TIKP: Text-to-Image Knowledge Preservation for Continual Semantic Segmentation

Zhidong Yu[1], **Wei Yang**[1,2*], **Xike Xie**[1,3*], **Zhenbo Shi**[1,3]

[1]School of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China
[2]Hefei National Laboratory, Hefei 230088, China
[3]Suzhou Institute for Advanced Research, University of Science and Technology of China, Suzhou 215123, China
qubit@ustc.edu.cn

Abstract

Continual Semantic Segmentation (CSS) is an emerging trend, where catastrophic forgetting has been a perplexing problem. In this paper, we propose a Text-to-Image Knowledge Preservation (TIKP) framework to address this issue. TIKP applies Text-to-Image techniques to CSS by automatically generating prompts and content adaptation. It extracts associations between the labels of seen data and constructs text-level prompts based on these associations, which are preserved and maintained at each incremental step. During training, these prompts generate correlated images to mitigate the catastrophic forgetting. Particularly, as the generated images may have different distributions from the original data, TIKP transfers the knowledge by a content adaption loss, which determines the role played by the generated images in incremental training based on the similarity. In addition, for the classifier, we use the previous model from a different perspective: misclassifying new classes into old objects instead of the background. We propose a knowledge distillation loss based on wrong labels, enabling us to attribute varying weights to individual objects during the distillation process. Extensive experiments conducted in the same setting show that TIKP outperforms state-of-the-art methods by a large margin on benchmark datasets.

Introduction

Semantic segmentation is a fundamental computer vision task that is widely used in various real-world scenarios. Recently, numerous models (Chen et al. 2017, 2018; Long, Shelhamer, and Darrell 2015; Xie et al. 2021; Xiao et al. 2018) have been designed to solve this problem with promising results. Nevertheless, these deep models face a considerable challenge of catastrophic forgetting (Michieli and Zanuttigh 2019) in the scenario of continual semantic segmentation (CSS), which means that the network forgets the categories it has already acquired while learning new ones.

Michieli et al. (Michieli and Zanuttigh 2019) first proposed the CSS task and pointed out that the task suffers from catastrophic forgetting. After that, a number of methods have been proposed to solve this task with promising results. Distillation-based methods (Cermelli et al. 2020; Douillard et al. 2021; Michieli and Zanuttigh 2021a,b; Phan et al. 2022; Shang et al. 2023) consider other properties of this task in distilling the knowledge of the old model to the new one to obtain better results. For example, MiB (Cermelli et al. 2020) takes into account the problem of background bias in CSS and models it to mitigate the forgetting of old knowledge. Some approaches (Maracani et al. 2021; Zhu et al. 2023) use replay-based methods to improve the performance. In addition, there are works (Cha et al. 2021; Zhang et al. 2022; Xiao et al. 2023) that utilize other techniques to retain the knowledge acquired.

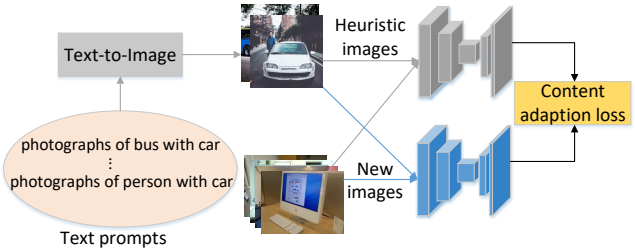

Figure 1: Overview of heuristic images-based CSS. In continual training, these heuristic images provide old knowledge to the model by content adaptation loss combined with added data.

In real life, when learning new knowledge, humans often recall old knowledge through key information, which strengthens their memory of it. Inspired by this phenomenon, we propose using Text-to-Image technology to heuristically generate images and assist in incremental training, thus avoiding catastrophic forgetting, as shown in Fig. 1. Text-to-Image technology can generate relevant images from text prompts that contain specific categories. Based on this, we extract prompts and their corresponding categories from all seen images and maintain a set of prompts between seen categories, which is inexpensive. During the incremental training process, we use these maintained prompts to generate images with similar labels to the old data and protect the old knowledge. Compared with GAN-based methods, images generated based on prompts have a higher quality and stronger heuristic ability.

