SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

July 2026, Vol. 69, Iss. 7, 172106:1–172106:14
https://doi.org/10.1007/s11432-024-4893-x

Text-guided bidirectional mapping distillation for continual semantic segmentation

Xuze HAO, Xuhao JIANG, Wenqian NI, Weimin TAN* & Bo YAN*

College of Computer Science and Artificial Intelligence, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai 201203, China

Received 23 August 2024/Revised 1 March 2025/Accepted 15 April 2025/Published online 8 June 2026

Abstract Continual semantic segmentation (CSS) has been extensively studied to address the challenge of catastrophic forgetting, which refers to the significant drop in a model's performance on old classes when learning new ones. Recently, CSS methods have mainly utilized knowledge distillation to tackle this problem. However, most distillation-based methods directly constrain the output of the new model to be similar to that of the old model, overlooking the need for plasticity. In this work, we explore a semantic relationship between old and new classes, which can be applied to facilitate knowledge transfer. Specifically, we leverage the text embeddings of image-level labels to model this relationship. Guided by this semantic relationship, we introduce a novel bidirectional mapping distillation that transfers old knowledge forward to facilitate learning new classes while transferring new knowledge backward to resist forgetting. By employing the text-guided bidirectional mapping distillation, our model achieves a better trade-off between stability and plasticity. Extensive experiments on the PASCAL VOC 2012 and ADE20K datasets under various CSS scenarios demonstrate that our method achieves competitive performance.

Keywords deep learning, continual learning, semantic segmentation, knowledge distillation, text embeddings

Citation Hao X Z, Jiang X H, Ni W Q, et al. Text-guided bidirectional mapping distillation for continual semantic segmentation. Sci China Inf Sci, 2026, 69(7): 172106, https://doi.org/10.1007/s11432-024-4893-x

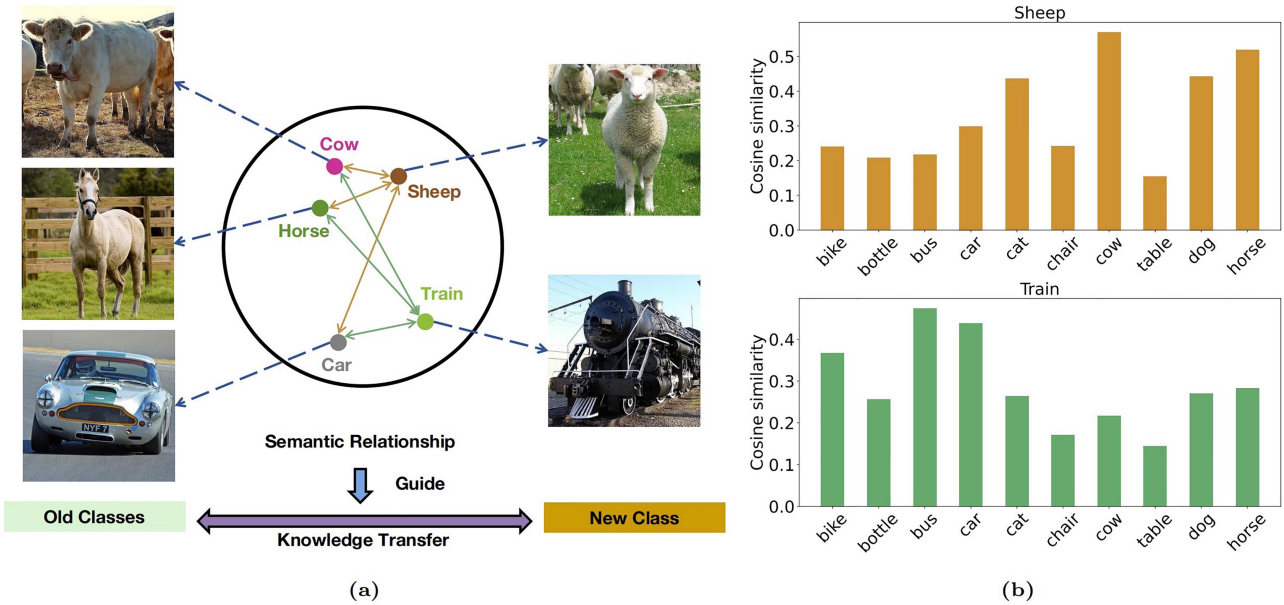
1 Introduction

Semantic segmentation, which aims to assign a class label to each pixel in an image, is a fundamental task in computer vision. In recent years, deep learning methods have achieved excellent performance in semantic segmentation [1–6]. However, these methods are primarily designed for closed-set scenarios, where they can only segment a fixed number of predefined classes, requiring the entire dataset to be presented simultaneously. In a more realistic scenario, deep learning models must be enabled to continuously learn new knowledge without a stable distribution, known as continual learning. Continual semantic segmentation (CSS) applies the continual learning paradigm to semantic segmentation. The primary objective of CSS is to incorporate new knowledge continually (i.e., plasticity) while preserving the discriminative ability for existing knowledge (i.e., stability).

The main challenge in CSS is catastrophic forgetting [7, 8]; i.e., the previously acquired knowledge tends to be interfered with or even completely forgotten after new classes are learned. To address this challenge, numerous methods have been proposed for CSS. Currently, most methods [9, 10] utilize knowledge distillation to maintain crucial information for old classes. For example, PLOP [9] distills long-range and short-range spatial relations via local-pooled-output distillation. Similarly, RCIL [10] introduces average pooling distillation across spatial and channel dimensions to preserve multi-scale features. Although these CSS methods effectively prevent forgetting (strong stability), they typically lack the transferability of knowledge to facilitate learning new classes (weak plasticity), thus suffering from the stability-plasticity dilemma [11].

Figure 1(a) illustrates the shared semantic information between the old and new classes within the dataset. For example, the new class *sheep* shares semantic attributes such as "body shape" and "four legs" with the old classes *cow* and *horse*. Similarly, the new class *train* exhibits similarity with the old class *car*, sharing the attribute "wheels." Based on the prior knowledge of old classes, humans demonstrate a remarkable ability to recognize new classes by leveraging shared attributes between old and new classes [12]. Therefore, inspired by this cognitive ability, we propose exploiting the semantic relationship between new and old classes to guide knowledge transfer across different incremental steps in CSS, facilitating new class learning while mitigating forgetting.

* Corresponding author (email: wmtan@fudan.edu.cn, byan@fudan.edu.cn)



(a) (b)

Figure 1 (Color online) (a) Motivation illustration. The new class *sheep* shares semantic properties (e.g., "body shape" and "four legs") with old classes *cow* and *horse*, while "wheel" serves as a common attribute between the new class *train* and the old class *car*. Correspondingly, there exists a semantic relationship between old and new classes, which can be utilized to facilitate knowledge transfer in CSS. (b) Text-based semantic similarity between two new classes (*sheep* and *train*) and ten old classes (*bike*, *bottle*, *bus*, *car*, *cat*, *chair*, *cow*, *table*, *dog*, and *horse*). The new class *sheep* exhibits high similarity with *animal* classes, and *train* shows high similarity with *vehicle* classes, while both display low similarity with semantically unrelated classes.

In this paper, we leverage text embeddings (e.g., BERT [13]) of image-level labels, which are cost-effective to obtain, as semantic information to measure similarity. As illustrated in Figure 1(b), the new classes exhibit high similarity with semantically related old classes and weak similarity with unrelated classes. This pattern validates the effectiveness of text embeddings in capturing semantic relationships between old and new classes. We then introduce a novel bidirectional mapping distillation to transfer knowledge based on the semantic relationship. In the forward mapping distillation, we transfer old knowledge to the new classes, facilitating learning new tasks. Simultaneously, in the backward mapping distillation, we extract the semantic relationship and transfer new knowledge back to the old classes, effectively overcoming catastrophic forgetting. By optimizing this bidirectional mapping distillation, our model achieves a better balance between plasticity and stability in CSS. Extensive experiments demonstrate the superiority of our method over state-of-the-art (SOTA) approaches on two benchmarks, i.e., PASCAL VOC 2012 and ADE20K.

In summary, our main contributions are as follows.

- We propose to exploit the semantic relationship between old and new classes by using text embeddings of image-level labels in CSS, which helps facilitate knowledge transfer.
- Based on the semantic relationship, we propose a novel bidirectional mapping distillation to strike a better trade-off between stability and plasticity. The forward mapping distillation encourages the transfer of knowledge from the old classes to the new ones, efficiently facilitating adaptation to new tasks. Meanwhile, the backward mapping distillation transfers new knowledge backward to prevent forgetting among the old classes.
- We conduct extensive experiments on two benchmark datasets: PASCAL VOC 2012 and ADE20K. The results demonstrate that our method achieves the best performance in CSS.

