Distilling Causal Effect of Data in Class-Incremental Learning

Xinting Hu[1], Kaihua Tang[1], Chunyan Miao[1], Xian-Sheng Hua[2], Hanwang Zhang[1]

[1]Nanyang Technological University, [2]Damo Academy, Alibaba Group

xinting001@e.ntu.edu.sg, kaihua001@e.ntu.edu.sg,

ascymiao@ntu.edu.sg, xiansheng.hxs@alibaba-inc.com, hanwangzhang@ntu.edu.sg

Abstract

We propose a causal framework to explain the catastrophic forgetting in Class-Incremental Learning (CIL) and then derive a novel distillation method that is orthogonal to the existing anti-forgetting techniques, such as data replay and feature/label distillation. We first 1) place CIL into the framework, 2) answer why the forgetting happens: the causal effect of the old data is lost in new training, and then 3) explain how the existing techniques mitigate it: they bring the causal effect back. Based on the causal framework, we propose to distill the Colliding Effect between the old and the new data, which is fundamentally equivalent to the causal effect of data replay, but without any cost of replay storage. Thanks to the causal effect analysis, we can further capture the Incremental Momentum Effect of the data stream, removing which can help to retain the old effect overwhelmed by the new data effect, and thus alleviate the forgetting of the old class in testing. Extensive experiments on three CIL benchmarks: CIFAR-100, ImageNet-Sub&Full, show that the proposed causal effect distillation can improve various state-of-the-art CIL methods by a large margin (0.72%–9.06%). [1]

1. Introduction

Any learning systems are expected to be adaptive to the ever-changing environment. In most practical scenarios, the training data are streamed and thus the systems cannot store all the learning history: for animate systems, the neurons are genetically designed to forget old stimuli to save energy [6, 34]; for machines, old data and parameters are discarded due to limited storage, computational power, and bandwidth [28, 44, 50]. In particular, a practical learning task as Class-Incremental Learning[1] (CIL) [1, 13, 35, 47] is studied, where each incremental data batch consists of new samples and their corresponding new class labels. The

(a) Forgetting in Class-Incremental Learning

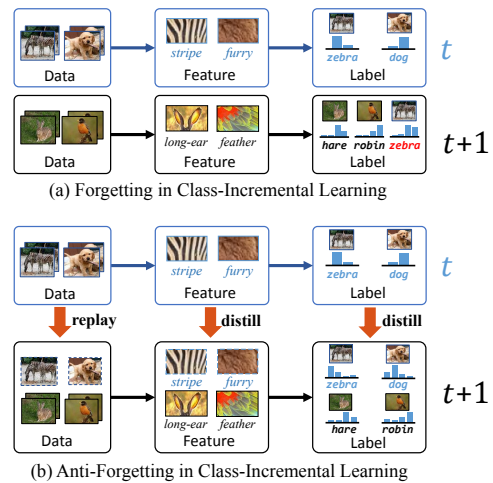(b) Anti-Forgetting in Class-Incremental Learning

Figure 1: Forgetting and existing anti-forgetting solutions in Class-Incremental Learning at step t and $t + 1$. (a): Forgetting happens when the features are overridden by the $t + 1$ step training only on the new data. (b): The key to mitigate forgetting is to impose data, feature, and label effects from step t on step $t + 1$.

key challenge of CIL is that discarding old data is especially catastrophic for deep-learning models, as they are in the data-driven and end-to-end representation learning nature. The network parameters tuned by the old data will be overridden by the SGD optimizer using new data [11, 33], causing a drastic performance drop — catastrophic forgetting [36, 10, 28] — on the old classes.

We illustrate how CIL forgets old classes in Figure 1 (a). At step t, the model learns to classify *"dog"* and *"zebra"*; at the next step $t+1$, it updates with new classes *"hare"* and *"robin"*. If we fine-tune the model only on the new data, old features like "stripe" and "furry" will be inevitably overridden by new features like "long-ear" and "feather", which are discriminative enough for the new classes. Therefore, the new model will lose the discriminative power — forget — about the old classes (*e.g.*, the red *"zebra"* at step $t + 1$).

To mitigate the forgetting, as illustrated in Figure 1 (b), existing efforts follow the three lines: 1) data replay [35, 13, 25], 2) feature distillation [13, 9], and 3) la-

---

[1]Code is available at https://github.com/JoyHuYY1412/DDE_CIL

[1]There are also other settings like task-incremental [14, 26].

bel distillation [1, 47]. Their common nature is to impose the old training *effect* on the new training (the red arrows in Figure 1 (b), which are formally addressed in Section 3.3). To see this, for replay, the imposed effect can be viewed as including a small amount of old data (denoted as dashed borders) in the new data; for distillation, the effect is the features/logits extracted from the new data by using the old network, which is imposed on the new training by using the distillation loss, regulating that the new network behavior should not deviate too much from the old one.

However, the distillation is not always coherent with the new class learning — it will even play a negative role at chances [42, 43]. For example, if the new data are out-of-distribution compared to the old one, the features of the new data extracted from the old network will be elusive, and thus the corresponding distillation loss will mislead the new training [23, 20]. We believe that the underlying reason is that the feature/label distillation merely imposes its effect at the output (prediction) end, violating the end-to-end representation learning, which is however preserved by data replay. In fact, even though that the ever-increasing extra storage in data replay contradicts with the requirement of CIL, it is still the most reliable solution for anti-forgetting [35, 45, 5]. Therefore, we arrive at a dilemma: the end-to-end effects of data replay are better than the output-end effects of distillation, but the former requires extra storage while the latter does not. We raise a question: *is there a "replay-like" end-to-end effect distillation*?