The heuristic images (or generated images) can assist the model in retaining the old knowledge through two approaches: pseudo-label and knowledge distillation. Note that

*Corresponding Authors.

the distributions between these images and old data may or may not be identical. When the heuristic images are similar to the original distribution, it is appropriate to transfer the old knowledge using a pseudo-label. For generated images that are dissimilar to the seen data, it is more suitable to transfer the old knowledge by knowledge distillation. To this end, we introduce a content adaptation loss for incremental semantic segmentation. This loss calculates the content similarity between the generated image with the original data, and uses it as the magnitude of the difference in the distribution between them. Generated images with smaller differences from the seen image are biased to provide the old knowledge to the model using pseudo-label, while those with larger differences are used for knowledge distillation.

Moreover, in the process of continual learning, the model may misclassify some pixels belonging to the added classes as old objects instead of the background. This will cause the performance of these old objects to degrade significantly faster than others, as their features are quickly covered. Therefore, it is important to provide more protection to those classes that have been misclassified. To achieve this, we propose a knowledge distillation loss based on wrong labels. This loss assigns different weights to each class based on the ratio of the number of pixels of the new classes to those of the classes that are misclassified. The weights of all seen classes are added to the knowledge distillation loss to avoid forgetting these old classes quickly.

Finally, extensive experiments on benchmark datasets demonstrate the effectiveness of TIKP. The quantitative results show that our method achieves better performance than competitors by a large margin, and generates more reasonable segmentation results by better retaining the old knowledge through heuristic images.

Our main contributions can be summarized as follows:

- We propose TIKP, which extracts and maintains a set of text prompts during training, and uses these prompts to generate heuristic images to address the catastrophic forgetting issue.
- We design a content adaptation loss, which dynamically adjusts the way the generated images retain the old knowledge to avoid performance degradation due to inconsistent distribution among images.
- We put forward a knowledge distillation loss based on wrong labels to protect classes that are misclassified and prevent rapid forgetting during continual learning.
- We show through extensive experiments that TIKP performs significantly better than state-of-the-art methods in existing scenarios and datasets.

Related Work

Continual Learning

There are gradually increasing concerns about continual learning (also known as incremental or lifelong learning). Previous works are divided into three main categories: replay-based, regularization-based, and parameter isolation-based. Replay-based methods (Rebuffi et al. 2017; Hou et al. 2019; Iscen et al. 2020) select or generate examples of previous tasks in some way. Then, the model employs these examples along with the new data to learn the new classes. Regularization-based methods (Zenke, Poole, and Ganguli 2017; Douillard et al. 2020; He et al. 2020) adopt some techniques, such as the distillation, to generate an additional loss that acts as a regularization constraint to prevent forgetting. Parameter isolation-based methods (Rusu et al. 2016; Liu et al. 2020) allocate an independent set of model parameters to each task to prevent forgetting.

Continual Semantic Segmentation

Michieli et al. (Michieli and Zanuttigh 2019) first propose continual learning for semantic segmentation and put forward a general framework to address the problem of catastrophic forgetting, which retains the old knowledge through the knowledge distillation of the output and feature spaces of the model. Subsequently, several distillation-based works (Cermelli et al. 2020; Douillard et al. 2021; Michieli and Zanuttigh 2021a; Phan et al. 2022) are proposed. MiB (Cermelli et al. 2020) first points out the background shift problem in CSS, and models the background to mitigate the transfer problem. PLOP (Douillard et al. 2021) proposes Local POD that preserves long and short-distance spatial relationships at the feature level. SDR (Michieli and Zanuttigh 2021a) uses prototype matching and contrast learning to construct robust features. REMINDER (Phan et al. 2022) designs CSW-KD, which adjusts the distillation weights of each class based on the similarity between objects.

In addition, RECALL (Maracani et al. 2021) retains the seen classes using images generated by GAN or crawled from the Web. Some other approaches (Cha et al. 2021; Zhang et al. 2022) achieve promising results with additional models or structures. SSUL (Cha et al. 2021) relies on the saliency detection model to discover potential objects, which requires models trained on other datasets. RCIL (Zhang et al. 2022), on the other hand, utilizes parallel convolutions to improve performance.