2 Related work

2.1 Semantic segmentation

Semantic segmentation is a foundational problem in computer vision, tasked with segmenting objects and scenes in images and assigning them semantic classifications. The advancement of deep neural networks has significantly enhanced semantic segmentation capabilities. In the early years, semantic segmentation methods primarily relied on fully convolutional networks (FCNs) [14], which can accept input images of arbitrary sizes and incorporate more spatial information to achieve impressive results. To address the limited receptive field of these models, numerous

approaches have employed multi-scale feature fusion. EFCN [15] improves segmentation results by preserving low-level spatial information and integrating context for high-level features. PSPNet [16] introduces a pyramid pooling module to aggregate multi-scale context across convolution layers. The DeepLab series [2, 17, 18] uses an atrous spatial pyramid pooling module, which has gained popularity for preserving richer long-range contextual information. Subsequently, follow-up efforts shifted to integrating the attention mechanism into the network [4, 19–22]. The Non-local Net [19] utilizes self-attention modules to capture long-range dependencies. More recently, transformer-based architectures [23–26] have demonstrated significant improvements by aggregating features at different scales. However, these studies assume that all data are concurrently available for training, which may not be feasible in open-world scenarios. Therefore, a more realistic setting needs to be considered to address this limitation.

2.2 Continual learning

Continual learning aims to train a model to continuously learn new concepts over a sequence of data [27–30]. The main challenge in continual learning is catastrophic forgetting while adapting to new tasks. To address this problem, various methods have been proposed, which can be roughly divided into three categories: rehearsal-based, structure-based, and distillation-based methods. Rehearsal-based methods [31–33] typically store a small number of exemplars from previous steps and rehearse them when learning new classes. However, the performance of these methods degrades as the memory buffer size decreases or the number of learned classes increases significantly. Moreover, previous data may be inaccessible because of safety and privacy limitations [34]. Some other studies [35,36] utilize large generative models to synthesize samples for replay, but training such models for continual learning can be inefficient and challenging. Structure-based methods [37–39] maintain the learned parameters related to old classes fixed and dynamically extend the network structure to acquire new knowledge. However, the growing network architecture increases training costs. Distillation-based methods [9, 40–42] employ knowledge distillation to force the outputs of the new model to approximate those of the old model, which can prevent the forgetting of previous knowledge. In this paper, we aim to address continual learning from a distillation-based perspective without accessing previous data.

2.3 Continual semantic segmentation

Recently, interest in the field of CSS has been growing [9, 10, 32, 40, 41, 43–47]. Previous studies generally adopt the knowledge distillation technique to alleviate catastrophic forgetting in CSS. For example, MiB [40] analyzes the background shift problem for the first time and applies unbiased knowledge distillation to mitigate this issue. Douillard et al. [9] proposed the PLOP method, which introduces a confidence-based pseudo-labeling strategy to address background shift and employs multi-scale spatial distillation on intermediate feature maps. RCIL [10] proposes a pooled cube distillation mechanism on channel and spatial dimensions, effectively mitigating the adverse effects of errors and noise in local feature maps. DKD [41] proposes a decomposed knowledge distillation to explicitly impose constraints on positive and negative reasoning scores, instead of class logits. Furthermore, EWF [43] adopts a weight fusion strategy to enhance existing distillation-based methods. However, most existing distillation-based methods focus solely on constraining the new model's outputs to match the old ones, which leads models to retain the discriminative ability for old knowledge (strong stability) but lack flexibility in adapting to new classes (weak plasticity). In contrast, we propose modeling the semantic relationship between old and new classes by leveraging the text embeddings of image-level labels, which can guide the distillation process to effectively facilitate knowledge transfer across different steps, thereby striking a better balance between stability and plasticity.

3 Method

3.1 Problem definition and notation

In CSS, a sequence of tasks $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, ..., \mathcal{D}_T\}$ is introduced to the model over time. At each step t, the model can only access a training dataset $\mathcal{D}_t$ comprising a set of pairs (X^t, Y^t), where X^t denotes an input image of size $H \times W$, and Y^t is the corresponding ground truth segmentation map. The label space of dataset $\mathcal{D}_t$ is denoted by $\mathcal{C}^t$, where $\mathcal{C}^i$ and $\mathcal{C}^j$ are disjoint for all $i \neq j$, except for the background class b. This condition implies that all other classes (old classes $\mathcal{C}^{1:t-1}$ and future classes $\mathcal{C}^{t+1:T}$) are labeled as the background class b when training on $\mathcal{D}_t$. Once the training at step t is completed, the model should be able to segment all seen classes $\mathcal{C}^{1:t}$.

The model at step t can be assumed as $f^t = f^t_\theta \circ f^t_\phi(\cdot)$, where f^t_ϕ represents the backbone to extract features, and f^t_θ represents the classifier. The predicted segmentation logit map can be denoted by $V^t = f^t(X^t)$.

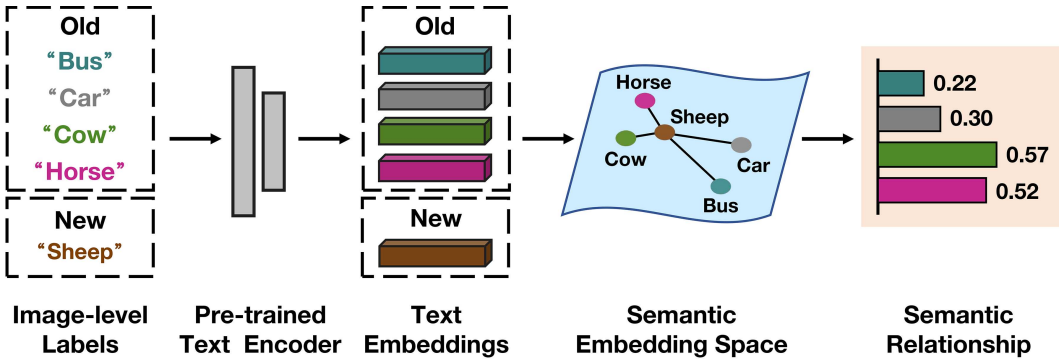


Figure 2 (Color online) Illustration of building the semantic relationship between old (e.g., *bus*, *car*, *cow*, and *horse*) and new classes (e.g., *sheep*) with the text embeddings of the image-level labels.

3.2 Bidirectional mapping distillation

Motivated by the relevance between old and new classes, we propose to leverage the semantic relationship between them to facilitate learning in CSS. To this end, as shown in Figure 2, we take the image-level labels of classes as input and generate text embeddings as their semantic representations by utilizing a pre-trained text embedding method (e.g., Word2Vec [48], GloVe [49], or BERT [13]). Here, the similarity between two text embeddings is expected to measure the semantic relationship between their corresponding classes. Specifically, the normalized similarity between a specific new class $i \in \mathcal{C}^t \setminus \{b\}$ and any old class $c \in \mathcal{C}^{1:t-1}$ is defined as follows:

$$\mathrm{Sim}\,(i,c) = \frac{\exp\langle E_i, E_c\rangle}{\sum_{j\in\mathcal{C}^{1:t-1}} \exp\langle E_i, E_j\rangle}, \tag{1}$$

where E_i is the text embedding for the i-th class, and $\langle\cdot,\cdot\rangle$ denotes cosine similarity between two embeddings. Here, the particularity of the background class in CSS should be noted. To address the issue of background shift [40], we set the similarity between the background class and any new class to 1 before normalization, i.e., $\langle E_i, E_b\rangle = 1$, $i \in \mathcal{C}^t \setminus \{b\}$.

Aided by the semantic relationship between the old and new classes, the model can effectively transfer old knowledge forward to facilitate learning new classes. Simultaneously, we can transfer the acquired knowledge of new classes to old classes, thereby mitigating the problem of forgetting. As a result, we employ a semantically guided distillation in two directions, i.e., forward and backward mapping distillation.

