Dual-Estimator: Decoupling Global and Local Semantic Shift for Drift Compensation in Class-Incremental Learning (Appendix)

Fankang Xu[1] *, Lu Jin[1] *, Yanpeng Sun[2], Shiyu Xuan[1], Zechao Li[1] †
[1] School of Computer Science and Engineering, Nanjing University of Science and Technology
[2] Singapore University of Technology and Design
{frank.xu, lujin, yanpeng_sun, shiyu_xuan, zechao.li}@njust.edu.cn

A. Scope and Notational Conventions

This appendix offers additional experimental details and results to supplement the main paper. Specifically, Appendix 2 presents the pseudocode of the proposed method, Appendix 3 details the experimental settings and compared approaches, and Appendix 4 provides additional results omitted from the main paper due to space limitations, such as the performance under the 5-task split.

For clarity of exposition, the symbols used in this study are summarized below.

- D_t^{train}: the training data of task t.
- $\widetilde{D}_t^{train}$: augmented data via rotation-based pixel fusion.
- D_t^{test} : the test data of task t.
- (x_i, y_i): the i-th image-label pair.
- C_t : the candidate label set of task t.
- f_{θ_t} : the backbone of the model at task t.
- g_{π_t}: the fully connected classifier of the model at task t.
- $\mathbf{z}$: the embedding of sample x extracted by f_{θ_t}.
- $\mathbf{m}$: the embedding of sample x extracted by $f_{\theta_{t-1}}$.
- $\mathbf{p}_c$: the prototype vector of class c.
- $\mathbf{Z}$: the matrix formed by stacking $\mathbf{z}$.
- $\mathbf{M}$: the matrix formed by stacking $\mathbf{m}$.
- $\mathbf{P}$: the matrix formed by stacking $\mathbf{p}$.
- $(\mathbf{A}, \mathbf{B})$: form the low-rank mapping matrix of Ω^{LoR}.
- $\mathbf{Q}_k$: form the i-th mapping matrix of Ω^{MoE}.
- Ω^{LoR}: low-rank estimator.
- Ω^{MoE}: mixture-of-experts estimator.
- $\mathcal{T}_k^{local}$: the k-th transfer function, a linear layer in Ω^{MoE}.
- $\mathcal{T}^{global}$: the transfer function $(\mathbf{A}, \mathbf{B})$ in Ω^{LoR}.
- $\boldsymbol{\mu}_k$: the key vector used to route $\mathcal{T}_k^{local}$.

B. Learning Procedure in Pseudocode

The proposed Dual-E is a plug-and-play approach designed for EFCIL, consisting of MoE-E and LoR-E. Dual-E provides a reliable prototype compensation technique for prototype-based CIL methods, alleviating the issue of outdated prototype representations caused by feature drift. The drift compensation is typically applied after the model adapts to a new task and before performing subsequent operations using the old class prototypes. To facilitate understanding of the overall EFCIL process, Algorithm 1 presents the pseudocode that integrates Dual-E with the commonly used cross-entropy classification loss and knowledge distillation loss.

In step 1, the model for the new task inherits parameters from the previous task. In steps 2–5, the model is trained on the new class data. Then, in step 9, the samples in the old-class representation space are clustered into K groups using k-means, and the parameters $\mathbf{Q}_k$ of the transfer functions in MoE-E are updated accordingly. For steps 15–17, the parameters $\mathbf{A}$ and $\mathbf{B}$ of LoR-E are updated for n iterations, where n is small and set to 2 in our implementation. In steps 19 and 20, the prototypes are updated using MoE-E and LoR-E, respectively, and the two are fused based on the weights obtained in step 21. Finally, during inference in step 24, NCM-based classification is conducted using the new class prototypes together with the corrected old class prototypes.