Text-to-Image Technology

Text-to-image is an emerging technique that uses generative models to generate images that are inspired by textual descriptions. Some approaches rely on GANs (Brock, Donahue, and Simonyan 2019; Karras et al. 2020) and achieve satisfactory results. Moreover, advances in diffusion probabilistic models (DMs) (Sohl-Dickstein et al. 2015) lead to state-of-the-art results in terms of both density estimation (Kingma et al. 2021) and sample quality (Dhariwal and Nichol 2021). Building on these advances, LDM (Rombach et al. 2022) introduces the technique for high-quality image synthesis. Different from these works, we extract text prompts from the dataset and use them to generate heuristic images that help retain the old knowledge during incremental training.

Methodology

Overview

Before formulating the framework, we first introduce some related concepts. The purpose of CSS is to train a segmentation model in T steps to learn new classes without forgetting

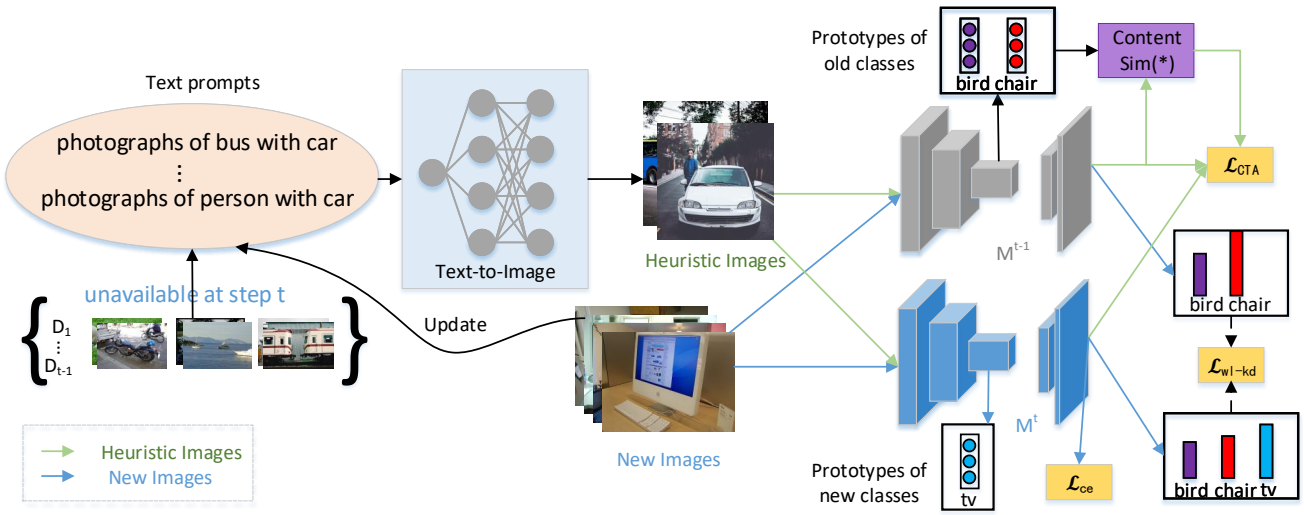

Figure 2: Overview of TIKP. The framework contains a text prompts that is used to instruct the Text-to-Image model to generate relevant images. These images are trained incrementally in conjunction with the added data. The generation of correlated images is supervised by the content adaption loss to avoid performance degradation due to the inconsistent distribution. The added images are supervised by the knowledge distillation loss based on wrong labels. Models in gray indicate that they are frozen during training.

old ones. We define that C^t is the class learned at step t, and $C^{1:t-1}$ denotes all the seen classes from step 1 to step $t-1$. For step t, we present a dataset D_t that comprises a set of pairs (X^t, L^t), where X^t is an image with a size of $H \times W$, and L^t is the ground truth segmentation map, which contains only the class C^t learned in the current step. Besides, we denote Y^t as the one-hot label of L^t.

Typically, the segmentation model at step t is denoted as M^t, which contains a feature extractor f^t and a classifier g^t. The feature extractor f^t is used to extract the features of the image, while the classifier g^t outputs the corresponding semantic segmentation prediction maps Z^t.