Forward mapping distillation. For forward mapping distillation, the primary objective is to facilitate the learning of new classes. To achieve this, we first design the forward semantic mapping to build the mapping outputs $\hat{V}^t$ by reusing the predictions of the old model V^{t-1} based on the semantic relationship. In the implementation, the mapping outputs of the old classes remain unchanged, while those of the new classes are constructed by aggregating the weighted outputs of all old classes (see Figure 3, blue module). Thus, the forward mapping operation can be expressed as follows:

$$\hat{V}_i^t = \begin{cases} V_i^{t-1}, & \text{if } i \in \mathcal{C}^{1:t-1}, \\ \sum\limits_{j\in\mathcal{C}^{1:t-1}} \mathrm{Sim}\,(i,j)\, V_j^{t-1}, & \text{if } i \in \mathcal{C}^t \setminus \{b\}\,. \end{cases} \tag{2}$$

Then, we distill the forward mapping outputs $\hat{V}^t$ into the current model:

$$\mathcal{L}_{fmd} = \frac{1}{HW} \sum_{h,w}^{H,W} \sum_{i\in\mathcal{C}^{1:t}} \left\| V_i^t(h,w) - \hat{V}_i^t(h,w) \right\|, \tag{3}$$

which forces the current model to predict like the mapping outputs $\hat{V}^t$. Similar to standard knowledge distillation [50], Eq. (3) aligns the predictions of the old and new models for the old classes, which helps mitigate the problem of forgetting. Meanwhile, forward mapping distillation enables the transfer of old knowledge to new classes based on the semantic relationship, facilitating the learning of the new classes.

Backward mapping distillation. For backward mapping distillation, we utilize the semantic relationship to transport the predictions on new classes to old ones and distill them into the old model, which acts as an additional constraint to help overcome forgetting. In this case, we construct the backward mapping outputs $\hat{V}^{t-1}$ for a specific

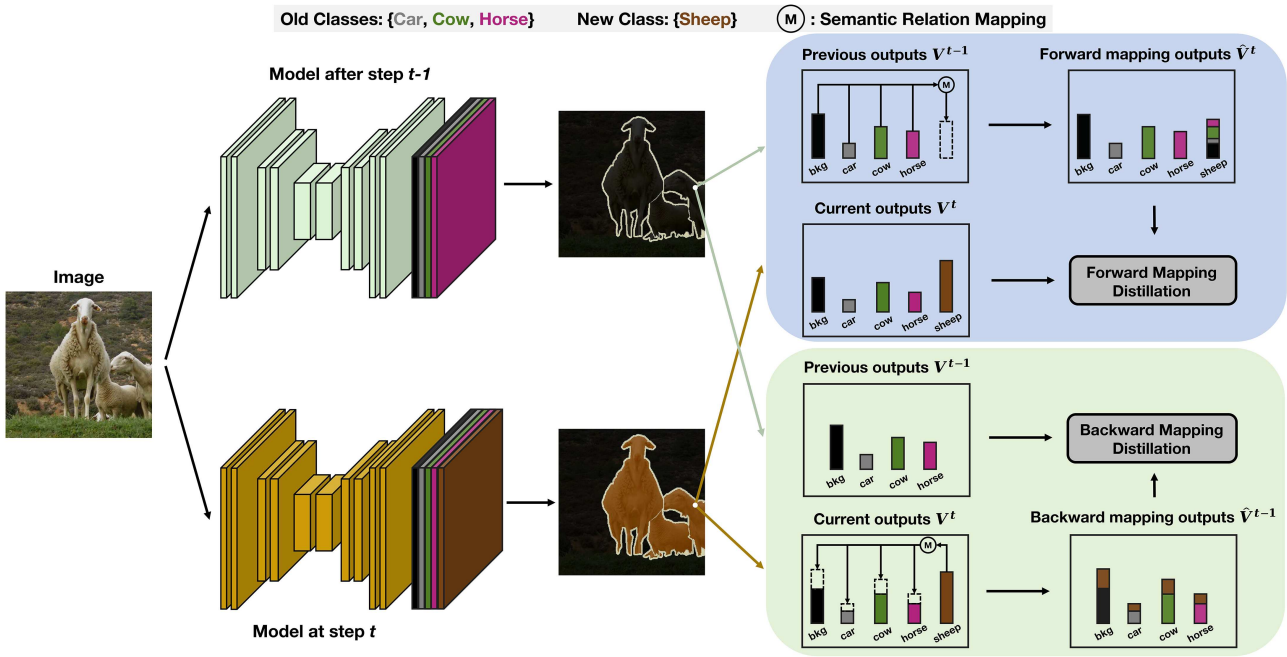


Figure 3 (Color online) Illustration of our method with a set of old classes (e.g., *car*, *cow*, and *horse*) and a new class (e.g., *sheep*). By leveraging the semantic relationship, we apply forward mapping distillation (blue module) to facilitate the learning of new classes by transferring old knowledge forward. Simultaneously, we employ backward mapping distillation (green module) to prevent forgetting among old classes by transferring new knowledge backward.

old class by summing the original outputs of the corresponding class with the weighted outputs of all new classes (see Figure 3, green module). The backward mapping operation can be defined as follows:

$$\hat{V}_i^{t-1} = V_i^t + \sum_{j \in \mathcal{C}^t \setminus \{b\}} \mathrm{Sim}\,(j,i)\, V_j^t, \quad i \in \mathcal{C}^{1:t-1}. \tag{4}$$

The backward mapping distillation distills the mapping outputs $\hat{V}^{t-1}$ into the old model:

$$\mathcal{L}_{bmd} = \frac{1}{HW} \sum_{h,w}^{H,W} \sum_{i \in \mathcal{C}^{1:t-1}} \left\| V_i^{t-1}(h,w) - \hat{V}_i^{t-1}(h,w) \right\|. \tag{5}$$

Optimizing (5) forces the mapping outputs $\hat{V}^{t-1}$ to be similar to the outputs of the old model V^{t-1}, thus enabling the backward transfer of new knowledge to prevent forgetting on the old classes.

The bidirectional mapping distillation loss is the combination of (3) and (5):

$$\mathcal{L}_{bdmd} = (1-\lambda)\mathcal{L}_{fmd} + \lambda \mathcal{L}_{bmd}\,, \tag{6}$$

where $\lambda = (e/N)^2$ is an adjustable weighting parameter. Here, e and N denote the current epoch index and the total number of epochs, respectively. Specifically, this parameter assigns greater weight to forward mapping distillation in the early stage of training to facilitate learning new classes at the beginning, while continuously increasing the importance of backward mapping distillation during training to preserve old knowledge.

3.3 Overall loss

To continually learn new classes, we employ a multiple binary cross-entropy (mBCE) loss $\mathcal{L}_{mbce}$ as the supervised segmentation loss, following [41], formulated as follows:

$$\mathcal{L}_{mbce} = -\frac{1}{HW} \sum_{h,w}^{H,W} \sum_{i \in \mathcal{C}^t} \gamma\, Y_i^t(h,w) \log \tilde{V}_i^t\,(h,w) + \left(1 - Y_i^t(h,w)\right) \log \left(1 - \tilde{V}_i^t\,(h,w)\right), \tag{7}$$

where $\tilde{V}^t = \sigma(V^t)$, σ is the logistic function, and γ is a positive weight for handling the imbalance between two terms in (7). Empirically, we set γ to 2 and 30 for PASCAL VOC 2012 and ADE20K, respectively, as in [41, 51]. The overall loss of our method is then computed as follows:

$$\mathcal{L} = \mathcal{L}_{mbce} + \alpha \mathcal{L}_{bdmd}, \tag{8}$$

where α is the balancing hyperparameter.

4 Experiments

4.1 Experimental setups

Datasets. We conduct extensive experiments for quantitative and qualitative evaluations on two standard semantic segmentation datasets: PASCAL VOC 2012 [52] and ADE20K [53]. PASCAL VOC 2012 comprises 10582 training images and 1449 validation images, spanning 21 classes, including 20 semantic classes and 1 background class. ADE20K is a much more challenging dataset, providing 20210 training images and 2000 validation images across 150 classes.

CSS protocols. Following previous studies [9, 32, 41], we evaluate our method on PASCAL VOC 2012 with 4 scenarios: 19-1 (2 tasks), 15-5 (2 tasks), 15-1 (6 tasks), and a more challenging 10-1 (11 tasks). For ADE20K, we conduct experiments with scenarios of 100-50 (2 tasks), 50-50 (3 tasks), and 100-10 (6 tasks). Each scenario is denoted by *A*-*B*, where *A* denotes the number of classes in the first task, and *B* denotes the number of newly added classes in each incremental task.

For all scenarios on both datasets, Ref. [40] introduced two different settings for CSS: *Disjoint* and *Overlapped*. In the *Disjoint* setting, each pixel in the images is assumed to belong exclusively to either the previous classes $\mathcal{C}^{1:t-1}$ or the current classes $\mathcal{C}^t$, without including any pixels from future classes $\mathcal{C}^{t+1:T}$. The *Overlapped* setting comprises all images containing at least one pixel belonging to the current classes $\mathcal{C}^t$, thereby allowing future classes to appear in the current training images. As the *Overlapped* setting is more challenging and realistic, we evaluate our method for CSS only in this setting, as in previous approaches [9, 42, 43].