C. Experimental Details

C.1. Dataset Details

In this study, we conduct experimental evaluations using three commonly adopted benchmark datasets in the CIL domain: CIFAR-100 [7], Tiny-ImageNet [8], and ImageNet-Subset [1]. CIFAR-100 comprises 60000 color images of size 32×32, evenly divided among 100 classes, with 500 images per class for training and 100 per class for testing. Tiny-ImageNet, a reduced-scale variant of ImageNet, contains 200 classes with 100000 training images and 10000 test images, each of size 64×64. These datasets serve as a solid foundation for assessing model adaptability in incremental learning scenarios. ImageNet-Subset, extracted from the ImageNet (ILSVRC 2012) dataset, consists of 100 selected classes, totaling 130000 training images and 5000 test images, all standardized to 224×224 pixels.

Beyond these standard benchmarks, two fine-grained image classification datasets, CUB-200 [14] and Stanford Cars [6], are included to evaluate the Dual-E approach under

*Equal contribution.
†Corresponding author.

Algorithm 1 Pseudocode of Dual-Estimator.

Input: Training data D_t^{train} of task t ($t > 1$), $f_{\theta_{t-1}}$, $g_{\pi_{t-1}}$, and randomly initialized matrices $\mathbf{A}, \mathbf{B}$ and $\{\mathbf{Q}_k\}_{k=1}^{K}$.
Output: Inference results.

/* —— New Task Adaptation ——*/
Initialize $f_{\theta_t} \leftarrow f_{\theta_{t-1}}$ and $g_{\pi_t} \leftarrow g_{\pi_{t-1}}$.
for i iterations **do**
  Sample $(x, y) \sim D_t^{train}$.
  compute the $\mathcal{L}_{cls}$ and $\mathcal{L}_{kd}$ using Eq. (1).
  Update model parameters using gradient descent.
end for
Generate class prototype $\mathbf{p}_c$ for $\forall c_j \in C_t$.
Freeze the backbone f_t.
/* —— Train MoE-Estimator ——*/
Obtain $\{S_k\}_{k=1}^{K}$ using Eq. (5).
for S_k in $\{S_k\}_{k=1}^{K}$ **do**
  Compute $\mathbf{Z}_k$ and $\mathbf{M}_k$.
  Update $\mathbf{Q}_k$ using Eq. (8).
end for
/* —— Train LoR-Estimator ——*/
Compute $\mathbf{Z}$, $\mathbf{M}$, and $\mathbf{P}$.
for n iterations **do**
  Update $\mathbf{A}$ and $\mathbf{B}$ using Eq. (11).
end for
/* —— Inference with Fused Prototypes ——*/
for c in $\cup_{j=1}^{t-1} C_j$ **do**
  Obtain $\Omega^{MoE}(\mathbf{p}_c)$ using Eq. (9).
  Obtain $\Omega^{LoR}(\mathbf{p}_c)$ using Eq. (12).
  Compute β_c using Eq. (16).
  Update $\mathbf{p}_c$ using Eq. (15).
end for
Perform NCM-based inference using the updated old class prototypes and the new task prototypes.

more challenging conditions characterized by high inter-class similarity and localized discriminative features. CUB-200 encompasses 200 bird species, with images resized to 224×224 pixels and divided into 5994 training images and 5794 test images. Stanford Cars comprises 196 car model classes, each image also standardized to 224×224 pixels, with 8144 images allocated for training and 8041 for testing. These datasets are particularly useful for examining the model's ability to capture subtle and localized semantic distinctions in an incremental learning context.

Additionally, experiments are conducted on a large-scale dataset ImageNet-1K [1], which includes 1000 classes with a total of 1.3 million images, each resized to 224×224 pixels. The class partition of the datasets used in this work under the CIL scenario is summarized in Table A1.

Table A1. Incremental setting of datasets, where each element is the number of classes per task when divided into T tasks, and the initial and subsequent tasks contain an equal number of classes.