Fig. 2 shows the framework of TIKP, which first extracts text prompts from all seen data and stores them in memory. In incremental training, relevant images are generated to address the catastrophic forgetting by the Text-to-Image model, which is denoted as G. These images are combined with the added data for incremental training. In addition, the added data is trained by the cross-entropy loss ($\mathcal{L}_{ce}$) and the knowledge distillation loss based on wrong labels ($\mathcal{L}_{wl-kd}$). The old knowledge of heuristic images is transferred from M^{t-1} to M^t by the content adaption loss ($\mathcal{L}_{CTA}$).

Generated Images by Text Prompts

TIKP leverages the Text-to-Image model to generate diverse and relevant images, effectively preserving old knowledge during continual learning. Text prompts are used to guide the image generation, allowing the model to recall important information from previously learned classes and incorporate it into training new classes. We use the old dataset to generate prompts for controlling image generation. By counting co-occurrence frequencies, we select the top combinations of classes to generate prompts, reducing their number. These prompts serve as a fixed base inspiration for subsequent incremental training.

In each step t ($t > 1$), we generate new prompts for added classes and update the prompt file. Pseudo-labels are obtained from the model trained in the previous step and combined with ground truth labels to create comprehensive pseudo-labels for all seen classes. These pseudo-labels are used to generate prompts for new classes based on their correlations with old classes.

The format of a text prompt for m classes is "*a color photograph of class* 1, *class* 2, ..., *and class* m". The Text-to-Image model uses these prompts to generate images $X^{t'}$ at step t, containing only seen classes. The pseudo-label $L^{t-1'}$ is obtained from M^{t-1}, and the corresponding one-hot label is $Y^{t-1'}$. These generated images and pseudo-labels assist the model in retaining learned knowledge during continual learning.

Knowledge Preservation by the Content Adaption Loss

As mentioned before, there are usually two methods: pseudo-label and knowledge distillation. The former refers to obtaining pseudo-labels for these generated images by the model M^{t-1} in step $t-1$, and then using these pseudo-labels to supervise the training of M^t, which is more applicable to the data with the same distribution. The latter refers to M^t learning the prediction distribution of M^{t-1} generated in these heuristic samples, and it is more applicable to the data with different distributions. Unlike them, we design the content adaption loss to solve the problem no matter whether the distribution between the generated and original images are the same. It will construct the distillation and cross-entropy loss weights *automatically* for each heuristic image based on

the similarity between the old and heuristic data.

Specifically, we use intermediate features to evaluate the similarity between data. For each seen class, we compute a feature prototype $\mathbf{P_c}$. To obtain the prototype $\mathbf{P_c}$ of the new class c at step t, we calculate an average on the features of all images belonging to D_t as follows:

$$\mathbf{P_c} = \frac{\sum_{(X^t,L^t)\in D_t} \sum_i^{HW} f^t(X^t)\mathbb{1}_{L^t=c}}{\sum_{(X^t,L^t)\in D_t} \sum_i^{HW} \mathbb{1}_{L^t=c}} \tag{1}$$

The cumulative prototypes of all classes from task 1 to task t are computed at the end of task t. In incremental step t, for each generated image $X^{t'}$, we first compute its pseudo-label $L^{t'}$ using M^{t-1}. Then, for each class c in the pseudo-label, we calculate the class features $\mathbf{F_c}$:

$$\mathbf{F_c} = \frac{\sum_i^{HW} f^{t-1}(X^{t'})\mathbb{1}_{L^{t-1'}=c}}{\sum_i^{HW} \mathbb{1}_{L^{t-1'}=c}} \tag{2}$$

Then, we calculate the cosine similarity between the features $\mathbf{F_c}$ and the feature prototypes $\mathbf{P_c}$ retained by the original data. The similarity of each class is computed by:

$$s_c = \frac{\mathbf{P_c} \cdot \mathbf{F_c}}{\|\mathbf{P_c}\|\|\mathbf{F_c}\|} \tag{3}$$

After that, the similarity of the whole generated image to the original data is defined:

$$s_{X^{t'}} = \frac{\sum_{c\in L^{t-1'}} s_c}{\sum_{c\in L^{t-1'}} 1} \tag{4}$$

This similarity is the content similarity of each image to the original data. The greater the similarity, the more similar the image is to the distribution of seen data, and the greater weight is applied to the pseudo-label to retain the old knowledge. On the other hand, a smaller similarity indicates that the image is less similar to the distribution of seen data, then a greater weight is applied to the distillation for retaining the old knowledge.