Metrics. We employ mean intersection over union (mIoU) as the metric for evaluating model performance. Specifically, we report mIoU scores for the old, new, and all classes to measure the model's stability and plasticity and the trade-off between them, respectively. Notably, a model with excellent performance on old classes but poor performance on new classes can still exhibit a good average result across all classes because of the high proportion of old classes within the overall class distribution. To address this issue, we introduce the harmonic mean (*HM*) of mIoU scores for the old and new classes, which can better demonstrate the overall performance of CSS by mitigating the impact of the imbalance between them.

Baselines. To validate the effectiveness of our proposed method, we conduct comparisons with SOTA CSS approaches, including ILT [50], MiB [40], PLOP [9], SSUL [32], ST [47], RCIL [10], DKD [41], EWF [43], LGKD [42], PGSD [54], and LAG [46]. For fair benchmarking, all selected methods are evaluated without utilizing rehearsed data.

Implementation details. Following [9, 32, 40], we employ the DeepLab-v3 [2] architecture with ResNet-101 [55] pre-trained on ImageNet [56] as our segmentation backbone. We use the SGD optimizer with a momentum of 0.9. In the initial step, our method is trained for 60 epochs on PASCAL VOC 2012 with a learning rate of 0.001 and for 100 epochs on ADE20K with a learning rate of 0.0025. Subsequently, in the incremental steps, the learning rate is reduced by a factor of 10 to 0.0001 and 0.00025 on the two datasets, respectively. We report on the final performance on the standard validation sets. Our proposed method is implemented with PyTorch [57] on two NVIDIA GeForce RTX 3090 GPUs. For all experiments, we set $\alpha = 0.01$. The details of the hyperparameter setting are included in Subsection 4.3.

4.2 Comparison with SOTA methods

PASCAL VOC 2012. The quantitative results of the last step on PASCAL VOC 2012 are presented in Table 1. Our method exhibits considerable advancements in stability and plasticity over SOTA approaches across all scenarios. For short-step learning, we outperform LGKD by 1.2% (0-19) and 2.9% (20) in terms of mIoU on VOC 19-1 (2 tasks). On VOC 15-5 (2 tasks), our method improves *HM* by 5.6% over LAG and by 1.8% over LGKD. For the long-step 15-1 (6 tasks) setting, we achieve mIoU and *HM* gains of 2.0% and 5.0% over the SOTA methods, respectively. Furthermore, we evaluate on a more challenging VOC 10-1 setting (11 tasks), where our method considerably outperforms existing approaches in terms of mIoU and *HM*, demonstrating its effectiveness in

Table 1 CSS results on PASCAL VOC 2012 for the different scenarios in terms of IoU (%). Numbers in bold denote the best results, and underlined numbers denote the second-best results.

Method	19-1 (2 tasks)				15-5 (2 tasks)				15-1 (6 tasks)				10-1 (11 tasks)			
	0-19	20	all	*HM*	0-15	16-20	all	*HM*	0-15	16-20	all	*HM*	0-10	11-20	all	*HM*
ILT (ICCVW2019)	67.8	10.9	65.1	18.7	67.1	39.2	60.5	49.5	8.8	8.0	8.6	8.4	7.2	3.7	5.5	4.9
MiB (CVPR2020)	71.4	23.6	69.2	35.5	76.4	50.0	70.1	60.4	34.2	13.5	29.3	19.4	12.3	13.1	12.7	12.7
PLOP (CVPR2021)	75.4	37.4	73.5	49.9	75.7	51.7	70.1	61.5	65.1	21.1	54.6	31.9	44.0	15.5	30.5	22.9
SSUL (NeurIPS2021)	77.7	29.7	75.4	43.0	77.8	50.1	71.2	61.0	77.3	36.6	67.6	49.7	71.3	46.0	59.3	55.9
ST (TNNLS2022)	76.1	43.4	74.5	55.3	76.7	54.3	71.1	63.6	71.4	40.0	63.6	51.3	–	–	–	–
RCIL (CVPR2022)	77.0	31.5	74.7	44.7	78.8	52.0	72.4	62.7	70.6	23.7	59.4	35.5	55.4	15.1	34.3	23.7
DKD (NeurIPS2022)	77.8	41.5	76.0	54.1	78.8	58.2	73.9	67.0	78.1	42.7	69.7	55.2	73.1	46.5	60.4	56.8
EWF (CVPR2023)	77.8	12.2	74.7	21.1	–	–	–	–	78.0	25.5	65.5	38.4	56.0	16.7	37.3	25.7
LGKD (ICCV2023)	77.3	42.9	75.7	55.2	79.5	56.1	73.9	65.8	70.6	30.9	61.1	43.0	–	–	–	–
PGSD (TCSVT2024)	77.6	41.8	75.9	54.3	78.7	57.3	73.6	66.3	76.7	40.9	68.2	53.4	70.9	45.4	58.8	55.4
LAG (TPAMI2024)	–	–	–	–	77.3	51.8	71.2	62.0	75.0	37.5	66.1	50.0	69.6	42.6	56.7	52.9
Ours	**78.5**	**45.8**	**76.9**	**57.8**	**79.7**	**58.7**	**74.7**	**67.6**	**78.9**	**48.7**	**71.7**	**60.2**	**73.2**	**51.6**	**62.9**	**60.5**

Table 2 CSS results on ADE20K for the different scenarios in terms of IoU (%). Numbers in bold denote the best results, and underlined numbers denote the second-best results.

Method	100-50 (2 tasks)				50-50 (3 tasks)				100-10 (6 tasks)			
	0-100	101-150	all	*HM*	0-50	51-150	all	*HM*	0-100	101-150	all	*HM*
ILT (ICCVW2019)	18.3	14.4	17.0	16.1	3.5	12.9	9.7	5.5	0.1	3.1	1.1	0.2
MiB (CVPR2020)	40.5	17.2	32.8	24.1	45.6	21.0	29.3	28.8	38.2	11.1	29.2	17.2
PLOP (CVPR2021)	41.9	14.9	32.9	22.0	48.8	21.0	30.4	29.4	40.5	13.6	31.6	20.4
SSUL (NeurIPS2021)	41.3	18.0	33.6	25.1	48.4	20.2	29.6	28.5	40.2	18.8	33.1	25.6
ST (TNNLS2022)	40.7	24.0	35.1	30.2	40.0	23.6	29.0	29.6	33.6	16.9	28.1	22.5
RCIL (CVPR2022)	42.3	18.8	34.5	26.0	48.3	25.0	32.5	32.9	39.3	17.6	32.1	24.3
DKD (NeurIPS2022)	42.4	22.9	36.0	29.7	48.8	26.3	33.9	34.2	41.6	19.5	34.3	26.6
EWF (CVPR2023)	41.2	21.3	34.6	28.1	–	–	–	–	41.5	16.3	33.2	23.4
LGKD (ICCV2023)	43.4	25.7	37.5	32.3	48.9	**29.4**	36.0	36.7	41.9	22.0	35.4	28.9
PGSD (TCSVT2024)	41.2	20.3	34.3	27.2	46.5	23.2	31.1	31.0	40.4	17.2	32.7	24.1
LAG (TPAMI2024)	41.6	19.7	34.3	26.8	47.7	26.1	33.3	33.8	41.0	18.7	33.6	25.7
Ours	**43.5**	**26.5**	**37.9**	**32.9**	**49.6**	29.2	**36.1**	**36.8**	**42.1**	**23.8**	**36.0**	**30.4**

handling longer continual learning settings. The performance at each step on VOC 15-1 and VOC 10-1 is depicted in Figure 4, clearly illustrating the consistent superiority of our method over other baselines throughout the continual learning process.

ADE20K. As shown in Table 2, our method exhibits superior performance across various experimental settings on the ADE20K dataset. For the short 100-50 (2 tasks) setting, we achieve improvements of 5.2% and 3.3% over EWF in mIoU for new and all classes. On ADE 50-50 (3 tasks), our method obtains substantial increases of 3.0% and 3.1% in mIoU for new and all classes, respectively, when compared with LGKD. On the long 100-10 (6 tasks) setting, our method surpasses the SOTA approach by 1.8% on newly learned classes (101–150) and achieves a 1.5% improvement on *HM*. These experimental results on ADE20K further demonstrate the superiority of our method in learning new classes and its effectiveness in achieving a better stability-plasticity trade-off.