Datasets	**T=5**	**T=7**	**T=10**	**T=14**	**T=20**	**T=50**
CIFAR-100	20	-	10	-	5	-
Tiny-ImageNet	40	-	20	-	10	4
ImageNet-Subset	20	-	10	-	5	-
CUB-200	60	-	20	-	10	-
Stanford Cars	-	28	-	14	-	-
ImageNet-1K	-	-	-	-	-	20

C.2. Implementation Details

In our experiments, the number of expert networks K in MoE-E is set to 4, while the rank of parameters in LoR-E is configured as 40% of the feature dimension. Since the proposed Dual-E serves as a plug-and-play module designed to update old-class prototypes after each incremental task, we next describe the new-task learning configurations based on commonly used baseline model that employ cross-entropy classification and knowledge distillation.

For CIFAR-100, the same data augmentation procedures as those used for CIFAR-10 within the PyCIL framework [17] are adopted to ensure consistency across all compared methods. To follow conventional training practices, random cropping and horizontal flipping are applied to the training samples for every dataset, consistent with previous studies [2, 3]. The network is optimized via stochastic gradient descent (SGD) with a momentum of 0.9. For the initial task, training is conducted for 200 epochs, beginning with a learning rate of 0.1, which is reduced by a factor of 0.1 at the 60, 120, and 160 epochs. Each subsequent task is trained for 100 epochs, with the learning rate set to 0.05 for CIFAR-100, 0.005 for Tiny-ImageNet, Stanford Cars, and CUB-200, and 0.01 for ImageNet-Subset and ImageNet-1K, all following the same decay schedule applied every 45 epochs. The weight decay of 5×10^{-4} is applied to the initial task and 2×10^{-4} to the following tasks. The batch size is configured as 128 for CIFAR-100, Stanford Cars, and CUB-200, and 64 for the other datasets. The backbone ResNet-18 [4] in our experiments is trained from scratch on all datasets, except for CUB-200 and Stanford Cars. For these two fine-grained datasets, the backbone is initialized with ImageNet-1K [1] pre-trained weights to strengthen its feature extraction ability, which is a widely adopted practice in fine-grained recognition tasks [3]. This initialization strategy effectively alleviates overfitting and accelerates convergence when handling datasets with limited samples and subtle inter-class distinctions. All experiments are implemented on a computing server equipped with two GeForce RTX 4090 GPUs. To reduce the influence of class ordering on experimental outcomes, each experiment is repeated

with three random seeds, 1993, 2017, and 2020, and the mean results are reported. The experimental configurations of the compared methods are detailed below.