In this paper, we positively answer this question by framing CIL into a causal inference framework [31, 29], which models the causalities among data, feature, and label at any consecutive t and $t+1$ learning steps (Section 3.1). Thanks to it, we can explain the reasons 1) why the forgetting happens: the *causal effect* from the old training is lost in the new training (Section 3.2), and 2) why data replay and feature/label distillation are anti-forgetting: they win the effect back (Section 3.3). Therefore, the above question can be easily reduced to a more concrete one: *besides replay, is there another way to introduce the causal effect of the old data?* Fortunately, the answer is yes, and we propose an effective yet simple method called: Distilling Colliding Effect (Section 4.1), where the desired end-to-end effect can be distilled without any replay storage. Beyond, we find that the imbalance between the new data causal effect (*e.g.*, hundreds of samples per new class) and the old one (*e.g.*, 5–20 samples per old class) causes severe model bias on the new classes. To this end, we propose an Incremental Momentum Effect Removal method to remove the biased data causal effect that causes forgetting (Section 4.2).

Through extensive experiments on CIFAR-100 [18] and ImageNet [7], we observe consistent performance boost by using our causal effect distillation, which is agnostic to methods, datasets, and backbones. For example, we

improve the two previously state-of-the-art methods: LU-CIR [13] and PODNet [9], by a large margin (0.72%–9.06%) on both benchmarks. In particular, we find that the improvement is especially larger when the replay of the old data is fewer. For example, the average performance gain is 16.81% without replay, 7.05% with only 5 samples per old class, and 1.31% with 20 samples per old class on CIFAR-100. The results demonstrate that our distillation indeed preserves the causal effect of data.

2. Related Work

Class Incremental Learning (CIL). Incremental learning aims to continuously learn by accumulating past knowledge [2, 19, 24]. Our work is conducted on CIL benchmarks, which need to learn a unified classifier that can recognize all the old and new classes combined. Existing works on tackling the forgetting challenge in CIL can be broadly divided into two branches: replay-based and distillation-based, considering the data and representation, respectively. *Replay-based* methods include a small percentage of old data in the new data. Some works [35, 1, 13, 47, 21] tried to select a representative set of samples from the old data, and others [25, 38, 15, 16] used synthesized exemplars to represent the distribution of the old data. *Distillation-based* methods combine the regularization loss with the standard classification loss to update the network. The regularization term is calculated between the old and new networks to preserve previous knowledge when learning new data [12, 22]. In practice, they enforce features [13, 9] or predicted label logits [1, 47, 25] of the new model to be close to those of the old model. *Our work* aims to find a "data-free" solution to distill the effect of old data without storing them. Our work is orthogonal to previous methods and brings consistent improvement to them.

Causal Inference [31, 37]. It has been recently introduced to various computer vision fields, including feature learning [3, 46], few-shot classification [49], long-tailed recognition [40], semantic segmentation [8] and other high-level vision tasks[41, 32, 48, 27]. Using causal inference in CIL can help formulate the causal effect of all the ingredients, point out the forgetting is all about the vanishing old data effect, and thus anti-forgetting is to retrieve it back.

3. (Anti-) Forgetting in Causal Views

3.1. Causal Graphs

To systematically explain the (anti-)forgetting in CIL in terms of the causal effect of the old knowledge, we frame the data, feature, and label in each incremental learning step into causal graphs [29] (Figure 2). The causal graph is a directed acyclic Bayesian graphical model, where the nodes denote the variables, and the directed edges denote the causality between two nodes.

Specifically, we denote the old data as D; the training samples in the new data as I; the extracted features from the current and old model as X and X_o, respectively; the predicted labels from the current and old model as Y and Y_o, respectively. The links in Figure 2 are as follows:

$I \rightarrow X \rightarrow Y$. $I \rightarrow X$ denotes that the feature X is extracted from the input image I using the model backbone, $X \rightarrow Y$ indicates that the label Y is predicted by using the feature X with the classifier.

$(D, I) \rightarrow X_o$ & $(D, X_o) \rightarrow Y_o$. Given the new input image I, using the old model trained on old data D, we can get the feature representation X_o in the old feature space. Similarly, the predicted logits Y_o in the old class vocabulary can be obtained by using the feature representation X_o and old classifier trained on D. These two causal sub-graphs denote the process of interpreting each new image I using the previous model trained on old data D.

$D \rightarrow I$. As the data replay strategy stores the old data and mixes them with the new, this link represents the process of borrowing a subset of the old data from D, and thus affecting the new input images I.

$X_o \rightarrow X$ & $Y_o \rightarrow Y$. The old feature representation X_o (old logits Y_o) regularizes the new feature X (new logits Y) by the feature (logits) distillation loss.

$X_o \nrightarrow X$. Though the new model is initialized from the old model, the effect of initial parameters adopted from the old model will be exponentially decayed towards 0 during learning [17], and thus is neglected in this case.

3.2. Forgetting

Based on the graph, we can formulate the causal effect between variables by *causal intervention*, which is denoted as the $do(\cdot)$ operation [30, 31]. $do(A = a)$ is to forcibly and externally assign a certain value a to the variable A, which can be considered as a "surgery" on the graph by removing all the incoming links of A, making its value independent of its parents (causes). After the surgery, for any node pair A and B, we can estimate the effect of A on B as $P(B \mid do(A = a)) - P(B \mid do(A = 0))$, where $do(A = 0)$ denotes the null intervention, *e.g.*, the baseline compared to $A = a$.

To investigate why the forgetting happens in the new training process, we consider the difference between the predictions Y with and without the existence of old data D. For each incremental step in CIL, the effect of old data D on Y for each image is formulated as:

$$Effect_D = P(Y\!=\!y \mid do(D\!=\!d)) - P(Y\!=\!y \mid do(D\!=\!0))$$
$$= P(Y\!=\!y \mid D\!=\!d) - P(Y\!=\!y \mid D\!=\!0), \quad (1)$$

where $P(Y|do(D)) = P(Y|D)$ since D has no parents.

