with infinite possibilities. In *NeurIPS*, 2020. 3

[42] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. *arXiv preprint arXiv:2103.12562*, 2021. 3

[43] Shuang Li, Kaixiong Gong, Chi Harold Liu, Yulin Wang, Feng Qiao, and Xinjing Cheng. Metasaug: Meta semantic augmentation for long-tailed visual recognition. *arXiv preprint arXiv:2103.12579*, 2021. 3

[44] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3

[45] Sudhanshu Mittal, Silvio Galesso, and Thomas Brox. Essentials for class incremental learning. *arXiv preprint arXiv:2102.09517*, 2021. 4, 5

[46] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *NeurIPS*, pages 4696–4705, 2019. 4

[47] Clayton Shonkwiler. Poincaré duality angles for riemannian manifolds with boundary. *arXiv preprint arXiv:0909.1967*, 2009. 4

[48] Jianming Miao and Adi Ben-Israel. On principal angles between subspaces in rn. *Linear algebra and its applications*, 171:81–98, 1992. 4

[49] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *ICML*, pages 1081–1090, 2019. 4

[50] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, 2009. 5, 8

[51] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. 5, 6

[52] Dustin G Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *arXiv preprint arXiv:2011.11619*, 2020. 5

[53] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019. 5

[54] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5, 2015. 5

[55] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 5

[56] Karsten Roth, Timo Milbich, Samarth Sinha, Prateek Gupta, Björn Ommer, and Joseph Paul Cohen. Revisiting training strategies and generalization performance in deep metric learning. In *ICML*, 2020. 6

[57] Leon Yao and John Miller. Tiny imagenet classification with convolutional neural networks. *CS 231N*. 8

[58] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. In *NeurIPS*, 2020. 8

[59] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 8

[60] Yu Liu, Sarah Parisot, Gregory G. Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *ECCV*, pages 699–716, 2020. 8

[61] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *CVPR*, pages 5138–5146, 2019. 8

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]