- **LwF** [10]. the first task is trained with an initial learning rate of 0.1, a momentum of 0.9, a batch size of 128, and a weight decay of 5×10^{-4} for 200 epochs. The learning rate is decreased by a factor of 10 at epochs 60, 120, and 160 across all benchmark datasets. For the subsequent tasks, the initial learning rate is set to 0.05 for CIFAR-100 and ImageNet-100, and 0.001 for Tiny-ImageNet, with the rate decayed by 2×10^{-4} after 45 and 90 epochs during a 100-epoch training schedule. The temperature parameter is fixed at 2, while the regularization coefficient is assigned a value of 10 for CIFAR-100 and Tiny-ImageNet, and 5 for ImageNet-100.
- **ABD** [13]. We follow the configurations provided in the official implementation. In particular, SGD is employed for optimization on all datasets, with an initial learning rate of 0.1 and a momentum value of 0.9. The loss weights for knowledge distillation μ and fine-tuning β are set to 0.1 and 1, respectively.
- **PASS** [18]. The experimental setup is based on the publicly available code from the original authors. The Adam optimizer is applied with an initial learning rate of 0.001, which is reduced by a factor of 10 every 45 epochs. Both the distillation weight γ and the coefficient for the prototype augmentation loss λ are assigned a value of 10.
- **FKSR** [16].The experiments follow the parameter configurations provided in the official code. The Adam optimizer is applied with a learning rate of 0.001 in all scenarios, except for the first task on ImageNet-Subset, where SGD is used with an initial learning rate of 0.1. The learning rate decays by a factor of 0.1 at the milestones $[80, 120, 150]$. The model hyperparameters λ_{pr} and λ_{pks} are both set to 10.
- **NAPA-VQ** [12]. We adopt the configurations from the publicly released code. The model is optimized using SGD with an initial learning rate of 0.001, which is reduced by a factor of 0.1 every 45 epochs. The weight for the prototype augmentation loss, λ_1, and the weight for the knowledge distillation loss, λ_2, are both set to 10.
- **FCS** [9]. For the first task on ImageNet-Subset, the model is optimized using SGD with an initial learning rate of 0.1, which is reduced by a factor of 0.1 at epochs 80, 120, and 150. For all other tasks, Adam is employed with a learning rate of 0.001, decaying by a factor of 0.1 every 45 epochs. The four hyperparameters α, β, γ, and λ are set following the original authors' guidance: $10/10/1/0.1$ for CIFAR-100 and Tiny-ImageNet, and $10/10/1/0.03$ for ImageNet-Subset.
- **EFC** [11]. We follow the configuration provided in the method's open-source code. The Adam optimizer is employed with a weight decay of 2×10^{-4}. A constant learning rate of 1×10^{-4} is used for Tiny-ImageNet and CIFAR-100, while for ImageNet-Subset, the backbone is trained with a learning rate of 1×10^{-5} and the heads with 1×10^{-4}. The two hyperparameters are set as $\lambda_{\mathrm{EFM}} = 10$ and $\eta = 0.1$.
- **SDC** [15]. Following the recommended settings, the Gaussian kernel parameter σ is fixed at 0.3 for all datasets.
- **ADC** [3]. We follow the configuration provided in their open-source code. The method builds on the LwF [10] framework within PyCIL [17], with the knowledge distillation weight fixed at 10. The step size, α, is set to 25, and the number of generated samples, m, is 100. During inference, predictions are obtained using the NCM classifier, as described in the original paper.
- **LDC** [2]. We adopt the official implementation and employ the Adam optimizer to train a linear layer with a learning rate of 0.001 for 20 epochs.
- **DP** [5]. Following the official implementation, the parameter γ is set to 200 for CIFAR-100 and 2000 for both Tiny-ImageNet and ImageNet-100, while ϵ is kept at 1×10^{-9} for all experiments.

C.3. Impact of SSD Ratio on Drift Estimator

Here we present the experimental details corresponding to the observations shown in Fig. 2 of the main paper. The experiments are conducted on the ImageNet dataset based on the drift compensation method LDC [2]. Fig. A1 provides a detailed illustration of how different SSD ratios affect the correction of the old class *Chihuahua* during drift estimator training. When the proportion of SSD in the training data is low, a bias emerges that shifts the representation away from the semantically similar new-class *Alsatian*. The drift estimator is essentially a linear network that learns the mapping from the old representation space to the new one. When the proportion of SSD corresponding to a specific class in the training data is low, it becomes difficult to learn a drift estimator that effectively preserves the semantics of that class.

D. More Results

D.1. Results on 5-task Splits

In addition to the 10- and 20-task experiments presented in the main paper, Table A2 reports performance of the compared methods under 5-task splits. Dual-E still shows clear advantages in this setting. For example, on the $\mathcal{A}_{last}$ metric, it outperforms the second-best method by 1.86%, 0.84%, and 0.91% on CIFAR-100, TinyImageNet, and ImageNet-Subset, respectively. This advantage arises from Dual-E's complementary MoE-E and LoR-E, which model local and global transfer patterns and allow the drift compensation process to adaptively balance their contributions. Existing drift compensation methods such as SDC, ADC, LDC, and