4.3 Ablation study

Effect of each component. Table 3 presents an ablation study on VOC 15-1 for analyzing the contribution of each component of our proposed method. The baselines (rows 1 and 2) consist of $\mathcal{L}_{mbce}$ combined with $\mathcal{L}_{kd}$ (the standard KD proposed in ILT [50]) and $\mathcal{L}_{unkd}$ (unbiased KD introduced in MiB [40]), respectively. Replacing both baseline KD losses with our forward and backward mapping distillation losses (rows 3 and 4) leads to significant performance improvements. Finally, combining all components (row 5) further improves performance from 68.5% to 71.7% (+3.2 percentage points). In conclusion, our ablation study verifies that each component plays a critical role in achieving the best performance.

Impact of different class similarity modeling methods. In this ablation study, we investigate the impact of uniform similarity (Uniform), learnable similarity (Learnable), and four different text embedding methods

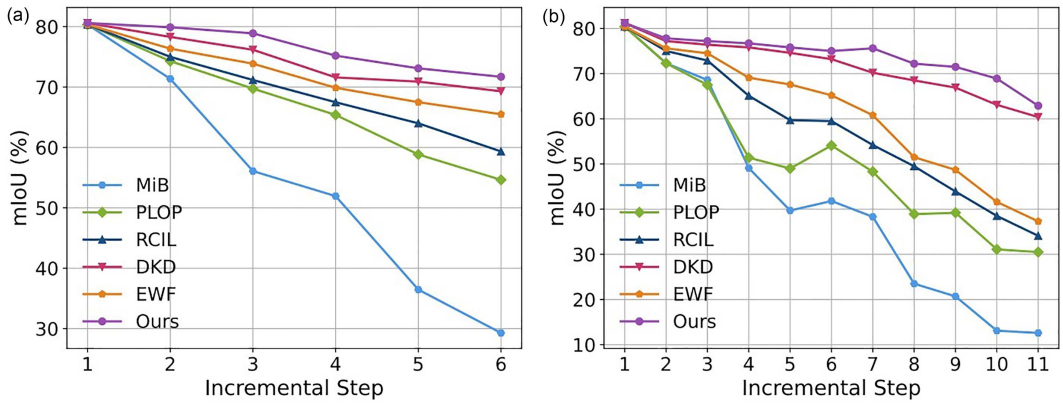


Figure 4 (Color online) mIoU visualization over steps on VOC 15-1 (a) and VOC 10-1 (b).

Table 3 Ablation study of the proposed components in our method on VOC 15-1. Numbers in bold denote the best results.

$\mathcal{L}_{mbce}$	$\mathcal{L}_{kd}$	$\mathcal{L}_{unkd}$	$\mathcal{L}_{bdmd}$		15-1 (6 tasks)			
			$\mathcal{L}_{fmd}$	$\mathcal{L}_{bmd}$	0-15	16-20	all	*HM*
✓	✓	✗	✗	✗	75.4	36.4	66.1	49.1
✓	✗	✓	✗	✗	76.7	36.8	67.2	49.7
✓	✗	✗	✓	✗	77.2	46.3	69.8	57.9
✓	✗	✗	✗	✓	77.5	39.7	68.5	52.5
✓	✗	✗	✓	✓	**78.9**	**48.7**	**71.7**	**60.2**

Table 4 Performance of different text embedding methods on VOC 15-1. Numbers in bold denote the best results.

Method	15-1 (6 tasks)			
	0-15	16-20	all	*HM*
Baseline	75.4	36.4	66.1	49.1
Uniform	76.9	40.3	68.2	52.9
Learnable	77.4	43.7	69.4	55.9
Word2Vec	78.1	48.1	71.0	59.5
GloVe	77.9	47.9	70.8	59.3
CLIP	78.2	48.3	71.1	59.7
BERT	**78.9**	**48.7**	**71.7**	**60.2**

Table 5 Per-class IoU (%) on PASCAL VOC 2012. Numbers in bold denote the best results.

	bkg	*airplane*	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*motorbike*	*person*	*plant*	*sheep*	*sofa*	*train*	*tvmonitor*	all
MiB	82.8	8.9	28.1	39.8	6.6	39.2	5.5	14.7	72.5	7.6	37.5	32.6	32.2	42.8	23.5	81.4	1.5	18.1	15.4	14.3	18.2	29.3
PLOP	80.3	77.5	29.5	61.6	58.7	64.1	79.4	77.9	80.1	30.4	62.8	53.8	76.5	67.0	72.1	70.3	14.6	45.5	11.0	24.0	10.4	54.6
RCIL	82.7	82.2	39.0	76.5	60.6	70.8	84.2	85.6	86.3	21.5	71.0	51.7	77.7	77.0	84.1	75.9	21.4	45.6	16.6	26.5	9.2	59.3
DKD	87.4	89.3	**40.7**	88.8	70.1	79.9	90.6	89.0	**93.1**	37.1	80.1	**62.7**	89.3	84.8	84.4	86.2	38.0	59.4	20.9	62.0	46.5	70.5
EWF	79.8	87.9	40.2	88.3	67.3	**82.1**	90.2	89.6	92.5	**39.2**	77.2	60.5	88.6	86.7	**85.5**	**86.8**	15.8	28.2	17.5	34.5	37.1	65.5
Ours	**89.9**	**89.5**	40.4	**88.9**	**70.2**	80.1	**91.3**	**90.2**	91.9	35.3	**83.4**	61.3	**90.7**	**88.4**	**85.5**	85.9	**38.8**	**62.4**	**27.7**	**65.9**	**48.9**	**71.7**

(Word2Vec [48], GloVe [49], BERT [13], and CLIP [58]) on the final performance. As shown in Table 4, while Uniform or Learnable can improve upon the baseline, both yield lower performance than the text embedding methods do. Among the embedding methods, the results indicate relatively similar performance, with a slight improvement observed when utilizing BERT. Specifically, employing BERT achieves an mIoU of 71.7% across all classes, compared with approximately 71.0% obtained with the other three methods. Hence, we select BERT as the text embedding method for all conducted experiments.

Per-class results. The per-class IoU results on VOC 15-1 are presented in Table 5. Our method achieves the highest IoU on most classes, particularly for new classes. This superior performance demonstrates that SOTA approaches primarily focus on enforcing output matching between old and new models, thus struggling to adapt to new classes. In contrast, our method leverages text embeddings to model semantic relationships and guide bidirectional knowledge transfer, effectively facilitating the distinction of new classes while preserving existing knowledge.

Analysis of computational cost. We analyze the efficiency of our method by evaluating the per-step training

Table 6 Training time cost and memory usage comparisons on VOC 15-1.

Method	Training time cost in each step (min)					Memory usage (MB)	mIoU (%)	all (↑)
	1	2	3	4	5	Avg (↓)	Avg (↓)	
MiB	11.0	7.3	10.7	10.8	11.6	10.3	13154	29.3
PLOP	13.9	10.6	13.5	13.2	14.0	13.0	17658	54.6
RCIL	16.6	12.1	14.9	15.5	16.3	15.1	20214	59.4
DKD	19.7	11.5	19.1	18.4	19.3	17.6	16752	69.7
EWF	11.8	8.9	11.1	11.7	12.5	11.2	14624	65.5
Ours	11.4	7.8	11.1	11.6	12.2	10.8	14096	71.7

Table 7 Performance of five different class orderings on VOC 15-1. Numbers in bold denote the best results.

Method	A	B	C	D	E	Avg ± Std
MiB	29.3	20.9	36.8	38.9	53.3	35.8 ± 10.8
PLOP	54.6	48.1	52.6	58.3	48.1	52.4 ± 3.9
RCIL	59.4	54.1	55.6	55.3	63.2	57.5 ± 3.4
DKD	69.7	64.1	68.9	69.1	67.0	67.8 ± 2.0
EWF	65.6	59.2	63.4	63.5	64.3	63.2 ± 2.2
Ours	**71.7**	**66.4**	**71.3**	**71.4**	**69.6**	$\mathbf{70.1 \pm 2.0}$

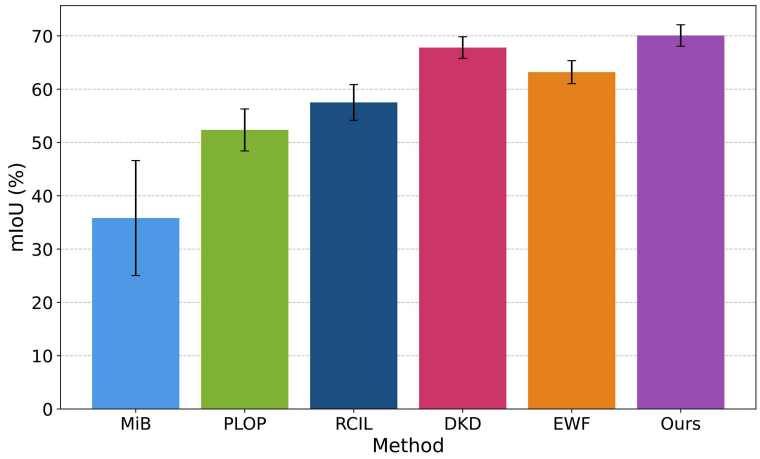


Figure 5 (Color online) mIoU distributions on VOC 15-1 under five different class orderings.

time and overall memory usage on VOC 15-1, as shown in Table 6. Our method demonstrates comparable or lower complexity than current distillation-based approaches. We thereby achieve superior performance at lower training costs.