As observed in Figure 2 (a), all the causal paths from D to Y is blocked by colliders. Interested readers may jump to the first paragraph of Section 4 to know the collider property. For example, the path $D \rightarrow X_o \leftarrow I \rightarrow X \rightarrow Y$ is blocked by the collider X_o. Therefore, we have:

(a) The Forgetting of Class-Incremental Learning

(b) Data Replay   (c) Feature Distillation   (d) Label Distillation
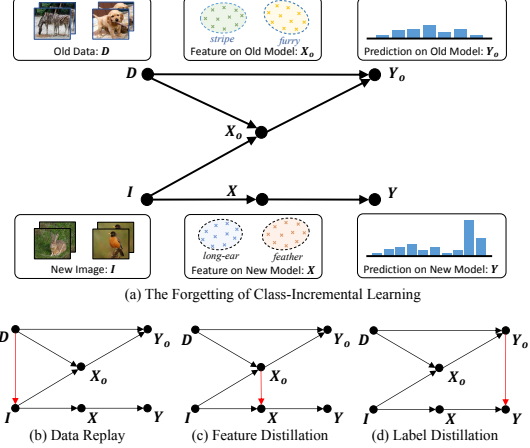
Figure 2: The proposed causal graphs explaining the forgetting and anti-forgetting in CIL. We illustrate the meaning of each node in the CIL framework in (a). The comparison between (a)-(d) shows the key to combating forgetting is the causal effect of old data.

$$Effect_D = P(Y\!=\!y \mid D\!=\!d) - P(Y\!=\!y \mid D\!=\!0)$$
$$= P(Y = y) - P(Y = y) = 0. \quad (2)$$

The zero effect indicates that D has no influences on Y in the new training — forgetting.

3.3. Anti-Forgetting

Now we calculate the causal effect $Effect_D$ when using existing anti-forgetting methods: data replay and feature/label distillation as follows:

Data Replay. Since the data replay strategy introduces a path from D to I as in Figure 2 (b), D and I is no longer independent with each other. By expanding Eq. (1), we have:

$$Effect_D = \sum_I P(Y \mid I, D = d) \, P(I \mid D = d)$$
$$- \sum_I P(Y \mid I, D = 0) \, P(I \mid D = 0)$$
$$= \sum_I P(Y \mid I) \, (P(I \mid D\!=\!d) - P(I \mid D\!=\!0)) \neq 0, \quad (3)$$

where $P(I \mid D\!=\!d) - P(I \mid D\!=\!0) \neq 0$ due to the fact that the replay changes the sample distribution of the new data, and $P(Y \mid I, D) = P(Y \mid I)$ as I is the only mediator from D to Y. It is worth noting that as the first term $P(Y \mid I, D)$ is calculated in an end-to-end fashion: from the data I to the label prediction Y, data replay takes the advantages of the end-to-end feature representation learning.

Feature & Label Distillation. Similarly, as the causal links $X_o \rightarrow X$ and $Y_o \rightarrow Y$ are introduced into the graph by using feature and logits distillation, the path $D \rightarrow X$ and $D \rightarrow Y$ are unblocked. Take the feature distillation for example, substituting X for I in Eq. (3), we have

$$Effect_D = \sum_X P(Y \mid X) \, (P(X \mid D\!=\!d) - P(X \mid D\!=\!0)) \neq 0, \quad (4)$$

3959

where $P(X \mid D=d) - P(X \mid D=0) \neq 0$ because the old network regularizes the new feature X via the distillation. Note that $P(Y \mid X)$ is not end-to-end as that in data replay.

Thanks to the causal graphs in Figure 2, we explain how the forgetting is mitigated by building unblocked causal paths between D and (I, X, Y), and then introduce the non-zero $Effect_D$. Most of the recent works in CIL can be considered as the combinations of these cases in Figure 2 (b)-(d). For example, iCaRL [35] is the combination of (b) and (d), where old exemplars are stored and the distillation loss is calculated from soft labels. LUCIR [13], which adopts the data replay and applies a distillation loss on the features while fixing old class classifiers, can be viewed as an integration of (b), (c), and (d).

4. Distilling Causal Effect of Data

Thanks to the causal views, we first introduce a novel distillation method that fundamentally equals to the causal effect of data replay, but without any cost of replay storage (Section 4.1 and Algorithm 1). Then, we remove the overwhelmed new data effect to achieve balanced prediction for the new and old classes (Section 4.2 and Algorithm 2).

Algorithm 1 One CIL step with Colliding Effect Distillation

1: **Input** : $\mathcal{I}, \Omega_o$ ▷ new training data, old model
2: **Output** : Ω ▷ new model
3:   $\Omega \leftarrow \Omega_o$ ▷ initialize new model
4:   $\mathcal{X}_o \leftarrow \Omega_o(\mathcal{I})$ ▷ represent new images in old features
5: **repeat** for $I \in \mathcal{I}$
6:     $\mathcal{N}_K \leftarrow$ K-NEAREST-NEIGHBOR$(\Omega_o(I), \mathcal{X}_o)$
7:     $P(Y \mid N_1), \dots, P(Y \mid N_K) \leftarrow \Omega(\mathcal{N}_K)$
8:     $W_1, \dots, W_K \leftarrow$ WEIGHTASSIGN(K) ▷ Eq. (7)
9:     $Effect \leftarrow \sum_{j=1}^{K} W_j \, P(Y \mid N_j; \, \Omega)$
10:     $\Omega \leftarrow \underset{\Omega}{\arg\min}(- \log(Effect))$
11: **until** converge

4.1. Distilling Colliding Effect

As we discussed in Eq. (3), to introduce the end-to-end data effect, the path between D and I is linked by replaying old data. However, the increasing storage contradicts the original intention of CIL [35]. Is there another way around to build the path between D and I without data replay?