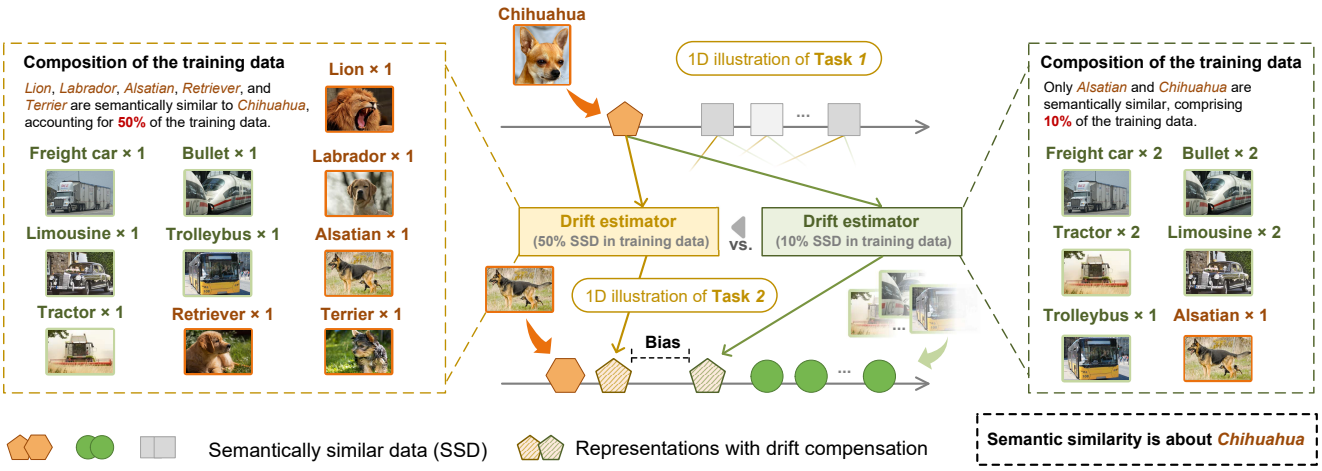

Figure A1. Drift compensation under varying proportions of semantically similar data (SSD). The proportion of SSD used to train the drift estimator influences the calibration of the old-class *Chihuahua*. Reducing SSD lowers the similarity between *Chihuahua* and *Alsatian*. This figure provides a detailed description of Fig. 2 in the main paper.

Table A2. Evaluation of EFCIL methods on three datasets with 5-task splits. Best results are in **bold**, and second-best are underlined.

Method	CIFAR-100		TinyImageNet		ImageNet-Subset	
	$\mathcal{A}_{last}$	$\mathcal{A}_{inc}$	$\mathcal{A}_{last}$	$\mathcal{A}_{inc}$	$\mathcal{A}_{last}$	$\mathcal{A}_{inc}$
LwF [10]	46.26	62.81	38.41	52.34	58.82	69.05
ABD [13]	46.13	61.19	28.79	44.32	51.75	67.31
PASS [18]	48.15	63.32	34.78	45.55	39.27	59.87
FKSR [16]	44.50	60.96	32.29	44.50	47.73	62.61
NAPA-VQ [12]	46.68	61.50	26.67	40.50	38.93	55.22
FCS [9]	45.47	61.93	40.85	54.07	52.29	66.94
EFC [11]	53.13	66.79	38.64	51.23	58.69	71.63
SDC [15]	53.08	65.46	43.52	57.25	65.23	76.35
ADC [3]	58.49	70.92	44.38	57.38	65.77	76.24
LDC [2]	58.54	71.01	44.41	57.49	67.37	76.15
DP [5]	58.85	71.11	44.51	56.78	64.28	75.35
Dual-E (Ours)	**59.37**	**72.97**	**46.36**	**58.33**	**68.53**	**77.26**

DP lack this capability. The limitations of DP stem from its neglect of non-uniform semantic distributions, causing unstable performance under different class orders. Methods like PASS and NAPA-VQ retain intermediate representations and apply augmentation to adjust the classifier head for new tasks, but they do not address feature drift, resulting in suboptimal performance.