Robustness to different class orderings. To demonstrate the robustness of our method across different class orderings, we conduct experiments on VOC 15-1 and report the mean and standard deviation of mIoU in Table 7 on five different class orderings, including one original alphabetical ordering and four random orderings, defined as follows:

A: {[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], [16], [17], [18], [19], [20]},

B: {[0, 12, 9, 20, 7, 15, 8, 14, 16, 5, 19, 4, 1, 13, 2, 11}, [17], [3], [6], [18], [10]},

C: {[0, 13, 19, 15, 17, 9, 8, 5, 20, 4, 3, 10, 11, 18, 16, 7], [12], [14], [6], [1], [2]},

D: {[0, 15, 3, 2, 12, 14, 18, 20, 16, 11, 1, 19, 8, 10, 7, 17], [6], [5], [13], [9], [4]},

E: {[0, 7, 5, 3, 9, 13, 12, 14, 19, 10, 2, 1, 4, 16, 8, 17], [15], [18], [6], [11], [20]}.

As illustrated in Figure 5, our method not only achieves higher performance but also exhibits lower variation than SOTA approaches. Consequently, these experimental results validate that our method is more robust to different class orderings.

Combination with other methods. Our proposed method can seamlessly integrate into existing CSS approaches. We apply our bidirectional mapping distillation to three competitive approaches: MiB [40], RCIL [10],

Table 8 Results of our method combined with other approaches. Absolute improvements (percentage points) are shown in parentheses. Numbers in bold denote the best results.

Method	15-1 (6 tasks)			
	0-15	16-20	all	*HM*
MiB	34.2	13.5	29.3	19.4
+ $\mathcal{L}_{bmd}$	39.3 (+5.1)	18.9 (+5.4)	34.4 (+5.1)	25.5 (+6.2)
+ $\mathcal{L}_{fmd}$	39.0 (+4.8)	20.5 (+7.0)	34.6 (+5.3)	26.9 (+7.5)
+ $\mathcal{L}_{bdmd}$	**41.8 (+7.6)**	**21.7 (+8.2)**	**37.0 (+7.7)**	**28.6 (+9.2)**
RCIL	70.6	23.7	59.4	35.5
+ $\mathcal{L}_{bmd}$	71.5 (+0.9)	25.8 (+2.1)	60.6 (+1.2)	37.9 (+2.4)
+ $\mathcal{L}_{fmd}$	70.9 (+0.3)	29.9 (+6.2)	61.1 (+1.7)	42.1 (+6.6)
+ $\mathcal{L}_{bdmd}$	**71.7 (+1.1)**	**31.3 (+7.6)**	**62.1 (+2.7)**	**43.6 (+8.1)**
EWF	78.0	25.5	65.5	38.4
+ $\mathcal{L}_{bmd}$	78.3 (+0.3)	27.7 (+2.2)	66.3 (+0.8)	40.9 (+2.5)
+ $\mathcal{L}_{fmd}$	78.1 (+0.1)	32.5 (+7.0)	67.2 (+1.7)	45.9 (+7.5)
+ $\mathcal{L}_{bdmd}$	**78.5 (+0.5)**	**34.1 (+8.6)**	**67.9 (+2.4)**	**47.5 (+9.1)**

Table 9 CSS results with a transformer-based framework. Numbers in bold denote the best results.

Method	Framework	15-1 (6 tasks)			
		0-15	16-20	all	*HM*
MiB (CVPR2020)	DeepLab-v3	34.2	13.5	29.3	19.4
PLOP (CVPR2021)	DeepLab-v3	65.1	21.1	54.6	31.9
RCIL (CVPR2022)	DeepLab-v3	70.6	23.7	59.4	35.5
Ours	DeepLab-v3	78.9	48.7	71.7	60.2
MiB (CVPR2020)	ViT-B/16	72.6	23.1	61.7	35.1
Incrementer (CVPR2023)	ViT-B/16	79.6	59.6	75.6	68.1
Ours	ViT-B/16	**81.8**	**63.2**	**77.4**	**71.3**

Table 10 Weakly supervised CSS results on the COCO-to-VOC setting in terms of mIoU (%). Numbers in bold denote the best results.

Method	COCO			VOC
	1-60	61-80	all	61-80
WILSON (CVPR2022)	39.8	41.0	40.6	55.7
WILSON + Ours	**40.9 (+1.1)**	**42.6 (+1.6)**	**41.9 (+1.3)**	**56.9 (+1.2)**
FMWISS (CVPR2023)	39.9	44.7	41.6	63.6
FMWISS + Ours	**40.9 (+1.0)**	**45.8 (+1.1)**	**42.7 (+1.1)**	**65.1 (+1.5)**

and EWF [43]. As shown in Table 8, we observe consistent performance improvements across all approaches on the VOC 15-1 setting. Specifically, integrating our method yields significant boosts of 7.6%, 1.1%, and 0.5% in the mIoU of old classes for MiB, RCIL, and EWF, respectively. Furthermore, this approach improves the respective baselines by 8.2%, 7.6%, and 8.6% in terms of mIoU on new classes. These considerable gains verify the efficacy and compatibility of our method.

Compatibility with a transformer-based framework. We conduct experiments validating the compatibility of our method with transformer-based architectures, as shown in Table 9. Following Incrementer [59], we replace DeepLab-v3 [2] with a vision transformer (ViT-16/B [60] pre-trained on ImageNet [56]) as the encoder. Compared with CNNs, the vision transformer can capture richer global contexts via self-attention, benefiting knowledge transfer across steps in our method. With this ViT framework, our method surpasses the SOTA Incrementer by 3.2% on the *HM* metric, achieving 71.3% on VOC 15-1. Therefore, the results demonstrate that our method effectively facilitates continual learning for CNN and transformer models.

Generalization ability of our method. To validate the generalizability of our method, we apply it to the weakly supervised CSS task. We conduct experiments integrating our method with two existing methods: WILSON [61] and FMWISS [62]. Following the protocol in WILSON [61], we report mIoU on the COCO [63] and VOC test sets, respectively. As shown in Table 10, our method improves the performance of WILSON and FMWISS by over 1%. The consistent gains demonstrate our method's generalization capacity, which effectively facilitates CSS under supervised and weakly supervised scenarios.

Table 11 Sensitivity analysis of the loss weight hyperparameter α on VOC 15-1 and ADE 100-10. Numbers in bold denote the best results.

α	VOC				ADE			
	0-15	16-20	all	*HM*	0-100	101-150	all	*HM*
0.1	77.9	45.2	70.1	57.2	41.8	22.3	35.3	29.1
0.05	78.4	46.9	70.9	58.7	42.0	23.0	35.7	29.7
0.01	**78.9**	**48.7**	**71.7**	**60.2**	**42.1**	**23.8**	**36.0**	**30.4**
0.005	78.1	48.2	71.0	59.6	41.9	23.6	35.8	30.2
0.001	77.5	47.3	70.3	58.7	41.5	23.2	35.4	29.8