Reviewing the causal graph in Figure 1 (a), we observe the silver lining behind the collider X_o: If we control the collider X_o, *i.e.*, condition on X_o, nodes D and I will be correlated with each other, as shown in Figure 3 (a). We call this *Colliding Effect* [31], which can be intuitively understood through a real-life example: a causal graph *Intelligence → Grades ← Effort* representing a student's grades are determined by both the intelligence and hard work. Usually, a student's intelligence is independent of whether she works hard or not. However, considering a student who got

---

Algorithm 2 CIL with Incremental Momentum Effect Removal

1: **Input** : $\mathcal{I}_1, \mathcal{I}_2, \dots, \mathcal{I}_T$ ▷ training data of T steps
2: **Input** : I ▷ a testing image
3: **For** $t \in 1, 2, \dots, T$ ▷ each CIL step
4:    $\alpha, \beta, h_t \leftarrow$ MOVINGAVERAGE$(\mathcal{I}_t; \, \Omega_t)$ ▷ training
5:    $h \leftarrow (1-\beta) \, h_{t-1} + \beta \, h_t$ ▷ head direction
6:    $X \leftarrow$ FEATUREEXTRACTOR(I) ▷ inference
7:    $Y \leftarrow$ CLASSIFIER$(X - x^h)$ ▷ Eq. (8)

---

good grades and knowing that she is not smart, it tells us immediately that she is most likely hard-working, and vice versa. Thus, two independent variables become dependent upon knowing the value of their joint outcome: the collider variable (*Grades*).

Based on Eq. (1), after conditioning on X_o, the effect of old data can be written as:

$$Effect_D = \sum_I P(Y \mid I, X_o) \, \underline{(P(I \mid X_o, D=d) - P(I \mid X_o, D=0))}$$
$$= \sum_I P(Y \mid I) \, W(I, X_o, D), \tag{5}$$

where $P(Y \mid I, X_o) = P(Y \mid I)$ since I is the only mediator from X_o to Y, and the underlined term can be denoted as a weight term $W(I, X_o, D)$. By calculating the classification $P(Y \mid I = i)$ and the weight $W(I = i, X_o, D)$, we can distill the colliding effect into the new learning step without old data replay.

Intuitively, the value of $W(I = i, X_o, D)$ (written as W below for simplicity) implies the likelihood of one image sample $I = i$ given $X_o = x_o$ extracted by model trained on $D = d$, that is, the larger value of W shows that the more likely the image is like i. Given all the new images represented by using the old network ($D = d$), suppose an image i^* whose old feature is x_o. According to the meaning of W, the value is larger for an image that is more similar to i^*. Since the similarity of images can be measured by feature distance, we can infer that the image whose old feature X_o is closer to x_o should have a larger W. For the new data containing n image, we can sort all the images according to the distance to x_o ascendingly. The sorted n images are indexed as $\{N_1, N_2, \dots, N_n\}$.

In summary, the rule of assigning W among these n images can be summarized as: $W_{N_1} \geq W_{N_2} \geq \dots \geq W_{N_n}$.

For implementation, we select a truncated top-K fragment and re-write Eq. (5) as:

$$Effect_D = \sum_{i \in \{N_1, N_2, \dots, N_K\}} P(Y \mid I = i) \, W_i, \tag{6}$$

where $\{W_{N_1}, \dots, W_{N_K}\}$ subject to

$$\begin{cases} W_{N_1} \geq W_{N_2} \geq \dots \geq W_{N_K} \\ W_{N_1} + W_{N_2} + \dots + W_{N_K} = 1 \, . \end{cases} \tag{7}$$
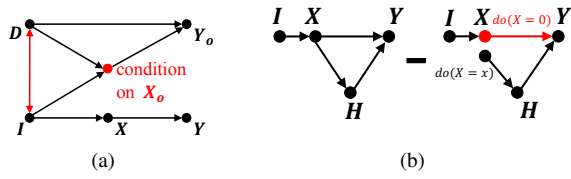
Figure 3: (a): The colliding effect in CIL. (b): The causal graph of removing incremental momentum effect.

It means that the classification probabilities are first calculated on each of the K images with the current model and then weighted averaged to obtain the final prediction. For more experiments on the choice of K and weight assignment, see Section 5.2.

Compared with data replay that builds $D \to I$, the merit of the above colliding effect distillation is that it can work with very limited or even without old data, which makes it not just much more efficient, but also applicable to large-vocabulary datasets, where each incremental step can only afford replaying very few samples for one class.

4.2. Incremental Momentum Effect Removal

Since the old data barely appear in the subsequent incremental steps, although we ensure the effect between D and Y to be *non-zero*, the moving averaged momentum μ in SGD optimizer [39] inevitably makes the prediction biased towards the new data, causing the CIL system suffers from the severe data imbalance. As shown in Figure 4, by visualizing the number of instances for classes in the CIL steps, we observe a clear long-tailed distribution, which is a challenging task.

To mitigate the long-tail bias, we follow [40] to pursue the direct causal effect of the feature X during inference, which removes a counterfactual biased head projection from the final logits. The causal graph of this approach is in Figure 3 (b), where the nodes H is the head direction caused by the SGD momentum. This causal graph is built on top of Figure 2 while only focuses on the current learning process. The subtracted term can be considered as the pure bias effect with a null input. However, the CIL cannot be simply treated as a classic long-tail classification problem because the data distributions across CIL steps are dynamic, *i.e.*, the current rare classes are the frequent classes in a previous step. Our experimental results (Figure 4) also reveal that: even the old classes have equally rare frequencies in an incremental step, the performance drop of a class increases with the time interval between the current step and the step when they are initially learned, which is not observed in the long-tail classification. It means that the momentum effect relies not only on the final momentum but also those in previous steps. Therefore, we propose an incremental momentum effect removing strategy.

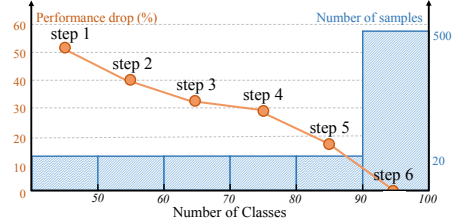For the incremental step t with total M SGD iterations,

Figure 4: The data distribution of CIFAR-100 [18] and the performance drop of learned classes at the final CIL step under the 5-step-20-replay setting.

by moving averaging the feature x_m in each iteration m using $x_m = \mu \cdot x_{m-1} + x_m$ with the SGD momentum μ, we can obtain a head direction h_t as the unit vector $x_M / \|x_M\|$ of the final x_M. As the moving average of all the samples trained in step t, h_t shows the momentum effect of new model. Unlike [40] where the h_t is directly adopted, in order to involve the momentum effect of old model, we design a dynamic head direction $h = (1-\beta)\, h_{t-1} + \beta\, h_t$, combining both the head direction of step t and the last step $t-1$. In inference, as illustrated in Algorithm 2, for each image i, we can get the de-biased prediction logits $[Y \mid I = i]$ by removing the proposed incremental momentum effect as:

$$[Y \mid I = i] = [Y \mid do(X{=}x)] - \alpha \cdot [Y \mid do(X{=}0, H{=}h)]$$
$$= [Y \mid X{=}x] - \alpha \cdot [Y \mid X{=}x^h], \qquad (8)$$

where x^h means the projection of feature x on the dynamic head direction h, and both trade-off parameters α and β are learned automatically in the back-propagation.