Fig. A2 shows the accuracy curves of the compared methods at each incremental stage under 5- and 10-task splits. It can be observed that under longer task sequences, the advantage of Dual-E becomes more pronounced. This indicates that the benefits of drift compensation accumulate with the number of tasks, while the inaccuracies in drift correction of other methods are progressively amplified, causing prototypes to gradually deviate from the oracle.

D.2. Results on Random Class Orders

Dual-E is designed to handle realistic non-uniform semantic distributions and shifts. To validate this, we simulate non-uniform scenarios by varying random seeds for class orderings. Tables A3–A4 report the performance of several drift compensation methods across 11 seeds using two metrics. Dual-E consistently achieves the best results. Although the recent DP method, like Dual-E, updates the drift estimator via an analytical solution, its second-best performance is due to modeling only the global semantic shift trends while ignoring class-specific local transfer patterns.

Table A3. Evaluation of CIFAR-100 (10 tasks) on last task accuracy under class orders perturbed by multiple random seeds.

Method	Seed											Avg.
	0	1	2	3	4	5	6	7	8	9	1993	
Base	42.33	42.80	41.79	43.55	42.44	40.73	40.06	41.82	43.16	40.90	33.93	41.23
SDC	42.47	42.51	41.20	42.58	41.97	40.93	40.21	41.71	42.33	39.69	33.85	40.86
LDC	46.62	46.57	46.80	47.04	47.18	45.45	45.81	45.43	47.10	47.00	45.43	46.40
ADC	47.89	46.17	45.68	47.28	46.94	45.71	45.16	46.11	46.90	46.74	44.32	46.26
DP	45.76	47.96	47.59	47.43	46.86	46.06	46.35	46.61	47.62	46.23	46.25	46.79
Dual-E (Ours)	**48.16**	**48.21**	**48.15**	**48.79**	**47.39**	**46.47**	**47.33**	**47.12**	**47.79**	**47.88**	**47.91**	**47.75**

Table A4. Evaluation of CIFAR-100 (10 tasks) on average incremental accuracy under class orders perturbed by multiple random seeds.

Method	Seed											Avg.
	0	1	2	3	4	5	6	7	8	9	1993	
Base	42.33	42.80	41.79	43.55	42.44	40.73	40.06	41.82	43.16	40.90	33.93	41.23
SDC	42.47	42.51	41.20	42.58	41.97	40.93	40.21	41.71	42.33	39.69	33.85	40.86
LDC	46.62	46.57	46.80	47.04	47.18	45.45	45.81	45.43	47.10	47.00	45.43	46.40
ADC	47.89	46.17	45.68	47.28	46.94	45.71	45.16	46.11	46.90	46.74	44.32	46.26
DP	45.76	47.96	47.59	47.43	46.86	46.06	46.35	46.61	47.62	46.23	46.25	46.79
Dual-E (Ours)	**48.16**	**48.21**	**48.15**	**48.79**	**47.39**	**46.47**	**47.33**	**47.12**	**47.79**	**47.88**	**47.91**	**47.75**