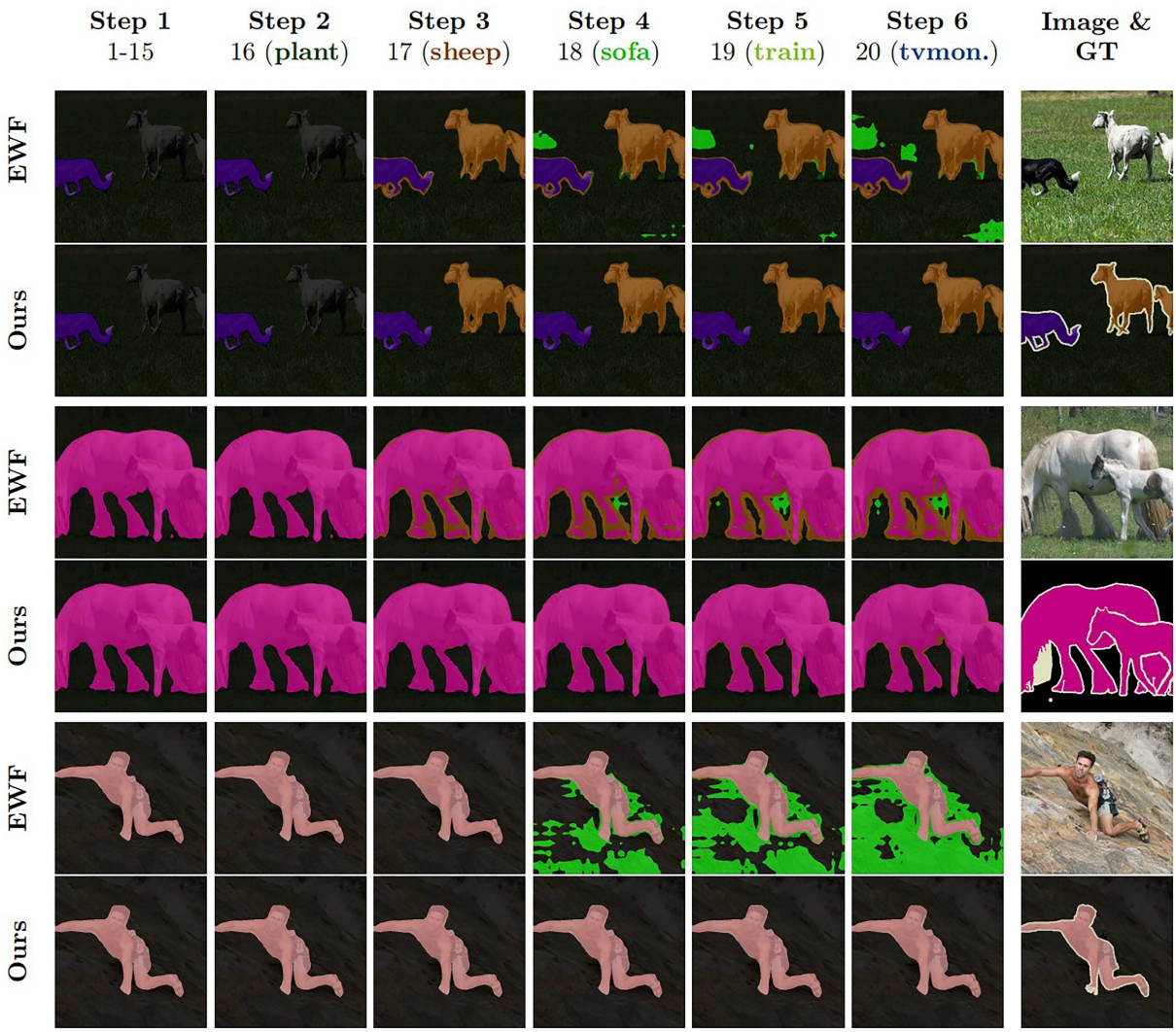


Figure 6 (Color online) Visualization of EWF and our method across six incremental steps on VOC 15-1. In the first step, 15 base classes are learned. In the subsequent steps (Steps 2–6), five new classes are introduced sequentially, denoted by class names with corresponding colors.

Sensitivity analysis of loss weight α. As shown in (8), we use the loss weight hyperparameter α to balance the segmentation training loss and our proposed bidirectional mapping distillation loss. To investigate the influence of α, we conduct experiments on the VOC 15-1 and ADE 100-10 settings, varying α from 0.001 to 0.1. As shown in Table 11, consistent performance is achieved across this range of α values, demonstrating the robustness of our method to the choice of this hyperparameter. We empirically set α to 0.01 for all experiments.

Visualization analysis. Figure 6 presents a qualitative analysis comparing our approach with the SOTA method EWF, using three test images on VOC 15-1. The visualizations demonstrate equivalent predictions from both methods in the initial step. However, for the first two images, we observe that EWF gradually forgets old classes (*dog* and *horse*) from step 3, with the prediction becoming biased toward the new class *sheep*. For the third image, EWF incorrectly segments background regions into the current foreground classes. In contrast, our

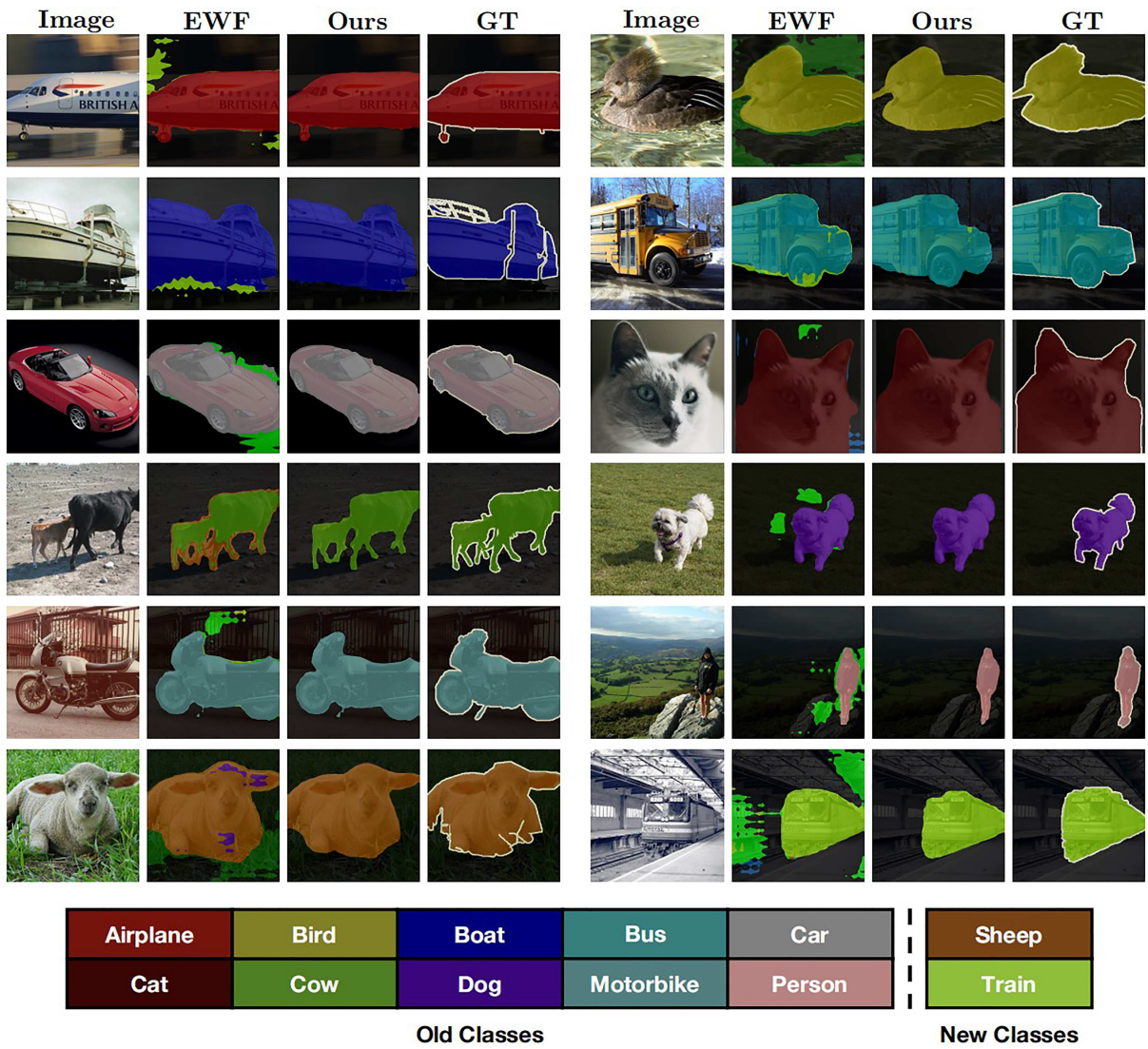


Figure 7 (Color online) Visualization of EWF and our method at the last step on VOC 15-1.

predictions exhibit significantly greater stability when learning new classes incrementally. Figure 7 further illustrates the final segmentation outputs of both methods. Our method consistently produces superior segmentation results across various classes (e.g., *airplane*, *bird*, *boat*, *bus*, *car*, *cat*, *cow*, *dog*, *motorbike*, *person*, *sheep*, and *train*).

4.4 Discussion

Here, we discuss the differences between our method and MiB [40]. MiB proposes an unbiased knowledge distillation loss to address the background shift issue by summing the probabilities of the new class and the background class during distillation. However, it fails to restrict the class-wise relationship. By contrast, our backward distillation combines the logits of the new class into all the old classes based on the semantic relationship. More importantly, MiB employs a unidirectional knowledge transfer, while our forward distillation loss can guide forward knowledge transfer, thereby facilitating learning in the new classes. Consequently, our method achieves a better balance between stability and plasticity.