5. Experiment

5.1. Settings

Datasets. We evaluated the proposed approach on two popular datasets for class-incremental learning, *i.e.*, CIFAR-100 [18] and ImageNet (ILSVRC 2012) [7]. Following previous works [35, 13], for a given dataset, the classes were first arranged in a fixed random order and then cut into incremental splits that come sequentially. Specifically, ImageNet is used in two CIL settings: ImageNet-Sub [35, 13, 25] contains the first 100 classes of the arranged 1,000 classes and ImageNet-Full contains the whole 1,000 classes.

Protocols. We followed [13, 9] to split the dataset into incremental steps and manage the memory budget. 1) For splitting each dataset, half of all the classes are used to train a model as the starting point, and the rest classes were divided equally to come in T incremental steps. 2) For the old data replay after each incremental step, a constant number R of images for each class was stored. In conclusion, the setting for CIL can be denoted as T-step-R-replay.

Evaluation Metrics. All models were evaluated for each step (including the first step). It can be shown as a trend

Table 1: Comparisons of average incremental accuracies (%) on CIFAR100, ImageNet-Sub, and ImageNet-Full under 5/10-step-5/10/20-replay settings with our distillation of data effect (DDE) and state-of-the-art.

	Methods	CIFAR-100		ImageNet-Sub		ImageNet-Full	
		T=5	10	5	10	5	10
(1) R=5	LUCIR†	50.76	53.60	66.44	60.04	62.61	58.01
	+ DDE (ours)	59.82+9.06	**60.53**+6.93	70.86+4.42	68.30+8.26	**66.18**+3.57	**62.89**+4.88
	PODNet†	53.38	55.97	71.43	64.90	61.01	55.36
	+ DDE (ours)	**61.47**+8.09	60.08+4.11	**74.59**+3.16	**69.26**+4.36	63.15+2.14	58.34+2.98
(2) R=10	LUCIR†	61.68	58.30	68.13	64.04	65.21	61.60
	+ DDE (ours)	**64.41**+2.73	**62.00**+3.70	71.20+3.07	69.05+5.01	**67.04**+1.83	**64.98**+3.38
	PODNet†	61.40	58.92	74.50	70.40	62.88	59.56
	+ DDE (ours)	63.40+2.00	60.52+1.60	**75.76**+1.26	**73.00**+2.60	64.41+1.53	62.09+2.53
(3) R=20	LUCIR†	63.57	60.95	70.71	67.60	66.84	64.17
	+ DDE (ours)	65.27+1.70	62.36+1.41	72.34+1.63	70.20+2.60	**67.51**+0.67	**65.77**+1.60
	PODNet†	64.70	62.72	75.58	73.48	65.59	63.27
	+ DDE (ours)	**65.42**+0.72	**64.12**+1.40	**76.71**+1.13	**75.41**+1.93	66.42+0.83	64.71+1.44
	iCaRL† [35]	57.17	52.27	65.04	59.53	51.36	46.72
	BiC [47]	59.36	54.20	70.07	64.96	62.65	58.72
	LUCIR [13]	63.17	60.14	70.84	68.32	64.45	61.57
	Mnemonics [25]	63.34	62.28	72.58	71.37	64.54	63.01
	PODNet [9]	64.83	63.19	75.54	74.33	66.95	64.13
	TPCIL [42]	65.34	63.58	76.27	74.81	64.89	62.88

Table 1: Comparisons of average incremental accuracies (%) on CIFAR100, ImageNet-Sub, and ImageNet-Full under 5/10-step-5/10/20-replay settings with our distillation of data effect (DDE) and state-of-the-art. Models with a dagger † are produced using their officially released code to assure the fair comparison. See Table 2 for more results without data replay.

of $T + 1$ classification accuracy curve or be average for all the steps, denoted as *Average Incremental Accuracy*. We also reported the *forgetting* curve following Arslan *et al.* [4]. The forgetting at step t $(t > 1)$ is calculated as $F_t = \frac{1}{t-1}\sum_{j=1}^{t-1} f_j^t$, where f_j^t denotes the performance drop of classes first learned in step j after the model has been incrementally trained up to step $t > j$ as:

$$f_j^t = \max_{l \in \{1,...,t-1\}} a_{l,j} - a_{t,j}, \forall j < t \qquad (9)$$

where $a_{i,j}$ is the accuracy of the classes first learned in steps i after the training step j. The lower f_t implies the less forgetting on previous steps, and the *Average Incremental Forgetting* is the mean of f_t for T incremental steps.

Implementation Details. Following [13, 9], for the network backbone, we adopted a 32-layer ResNet for CIFAR100, an 18-layer ResNet for ImageNet, and an SGD optimizer with momentum $\mu = 0.9$; for the classifier, a scaled cosine normalization classifier was adopted. Specifically, Douillard *et al.* [9] adopted an ensemble classifier, which is preserved in our experiments on PODNet [9]. All models were implemented with PyTorch and run on TITAN-X GPU. Since our methods can be implemented as a plug-in module, the common hyper-parameters, *e.g.*, learning rates, batch-sizes, training epochs, herding strategy, and the weights of the losses, were the same as their original settings [13, 9], which will be given in Appendix. For the proposed Distillation of Colliding Effect (DCE), we used cosine distance to measure the similarity between

images. For the number of neighborhood images except for the original image, we adopted 10 for CIFAR100 and ImageNet-Sub, and 1 for ImageNet-Full considering the calculation time. For the incremental Momentum Effect Removal (MER), α and β were trained in the finetuning stage using a re-sampled subset containing balanced old and new data. When there is no replay data, we abandoned the finetuning stage and set α and β to 0.5 and 0.8 empirically.