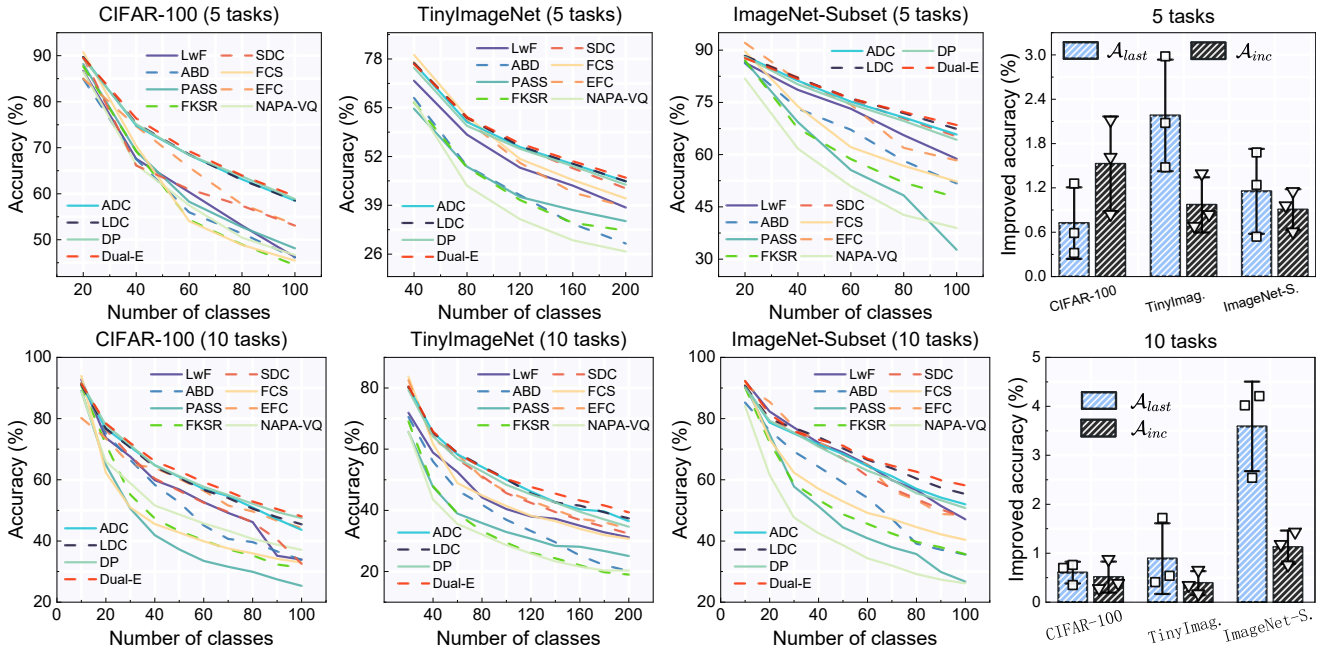

Figure A2. The line charts show the accuracy at each incremental stage under the 5- and 10-task split. The bar charts show the improvement of Dual-E over the second-best results. Experiments are repeated three times with different class orders, and the mean is reported.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009.

[2] Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D.

Bagdanov, Bartlomiej Twardowski, and Joost van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *Computer Vision – ECCV 2024*, page 473–490, 2024.

[3] Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bartłomiej Twardowski, and Joost Van De Weijer. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28525–28534, 2024.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[5] Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic shift estimation via dual-projection and classifier reconstruction for exemplar-free class-incremental learning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, pages 22392–22406, 2025.

[6] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[7] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[8] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

[9] Qiwei Li, Yuxin Peng, and Jiahuan Zhou. FCS: Feature calibration and separation for non-exemplar class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28495–28504, 2024.

[10] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[11] Simone Magistri, Tomaso Trinci, Albin Soutif, Joost van de Weijer, and Andrew D. Bagdanov. Elastic feature consolidation for cold start exemplar-free incremental learning. In *International Conference on Learning Representations (ICLR)*, 2024.

[12] Tamasha Malepathirana, Damith Senanayake, and Saman Halgamuge. NAPA-VQ: Neighborhood aware prototype augmentation with vector quantization for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11640–11650, 2023.

[13] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9354–9364, 2021.

[14] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[15] Lu Yu, Bartłomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6980–6989, 2020.

[16] Jiang-Tian Zhai, Xialei Liu, Lu Yu, and Ming-Ming Cheng. Fine-grained knowledge selection and restoration for non-exemplar class incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6971–6978, 2024.

[17] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: a python toolbox for class-incremental learning. *SCIENCE CHINA Information Sciences*, 66(9):197101, 2023.

[18] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5867–5876, 2021.