Thus, the content adaption loss is formulated as:

$$\begin{aligned}\mathcal{L}_{CTA} = \sum_{c\in C^{1:t-1}} &((s_{X^{t'}} - 1)M^{t-1}(X^{t'})_c \log(Z_c^{t'}) \\ &- s_{X^{t'}} Y_c^{t-1'} \log(Z_c^{t'}))\end{aligned} \tag{5}$$

where $M^{t-1}(X^{t'})_c$ is the $c-$th channel of the predictions of M^{t-1}, $Y^{t-1'}$ is the one-hot label of $X^{t'}$ output by M^{t-1}, and $Z_c^{t'} = M^t(X^{t'})$.

Pseudo-Labels for Added Data

In CSS, the background shift problem, where the pixels of the previous classes are labeled as the background in the current step, causes the old classes to be overwritten quickly. To solve this problem, the previous model trained at step $t-1$ is employed to generate the pseudo-label $\tilde{Y}^{t-1}$, which contains the labels of all seen classes. It is combined with the label Y^t of the current step to generate a new pseudo-label $\tilde{Y}^t$, which includes the label of the current class and the pseudo-label of all seen classes. The one-hot pseudo-label of the pixel i for class c at step t is formalized as:

$$\tilde{Y}_{i,c}^t = \begin{cases} Y_{i,c}^t, & \text{if } Y_{i,c_b}^t \neq 1 \\ \tilde{Y}_{i,c}^{t-1}, & \text{if } Y_{i,c_b}^t = 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where c_b is the background, and $Y_{i,c}^t$ is the one-hot ground truth of the pixel i for class c at step t.

And, the pseudo-label $\tilde{Y}^{t-1}$ is used as a label for the added data to compute $\mathcal{L}_{ce}$.

Knowledge Distillation Loss Based on Wrong Labels

Knowledge distillation at the prediction level contributes to the retention of the prior classes. The knowledge distillation loss is formulated as:

$$\mathcal{L}_{kd} = -\frac{1}{HW} \sum_i^{HW} \sum_{c\in C^{1:t-1}} Z_{i,c}^{t-1} log(Z_{i,c}^t) \tag{7}$$

However, we find that the model from step $t-1$ recognizes pixels belonging to the new classes added at the current step as previous objects (non-background), which leads to performance degradation of the new model on these classes. Therefore, we propose the knowledge distillation based on wrong labels, which generates higher weights for those pixels that are misclassified to induce the model to better protect these classes at the current step.

First, the weight of the old class $c_o \in C^{1:t-1}$ is:

$$w_{c_o} = \frac{\sum_i^{HW} \sum_{c\in C^t} \mathbb{1}_{\tilde{Y}_{i,c_o}^{t-1}=1 \& Y_{i,c}^t=1 \& c_o \neq c_b}}{\sum_i^{HW} \sum_{c\in C^t} \mathbb{1}_{Y_{i,c}^t=1}} \tag{8}$$

where $\mathbb{1}_{\tilde{Y}_{i,c_o}^{t-1}=1 \& Y_{i,c}^t=1 \& c_o \neq c_b}$ denotes negative samples belonging to class c, i.e., the label is class c, but the predicted value by the old model is c_o (non-background). We define $w_{c_o} = 0$ if class c_o is the background.