5.2. Results and Analyses

Comparisons with State-of-The-Art. To demonstrate the effectiveness of our method, we deployed it on LUCIR [13] and PODNet [9] using their officially released codes, and compared them with other strong baselines and state-of-the-art. Table 1 (3) lists the overall comparisons of CIL performances on the classic 5/10-step-20-replay settings. On CIFAR-100, we can observe that our Distillation of Data Effect (DDE) achieves state-of-the-art 65.42% and 64.12% average accuracy, improving the original LUCIR and POD-Net up to 1.7% and surpassing the previous best model [42] with additional data augmentation. On ImageNet-Sub, we boost LUCIR and PODNet by 1.79% on average and achieve the new state-of-the-art. For the most challenging ImageNet-Full, our model based on LUCIR achieves 67.51% and 65.77%, which surpasses the previous best model by 1.1% on average.

Data Effect Distill vs. Data Replay. Aiming to distill the data effect, our method can serve as a more storage-efficient
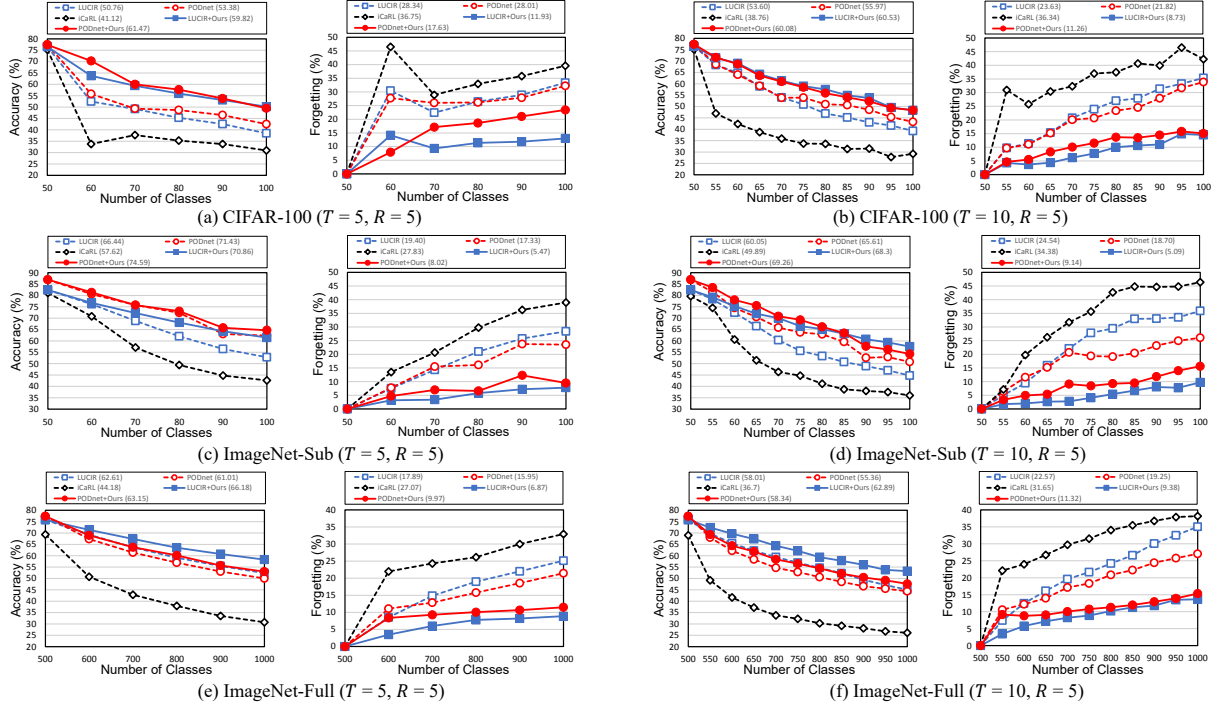
Figure 5: Comparisons of the step-wise accuracies and forgettings on CIFAR-100 (100 classes), ImageNet-Sub (100 classes) and ImageNet-Full (1000 classes) when $R = 5$.

replacement for the data-replay strategy. Figure 6 and Table 2 illustrate the effect of the data replay compared with our method. It is worth noting that when there is no data replay, *i.e.*, $R = 0$, our method achieves 59.11% average accuracy, surpassing the result of replaying five samples per class by 13.54%. To achieve the same result, traditional data-replay methods need to store around 1,000 old images in total, while our method needs nothing. The experiment empirically answers the question we asked initially: our proposed method does distill the end-to-end data effects without replay. Moreover, as the number of old data increases, data-replay based methods improve, while our method still consistently boosts the performance. The results demonstrated the data effect we distilled is not entirely overlapped with the data effect introduced by the old data.

Different Replay Numbers. To demonstrate the effectiveness and robustness of our data effect, we implemented the model on more challenging occasions where rarer old data are replayed ($R = 10, 5$). As shown in Table 1 (1) and Table 1 (2), original methods suffer a lot from the forgetting due to the missing of old data, leading to the rapid decline in the accuracies, *e.g.*, the accuracy drops 12.81% for CIFAR-100 and 4.23% for ImageNet as R decreases from 20 to 5. TPCIL also reported a 12.45% accuracy drop in their ablation study on CIFAR-100 [42]. In contrast, our method shows its effectiveness agnostic to the datasets and the number of replay data. Specifically, our method achieves more improvement when the data constrain is stricter. For ex-

ample, it obtains up to 9.06% and 4.88% improvement on CIFAR-100 and ImageNet when replaying only 5 data per class. To see the step-wise results, we draw Figure 5 to show the accuracy and forgetting for each learning step. Starting from the same point, our method (in solid) suffers from less forgetting and performs better at each incremental step.

