Appendix on "CafeBoost: Causal Feature Boost to Eliminate Task-Induced Bias for Class Incremental Learning"

Benliu Qiu Hongliang Li* Haitao Wen Heqian Qiu
Lanxiao Wang* Fanman Meng Qingbo Wu Lili Pan
University of Electronic Science and Technology of China, Chengdu, China
{qbenliu, haitaowen, hqqiu, lanxiao.wang}@std.uestc.edu.cn
{hlli, fmmeng, qbwu}@uestc.edu.cn panlili8255@gmail.com

This appendix will provide further details for the main paper, including more experimental results (Appendix A), basic knowledge of causal graph (Appendix B), a proof of Equation 6 in the main paper (Appendix C) and more implementation details (Appendix D).

A. Additional Results

Tab. 1 shows additional results on ImageNet-Full for 25 incremental tasks. We observe that our approach boosts the accuracy of LUCIR by 5.39% and reduces the corresponding forgetting by 1.11%. The curves of average incremental accuracy and forgetting rate are also given in Fig. 1. Our approach when combined with LUCIR, achieves a smoothly and slowly decreasing curve of accuracy and the corresponding increasing curve of forgetting. In addition, we plot curves of the forgetting rate on CIFAR-100 and ImageNet-Sub for 5, 10, and 25 incremental tasks, as shown in Fig. 3. It can be observed that our approach suffers from less forgetting in most tasks.

Method	LwF	BiC	iCaRL	LUCIR	GeoDL	iCaRL+ours	LUCIR+ours
Accuracy (%)	36.87	53.47	43.14	56.56	**62.20**	41.30 -1.84	61.95 +5.39
Forgetting (%)	49.84	33.17	38.80	30.30	15.11	33.59 -5.21	**14.00 -1.11**

Table 1. The average incremental accuracy and the forgetting rate on ImageNet-Full for 25 incremental tasks. 20 exemplars are used per class.

B. Basic Knowledge of Causal Graph

B.1. Causal Graph

As shown in Fig. 2, there are three basic causal configurations in causal theory: chain, collider, and fork, which consist of more complex causal graphs. **Chain** is a configuration of variables which contains three nodes and two edges with one edge directed into and one edge directed out of the middle variable, as shown in Fig. 2 (a). **Collider** with one node receives edges from two other nodes, is illustrated in Fig. 2 (b). **Fork** with two arrows emanating from the middle variable, is shown in Fig. 2 (c), which is the prime configuration we consider in the main paper.

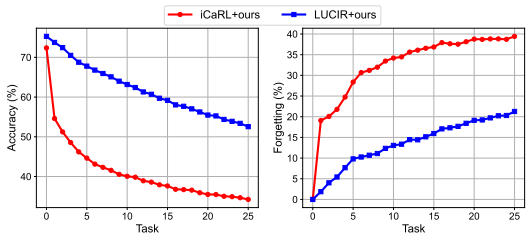

Figure 1. The curves of accuracy and forgetting on ImageNet-Full for 25 incremental tasks. 20 exemplars are used for each class.

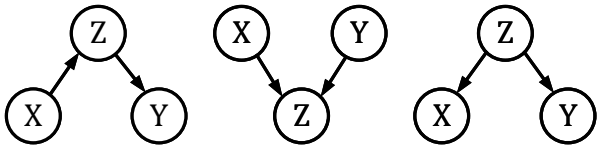

Figure 2. Three basic configurations of causal graphs.

B.2. Conditional independence

[5] summarizes three rules to determine dependencies and independencies of two end nodes in the above-mentioned three basic configurations. These rules are described below, which are easily understood with the help of Fig. 2.

Rule 1 (Conditional Independence in Chains) *Two variables X and Y are conditionally independent given Z, if there is only one unidirectional path between X and Y and Z is any set of variables that intercepts that path.*

Rule 2 (Conditional Independence in Colliders) *If a*

*Corresponding authors.