5 Conclusion

In this paper, we establish a semantic relationship between old and new classes in CSS by utilizing text embeddings of image-level labels. Guided by this semantic relationship, we propose a novel bidirectional mapping distillation to

enhance knowledge transfer across different steps. Forward distillation facilitates learning new classes, while backward distillation helps address catastrophic forgetting, thereby striking a balance between the model's stability and plasticity. Extensive experiments on two representative datasets, PASCAL VOC 2012 and ADE20K, demonstrate that our method achieves the best performance in CSS.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 62372117, 62472102) and Shanghai Municipal Science and Technology Major Project (Grant No. 2025SHZDZX025G11). The computations in this research were performed using the CFFF platform of Fudan University.

References

1 Badrinarayanan V, Kendall A, Cipolla R. SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell, 2017, 39: 2481–2495
2 Chen L C, Papandreou G, Schroff F, et al. Rethinking atrous convolution for semantic image segmentation. 2017. ArXiv:1706.05587
3 Zhang Z J, Pang Y W. CGNet: cross-guidance network for semantic segmentation. Sci China Inf Sci, 2020, 63: 120104
4 Ren D Y, Wu Z Y, Li J W, et al. Point attention network for point cloud semantic segmentation. Sci China Inf Sci, 2022, 65: 192104
5 Zhang D, Zhang L Y, Tang J H. Augmented FCN: rethinking context modeling for semantic segmentation. Sci China Inf Sci, 2023, 66: 142105
6 Li Y S, Chen W, Huang X, et al. MFVNet: a deep adaptive fusion network with multiple field-of-views for remote sensing image semantic segmentation. Sci China Inf Sci, 2023, 66: 140305
7 French R M. Catastrophic forgetting in connectionist networks. Trends Cogn Sci, 1999, 3: 128–135
8 Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks. In: Proceedings of the National Academy of Sciences, 2017. 3521–3526
9 Douillard A, Chen Y, Dapogny A, et al. Plop: learning without forgetting for continual semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 4040–4050
10 Zhang C B, Xiao J W, Liu X, et al. Representation compensation networks for continual semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 7053–7064
11 Mermillod M, Bugaiska A, Bonin P. The stability-plasticity dilemma: investigating the continuum from catastrophic forgetting to age-limited learning effects. Front Psychol, 2013, 4: 54654
12 Lampert C H, Nickisch H, Harmeling S. Learning to detect unseen object classes by between-class attribute transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2009. 951–958
13 Devlin J, Chang M W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, 2019. 4171–4186
14 Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015. 3431–3440
15 Shuai B, Ding H, Liu T, et al. Toward achieving robust low-level and high-level scene parsing. IEEE Trans Image Process, 2018, 28: 1378–1390
16 Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2881–2890
17 Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell, 2017, 40: 834–848
18 Chen L C, Zhu Y, Papandreou G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision, 2018. 801–818
19 Wang X, Girshick R, Gupta A, et al. Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7794–7803
20 Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 3146–3154
21 Huang Z, Wang X, Huang L, et al. Ccnet: criss-cross attention for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019. 603–612
22 Zhang L, Xu D, Arnab A, et al. Dynamic graph message passing networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 3726–3735
23 Chen C F R, Fan Q, Panda R. Crossvit: cross-attention multi-scale vision transformer for image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 357–366
24 Wang W, Xie E, Li X, et al. Pyramid vision transformer: a versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 568–578
25 Xie E, Wang W, Yu Z, et al. Segformer: simple and efficient design for semantic segmentation with transformers. In: Proceedings of Advances in Neural Information Processing Systems, 2021, 34: 12077-12090
26 Cheng B, Misra I, Schwing A G, et al. Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 1290–1299
27 Zhou D W, Ye H J, Zhan D C. Co-transport for class-incremental learning. In: Proceedings of the 29th ACM International Conference on Multimedia, 2021. 1645–1654
28 De Lange M, Aljundi R, Masana M, et al. A continual learning survey: defying forgetting in classification tasks. IEEE Trans Pattern Anal Mach Intell, 2021, 44: 3366–3385
29 Sun G, Liang W, Dong J, et al. Create your world: lifelong text-to-image diffusion. IEEE Trans Pattern Anal Mach Intell, 2024, 46: 6454–6470
30 Liang W, Sun G, Liu C, et al. I3dod: towards incremental 3D object detection via prompting. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, 2023. 5738–5743
31 Isele D, Cosgun A. Selective experience replay for lifelong learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018. 32
32 Cha S, Yoo Y J, Moon T. Ssul: semantic segmentation with unknown label for exemplar-based class-incremental learning. In: Proceedings of Advances in Neural Information Processing Systems, 2021. 10919–10930
33 Oh Y, Baek D, Ham B. Alife: adaptive logit regularizer and feature replay for incremental semantic segmentation. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 14516–14528
34 Shokri R, Shmatikov V. Privacy-preserving deep learning. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015. 1310–1321
35 Shin H, Lee J K, Kim J, et al. Continual learning with deep generative replay. In: Proceedings of Advances in Neural Information Processing Systems, 2017. 30
36 Maracani A, Michieli U, Toldo M, et al. Recall: replay-based continual learning in semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 7026–7035
37 Mallya A, Lazebnik S. Packnet: adding multiple tasks to a single network by iterative pruning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 7765–7773

38 Yan S, Xie J, He X. Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. 3014–3023
39 Douillard A, Ramé A, Couairon G, et al. Dytox: transformers for continual learning with dynamic token expansion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 9285–9295
40 Cermelli F, Mancini M, Bulo S R, et al. Modeling the background for incremental learning in semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 9233–9242
41 Baek D, Oh Y, Lee S, et al. Decomposed knowledge distillation for class-incremental semantic segmentation. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 10380–10392
42 Yang Z, Li R, Ling E, et al. Label-guided knowledge distillation for continual semantic segmentation on 2D images and 3D point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 18601–18612
43 Xiao J W, Zhang C B, Feng J, et al. Endpoints weight fusion for class incremental semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 7204–7213
44 Cong W, Cong Y, Dong J, et al. Gradient-semantic compensation for incremental semantic segmentation. IEEE Trans Multimedia, 2023, 26: 5561–5574
45 Lin Z, Wang Z, Zhang Y. Preparing the future for continual semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023. 11910–11920
46 Yuan B, Zhao D, Shi Z. Learning at a glance: towards interpretable data-limited continual semantic segmentation via semantic-invariance modelling. IEEE Trans Pattern Anal Mach Intell, 2024, 46: 7909–7923
47 Yu L, Liu X, van de Weijer J. Self-training for class-incremental semantic segmentation. IEEE Trans Neural Netw Learn Syst, 2022, 34: 9116–9127
48 Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space. 2013. ArXiv:1301.3781
49 Pennington J, Socher R, Manning C D. Glove: global vectors for word representation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2014. 1532–1543
50 Michieli U, Zanuttigh P. Incremental learning techniques for semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, 2019
51 Chen J, Cong R, Luo Y, et al. Saving 100× storage: prototype replay for reconstructing training sample distribution in class-incremental semantic segmentation. In: Proceedings of Advances in Neural Information Processing Systems, 2024. 36
52 Everingham M, Eslami S M A, Van Gool L, et al. The pascal visual object classes challenge: a retrospective. Int J Comput Vis, 2015, 111: 98–136
53 Zhou B, Zhao H, Puig X, et al. Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 633–641
54 Hao X, Jiang X, Ni W, et al. Prompt-guided semantic-aware distillation for weakly supervised incremental semantic segmentation. IEEE Trans Circuits Syst Video Technol, 2024, 34: 10632–10645
55 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
56 Deng J, Dong W, Socher R, et al. Imagenet: a large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009. 248–255
57 Paszke A, Gross S, Massa F, et al. Pytorch: an imperative style, high-performance deep learning library. In: Proceedings of Advances in Neural Information Processing Systems, 2019
58 Radford A, Kim J W, Hallacy C, et al. Learning transferable visual models from natural language supervision. In: Proceedings of International Conference on Machine Learning, 2021. 8748–8763
59 Shang C, Li H, Meng F, et al. Incrementer: transformer for class-incremental semantic segmentation with knowledge distillation focusing on old class. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 7214–7224
60 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16×16 words: transformers for image recognition at scale. 2020. ArXiv:2010.11929
61 Cermelli F, Fontanel D, Tavera A, et al. Incremental learning in semantic segmentation from image labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 4371–4381
62 Yu C, Zhou Q, Li J, et al. Foundation model drives weakly incremental learning for semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023. 23685–23694
63 Lin T Y, Maire M, Belongie S, et al. Microsoft coco: common objects in context. In: Proceedings of the European Conference on Computer Vision, 2014. 740–755