Effectiveness of Each Causal Effect. Table 3 shows the results of using Distillation of Colliding Effect (DCE) and incremental Momentum Effect Removal (MER) alone and together. One can see that the boost of using each component alone is lower than using them combined. Therefore, they do not conflict with each other and demonstrate that both the causal effects play an important role in CIL.

Different Weight Assignments. Table 4 shows the results of using Distillation of Colliding Effect (DCE) with different weight assignment strategies. *Top-n* means for each image i, we select n nearest neighbor images as discussed in Section 4.1 and calculate the average prediction of the images i and the mean of those n images. *Rand* and *Bottom* use randomly selected images or the most dissimilar (farthest) images. *Variant1* and *Variant2* select the same images as *Top 10*, but modify the weight value for those $n + 1$ images (See Appendix for detail). *Variant1* lower the weight of the image itself and split more weight to the neighbors, while *Variant2* split the weight in proportion to the softmax of the cosine similarity between images. By comparing the results of *Top n*, *Rand*, and *Bottom*, we demonstrated the importance of sorting images based on the similarity of

their old feature. Larger n can lead to somewhat better performance, while as the nearer images should have larger weights according to Section 4.1, only selecting the nearest image ($n = 1$) is also good. Different variants to assign weight values among images, as long as they follow Eq. (7), will not affect the final prediction much.
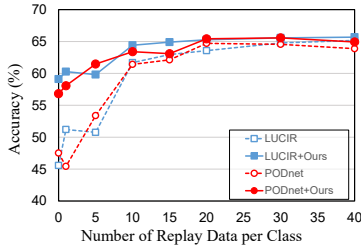

Figure 6: Comparisons of average accuracies with different number of replay data per class. Experiments are conducted on CIFAR-100 with five incremental steps.

Methods	CIFAR-100		ImageNet-Sub	
	T = 5	10	5	10
Baseline	45.57	32.72	58.55	45.06
Ours	**59.11**+13.54	**55.31**+22.59	**69.22**+10.67	**65.51**+20.45

Table 2: Comparisons of average accuracies with no replay data. Baseline is LUCIR.

Robustness of Incremental Momentum Effect Removal. In Table 5, we demonstrated the robustness of our incremental Momentum Effect Removal (MER) under the different number of incremental steps. Even in a challenging situation where $N = 25$, it still has a significant improvement.

Different Size of the Initial Task. As shown in Table 6, the performance of all the methods decreases due to the degraded representation learned from fewer classes. However, our method still shows a steady improvement compared to the baseline.

6. Conclusions

Can we save the old data for anti-forgetting in CIL without actually storing them? This paradoxical question has been positively answered in this paper. We first used causal graphs to consider CIL and its anti-forgetting techniques in a causal view. Our finding is that the data replay, arguably the most reliable technique, benefits from introducing the causal effect of old data in an end-to-end fashion. We proposed to distill the causal effect of the collider: the old feature of a new sample, and show that such distillation is causally equivalent to data replay. We further proposed to remove the incremental momentum effect during testing, to achieve balanced old and new class prediction. Our causal solutions are model-agnostic and helped two strong baselines surpass their previous performance and other methods. In future, we will investigate more causal perspectives in CIL, which may have great potential to achieve the ultimate life-long learning.

	Methods	R = 5	10	20
Accuracy (%)	Baseline [13]	50.76	61.68	63.57
	+All	59.82+9.06	64.41+2.73	65.27+1.70
	+DCE	58.53+7.77	64.04+2.36	65.18+1.61
	+MER	53.55+2.79	63.40+1.72	64.43+0.86
Forgetting (%)	Baseline [13]	28.34	17.51	14.08
	+All	11.93-16.41	6.23-11.28	7.11-6.97
	+DCE	16.82-11.52	10.16-7.35	8.41-5.67
	+MER	17.45-10.89	12.15-5.36	10.01-4.07

Table 3: The individual improvements of the proposed Distillation of Colliding Effect (DCE) and incremental Momentum Effect Removal (MER). Experiments are conducted on CIFAR-100 with 5 incremental steps.

R	Baseline	Top1	Top5	Top10	Rand	Bottom	Variant1	Variant2
5	50.76	55.92	58.12	58.53	54.88	41.70	58.22	58.41
10	61.68	63.51	63.66	64.04	53.80	51.41	63.54	63.97
20	63.57	64.76	64.93	65.18	64.16	57.07	64.87	65.22

Table 4: Comparisons of average accuracies using the Distillation of Colliding Effect (DCE) with different weight assignments. Experiments are conducted on CIFAR-100 with 5 incremental steps. Baseline is LUCIR.

	Methods	N=1	2	5	10	25
Accuracy(%)	Baseline [13]	67.98	64.04	50.76	53.6	43.52
	+MER	68.48	65.44	53.55	57.86	45.95
RI (%)	+MER	0.74	2.19	5.50	7.95	5.58
Forgetting (%)	Baseline [13]	28.82	23.74	28.34	23.63	32.22
	+MER	25.24	16.55	17.45	11.68	20.47
RI (%)	+MER	12.42	30.29	38.43	50.57	36.47

Table 5: Evaluation of average accuracies using incremental Momentum Effect Removal (MER) with varying incremental steps. RI(%) is the relative improvement in increasing accuracy and lowering forgetting with MER. Experiments are conducted on CIFAR-100 for $R = 5$.

Class size of initial task	Methods	Average Accuracy(%)		
		R = 5	10	20
10	Baseline [13]	50.45	54.39	58.87
	Ours	**52.15**+1.70	**55.71**+1.32	**59.17**+0.30
20	Baseline [13]	51.95	56.31	59.42
	Ours	**55.88**+3.93	**58.38**+2.07	**59.93**+0.51

Table 6: Comparisons of average accuracies with the initial task of fewer classes. Experiments are conducted on CIFAR-100 with each incremental step of 10 classes.