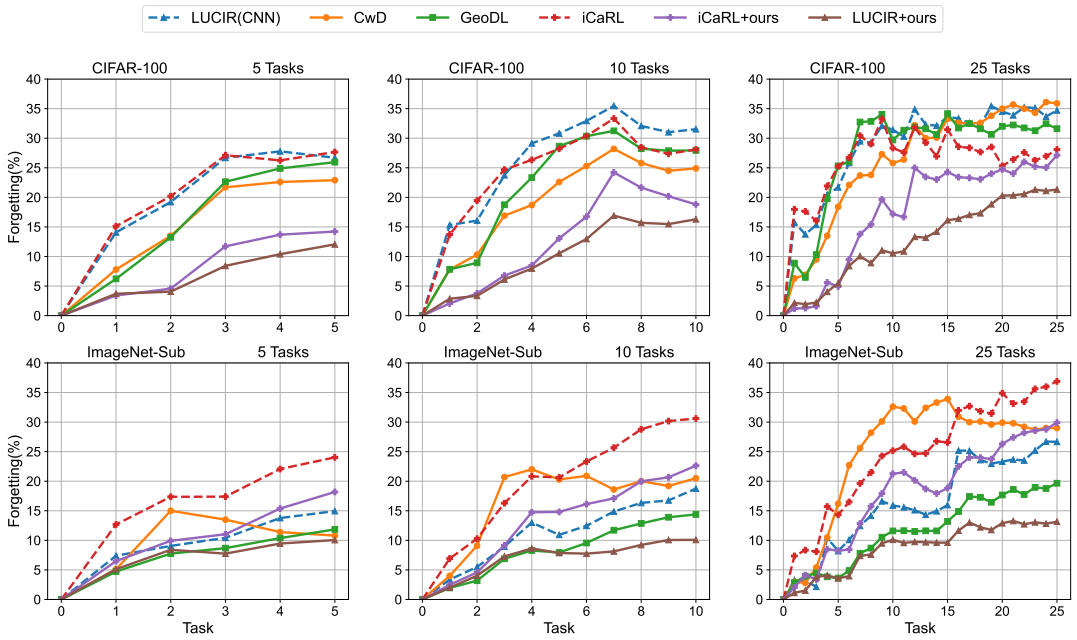

Figure 3. The forgetting rate on CIFAR-100 and ImageNet-Sub for 5, 10, and 25 incremental tasks. 20 exemplars are used for each class.

variable Z is the collision node between two variables X and Y, and there is only one path between X and Y, then X and Y are unconditionally independent but are dependent conditional on Z and any descendants of Z.

Rule 3 (Conditional Independence in Forks) *If a variable Z is a common cause of variables X and Y, and there is only one path between X and Y, then X and Y are independent conditional on Z.*

According to **Rule 3**, in causal path $X \leftarrow T \rightarrow Y$, variable X cannot exert a causal effect on variable Y when conditioned on variable T. Therefore, the spurious correlation between X and Y, i.e., the task bias, does not exist for task incremental learning in the main paper . However, the task identifier T is not conditioned in class incremental learning, and hence X and Y are likely dependent, causing the task bias.

B.3. Causal intervention

B.3.1 *do*-operator

[5] introduces a do-operator $do(\cdot)$. $do(X = x)$ means we fix $X = x$ while $X = x$ denotes the variable X takes a value x. As a result, $P(Y = y|X = x)$ is the probability of $Y = y$ conditional on $X = x$, whereas $P(Y = y|do(X = x))$ is the probability of $Y = y$ when we intervene to make $X = x$. In our main paper and this appendix, we simplify $P(Y = y|X = x), P(Y = y|do(X = x))$ as $P(Y|X), P(Y|do(X)$, respectively for convenience.

B.3.2 backdoor criterion

Definition 1 (The Backdoor Criterion) *Given an ordered pair of variables (X, Y) in a directed acyclic graph G, a set of variables Z satisfies the backdoor criterion relative to (X, Y) if no node in Z is a descendant of X, and Z blocks every path between X and Y that contains an arrow into X.*

The backdoor criterion answers in what conditions the structure of the causal graph is sufficient for computing a causal effect from a given data set. [5] points out that if a set of Z satisfies the backdoor criterion for X and Y, the causal effect of X on Y is given by

$$P(Y|do(X)) = \sum_z P(Y|X, Z)P(Z). \tag{1}$$

In Fig. 3 (c) of the fork structure, the variable Z is not a descent of X and it blocks the only path $X \leftarrow Z \rightarrow Y$ between X and Y. Therefore, Z satisfies the backdoor criterion and the causal effect of X on Y can be calculated by Eq. (1). In our main paper, we adopt the backdoor criterion and achieve **Equation 3**.

C. Proof of Equation 6

Following the work [9] that firstly proposed NWGM to tackle image captioning task, we substitute NWGM[Softmax($f_y(\boldsymbol{x}, \boldsymbol{d}_t)$)] for $\mathbb{E}_{\boldsymbol{d}_t}$[Softmax($f_y(\boldsymbol{x}, \boldsymbol{d}_t)$)]. We denote $f_y(\boldsymbol{x}, \boldsymbol{d}_t)$ as $n_{y,t}$ and the probability of $\boldsymbol{d}_t$ as $P(\boldsymbol{d}_t)$. Combining Softmax($f_y(\boldsymbol{x}, \boldsymbol{d}_t)$) $\propto$ exp($f_y(\boldsymbol{x}, \boldsymbol{d}_t)$),

Methods	Epochs	LR	LR schedule	LR decay rate	Weight decay	Momentum	Batch size	Other hyper-parameters
iCaRL	160 (90)	0.1	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$T = 2.0, \beta = 0.25$
LUCIR	160 (90)	0.1	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$dist = 0.5, K = 2, rr = 0.0, \lambda_{lf} = 5(10), \lambda_{mr} = 1.0$
GeoDL	160 (90)	0.1	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$dist = 0.5, K = 2, rr = 0.0, \lambda_{lf} = 5(10)$
CwD	160 (90)	0.1	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$dist = 0.5, K = 2, \lambda_{lf} = 5(10), \lambda_{mr} = 1.0$
CSCCT	160 (90)	0.4 (0.02)	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$dist = 0.5, K = 2, \lambda_{lf} = 5(10), \lambda_{mr} = 1.0$
CafeBoost	160 (90)	0.1	80 120 (30 60)	0.1	5×10^{-4} (1×10^{-4})	0.9	128	$dist = 0.5, K = 2, rr = 0.0, \lambda_{lf} = 5(10), \lambda_{mr} = 1.0$