7. Acknowledgements

References

[1] Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Cordelia Schmid, and Karteek Alahari. End-to-End Incremental Learning. In *ECCV*, 2018. 1, 2

[2] Gert Cauwenberghs and Tomaso Poggio. Incremental and Decremental Support Vector Machine Learning. In *Adv. Neural Inf. Process. Syst.*, 2001. 2

[3] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. Visual Causal Feature Learning. *arXiv preprint arXiv:1412.2309*, 2015. 2

[4] Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian Walk for Incremental Learning: Understanding Forgetting and Intransigence. In *ECCV*, 2018. 6

[5] Yulai Cong, Miaoyun Zhao, Jianqiao Li, Sijia Wang, and Lawrence Carin. GAN Memory with No Forgetting. In *NeurIPs*, 2020. 2

[6] Ronald Davis and Yi Zhong. The Biology of Forgetting–A Perspective. In *Neuron*, 2017. 1

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei Fei Li. ImageNet: a Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 2, 5

[8] Zhang Dong, Zhang Hanwang, Tang Jinhui, Hua Xiansheng, and Sun Qianru. Causal Intervention for Weakly Supervised Semantic Segmentation. In *NeurIPS*, 2020. 2

[9] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *ECCV*, 2020. 1, 2, 5, 6

[10] Robert French. Catastrophic Forgetting in Connectionist Networks. *Trends in Cognitive Sciences – TRENDS COGN SCI*, 2006. 1

[11] S. I. Hill and R. C. Williamson. An analysis of the exponentiated gradient descent algorithm. In *ISSPA*, 1999. 1

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[13] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a Unified Classifier Incrementally via Rebalancing. In *CVPR*, 2019. 1, 2, 4, 5, 6, 8

[14] Yen-Chang Hsu, Yen-Cheng Liu, and Zsolt Kira. Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines. In *NeurIPS Continual Learning Workshop*, 2018. 1

[15] Nitin Kamra, Umang Gupta, and Yan Liu. Deep Generative Dual Memory Network for Continual Learning. *arXiv preprint arXiv:1710.10368*, 2017. 2

[16] Ronald Kemker and Christopher Kanan. FearNet: Brain-Inspired Model for Incremental Learning. In *ICLR*, 2018. 2

[17] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *PNAS*, 2016. 3

[18] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *Tech. Rep., University of Toronto*, 2012. 2, 5

[19] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From N to N+1: Multiclass Transfer Incremental Learning. In *CVPR*, 2013. 2

[20] Kimin Lee, Kibok Lee, H. Lee, and Jinwoo Shin. A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks. In *NeurIPS*, 2018. 2

[21] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming Catastrophic Forgetting With Unlabeled Data in the Wild. In *ICCV*, 2019. 2

[22] Zhizhong Li and Derek Hoiem. Learning without Forgetting. In *TPAMI*, 2017. 2

[23] Zhizhong Li and Derek Hoiem. Improving Confidence Estimates for Unfamiliar Examples. In *CVPR*, 2020. 2

[24] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive Aggregation Networks for Class-Incremental Learning. In *CVPR*, 2021. 2

[25] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru Sun. Mnemonics Training: Multi-Class Incremental Learning Without Forgetting. In *CVPR*, 2020. 1, 2, 5, 6

[26] Arun Mallya and Svetlana Lazebnik. PackNet: Adding Multiple Tasks to a Single Network by Iterative Pruning. In *CVPR*, 2018. 1

[27] Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. *arXiv preprint arXiv:2006.04315*, 2020. 2

[28] German Parisi, Ronald Kemker, Jose Part, Christopher Kanan, and Stefan Wermter. Continual Lifelong Learning with Neural Networks: A Review. In *Neural Networks*, 2018. 1

[29] Judea Pearl. Causality: Models, reasoning, and inference, second edition. *Causality*, 2000. 2

[30] Judea Pearl. Interpretation and Identification of Causal Mediation. *Psychological methods*, 19, 2014. 3

[31] Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016. 2, 3, 4

[32] Jiaxin Qi, Yulei Niu, Jianqiang Huang, and Hanwang Zhang. Two causal principles for improving visual dialog. In *CVPR*, 2020. 2

[33] Ning Qian. On the momentum term in gradient descent learning algorithms. In *Neural Networks*, 1999. 1

[34] Anders Rasmussen, Riccardo Zucca, Fredrik Johansson, Dan-Anders Jirenhed, and Germund Hesslow. Purkinje cell activity during classical conditioning with different conditional stimuli explains central tenet of Rescorla-Wagner model. In *PNAS*, 2015. 1

[35] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental Classifier and Representation Learning. In *CVPR*, 2017. 1, 2, 4, 5, 6

[36] A. Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. In *Connection Science*, 1995. 1

[37] Bernhard Schölkopf. Causality for Machine Learning. *arXiv preprint arXiv:1911.10500*, 2019. 2

[38] Hanul Shin, Jung Lee, Jaehong Kim, and Jiwon Kim. Continual Learning with Deep Generative Replay. In *NeurIPS*, 2017. 2

[39] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, pages 1139–1147, 2013. 5

[40] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *NeurIPS*, 2020. 2, 5

[41] Kaihua Tang, Yulei Niu, Jianqiang Huang, Jiaxin Shi, and Hanwang Zhang. Unbiased scene graph generation from biased training. In *CVPR*, 2020. 2

[42] Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. Topology-Preserving Class-Incremental Learning. In *ECCV*, 2020. 2, 6, 7

[43] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-Shot Class-Incremental Learning. In *CVPR*, 2020. 2

[44] Sebastian Thrun. Lifelong Learning Algorithms. In *Springer US*, 1998. 1

[45] Gido van de Ven, Hava Siegelmann, and Andreas Tolias. Brain-inspired replay for continual learning with artificial neural networks. In *Nature Communications*, 2020. 2

[46] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual commonsense r-cnn. In *CVPR*, 2020. 2

[47] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large Scale Incremental Learning. In *CVPR*, 2019. 1, 2, 6

[48] Xu Yang, Hanwang Zhang, and Jianfei Cai. Deconfounded image captioning: A causal retrospect. *arXiv preprint arXiv:2003.03923*, 2020. 2

[49] Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. In *NeurIPS*, 2020. 2

[50] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-Shot Incremental Learning with Continually Evolved Classifiers. In *CVPR*, 2021. 1