Table 2. Hyper-parameters for all methods. Values in parentheses are for ImageNet.

we can obtain

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{d}_t}[\text{Softmax}(f_y(\boldsymbol{x}, \boldsymbol{d}_t))] \\
\approx &\text{NWGH}[\text{Softmax}(f_y(\boldsymbol{x}, \boldsymbol{d}_t))] \\
= &\frac{\prod_t \exp(n_{y,t})^{P(\boldsymbol{d}_t)}}{\sum_i \prod_t \exp(n_{i,t})^{P(\boldsymbol{d}_t)}} \\
= &\frac{\exp(\mathbb{E}_{\boldsymbol{d}_t}[n_{y,t}])}{\sum_i \exp(\mathbb{E}_{\boldsymbol{d}_t}[n_{i,t}])} \\
= &\text{Softmax}(\mathbb{E}_{\boldsymbol{d}_t}[f_y(\boldsymbol{x}, \boldsymbol{d}_t)]),
\end{aligned} \tag{2}$$

i.e, **Equation 6** in the main paper.

D. Implementation Details

We adopt a 18-layer ResNet [2] for both CIFAR-100 and ImageNet. Random cropping and horizontal flipping are only data augmentations used in training datasets for all the methods. Details of hyper-parameters of all methods are listed in Tab. 2. Additional settings of some baseline methods are described below.

iCaRL [6]: T denotes the temperature for distillation, set to 2.0. β is the weight coefficient for distillation, set to 0.25. Both classification loss and KD loss use multi-class cross entropy loss, following [4].

LUCIR [3]: Distance $dist$ and K for margin ranking loss is set to 0.5 and 2, respectively. The weight of less forget loss is $\lambda_{lf} = 5$ for CIFAR-100 and 10 for ImageNet. Margin ranking loss gains a weight $\lambda_{mr} = 1.0$ for both datasets.

GeoDL [8]: We implement GeoDL on the base of LUCIR, and thus its corresponding hyper-parameters are the same as LUCIR. The weight of distillation loss considering geodesic flow is set to 1.0.

CwD [7]: CwD is also implemented on LUCIR and its hyper-parameters are mostly the same as LUCIR. For CIFAR-100, CwD sets the learning rate for first task to 0.1, whereas 0.2 for ImageNet. The rejection threshold of class-wise decorrelation loss (L_{CwD}) is 1. The weight of L_{CwD} is 0.5 and 0.75 for CIFAR-100 and ImageNet, respectively.

CSCCT [1]: As a plugin method, CSCCT is plug into LUCIR by us. The coefficient of cross-space clustering loss and controlled transfer loss is 3 and 1.5 respectively. The temperature of controlled transfer loss is set to 2.0.

CafeBoost (ours): Our plugin method can be added on the base of iCaRL and LUCIR. Therefore, the corresponding hyper-parameters are same as theirs. The causal debias module is trained for 160(90) epochs in its first training stage for CIFAR-100 (ImageNet), where the feature extractor of ResNet is fixed, as shown in Algorithm 1 of the main paper. Moreover, we update the feature extractor by a learning rate 0.01 during class incremental phases.

E. Memory and Time Cost

For both CIFAR-100 and ImageNet-Sub with 5 incremental phases, CafeBoost model consumes 49.2 MB memory space, only 9.4% more than LUCIR (44.9 MB). For training time using NVIDIA TITAN Xp, CafeBoost requires 5-6 h and 18-22 h training time on CIFAR-100 and Image-Sub, respectively.

References

[1] Arjun Ashok, K. J. Joseph, and Vineeth Balasubramanian. Class-incremental learning with cross-space clustering and controlled transfer. In *ECCV*, 2022. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[3] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 2

[4] Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. In *ACCV*, 2019. 2

[5] Judea Pearl, Madelyn Glymour, and Nicholas P. Jewell. *Causal Inference In Statistics: A primer*. Wiley, 2016. 1, 2

[6] Sylvestre Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *CVPR*, 2017. 2

[7] Yujun Shi, Kuangqi Zhou, Jian Liang, Zihang Jiang, Jiashi Feng, Philip H. S. Torr, Song Bai, and Vincent Y. F. Tan. Mimicking the oracle: An initial phase decorrelation approach for class incremental learning. In *CVPR*, 2022. 2

[8] Christian Simon, Piotr Koniusz, and Mehrtash Harandi. On learning the geodesic path for incremental learning. In *CVPR*, 2021. 2

[9] Kelvin Xu, Jimmy Lei Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. 2015. 2