Subsequently, The normalized weight $\hat{w}_{c_o}$ of class c_o is formulated as:

$$\hat{w}_{c_o} = \frac{e^{w_{c_o}}}{\sum_{c=1}^{C} e^{w_{c_o}}} \tag{9}$$

Thus, the knowledge distillation based on wrong labels $\mathcal{L}_{wl-kd}$ is formulated as:

$$\mathcal{L}_{wl-kd} = -\frac{1}{HW} \sum_i^{HW} \sum_{c_o\in C^{1:t-1}} \hat{w}_{c_o} Z_{i,c_o}^{t-1} log(Z_{i,c_o}^t) \tag{10}$$

Notice the difference between our $\mathcal{L}_{wl-kd}$ and CSW-KD (Phan et al. 2022), which uses prototypes between classes to re-weight them. In contrast, we use wrong labels of the old model to re-weight each class and do not use the prototypes. Therefore, they are two orthogonal techniques.

Finally, the combined loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{CTA} + \lambda_2 \mathcal{L}_{wl-kd} \tag{11}$$

where λ_1 and λ_2 are weighting factors.

Method	19-1 (2 tasks)				15-5 (2 tasks)				15-1s (6 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT	67.75	10.88	65.05	71.23	67.08	39.23	60.45	70.37	8.75	7.99	8.56	40.16
MiB	70.57	22.82	68.30	72.95	75.30	48.68	68.96	75.07	39.47	14.50	33.53	54.44
SDR	68.52	23.29	66.37	71.48	75.21	46.72	68.64	74.32	43.08	19.31	37.42	54.52
PLOP	75.50	30.22	73.35	75.43	75.44	49.65	69.30	74.82	63.41	26.76	54.68	66.96
RECALL	67.90	**53.50**	68.40	-	66.60	50.90	64.00	-	65.70	**47.80**	62.70	-
REMINDER	76.48	32.34	74.38	76.22	76.11	50.74	70.07	75.36	68.30	27.23	58.52	68.27
RICL	76.48	35.36	74.52	76.35	78.80	52.00	72.40	76.65	70.60	27.40	59.40	69.12
TIKP (Ours)	**77.40**	40.41	**75.64**	**76.82**	**78.81**	**55.50**	**73.26**	**78.46**	**73.77**	42.31	**66.28**	**71.94**

Table 1: mIoU for different continual learning settings on the dataset Pascal VOC 2012. Herein, best results are marked in boldface, and second best results are underlined.

Experiments

Experimental Setup

Datasets. We validate our method on three benchmark datasets: Pascal VOC 2012 (Everingham et al. 2010), Cityscapes (Cordts et al. 2016) and ADE20k (Zhou et al. 2017). The Pascal VOC 2012 dataset contains 20 object classes and one background. Its training and validation sets include 10,582 and 1,449 images, respectively. The ADE20k dataset contains 150 objects with 20,210 training images and 2,000 test images. The Cityscapes dataset contains 2,975 training images, 500 validation images and 1,525 test images. There are 19 classes from 21 cities.

Experimental Setting. Initially, MiB (Cermelli et al. 2020) sets two different experimental settings, namely disjoint and overlapped. Previous works (Douillard et al. 2021; Phan et al. 2022) mainly report their results in the overlapped setting, as the latter is more realistic and challenging. Therefore, we likewise evaluate the performance of the model in the overlapped setting. For the Pascal VOC 2012 dataset, we perform experiments in six settings, including adding 1 class after training 19 classes (19-1), adding 5 classes after training 15 classes (15-5), adding 5 classes sequentially after training 15 classes (15-1s), and more challenging settings of 10-10, 10-5s, and 10-1s. For the ADE20k dataset, we perform experiments in four settings, which are adding 50 classes after training 100 classes (100-50), adding 50 classes each time after training 50 classes (50-50s), and adding 10 classes each time sequentially after training 100 classes (100-10s). For the Cityscapes dataset, we follow the approach of (Douillard et al. 2021) and treat the training data for each city as a domain. We evaluate our method in three settings: adding 5 domains each time after training 11 domains (11-5), adding 5 domains each time sequentially after training 11 domains (11-1s), and adding one domain at a time (1-1s).

Metrics. For semantic segmentation, the mean Intersection over Union (mIoU) metric is frequently used to measure the performance of the model. In CSS, we report four different mIoUs. First, the mIoU of all initial classes is used to indicate the ability of the model to retain the old knowledge. Second, the mIoU of all incremental classes is used to indicate the ability of the model to learn the new knowledge. Then, the mIoU of all classes (all) shows the combination performance of the model. Finally, the average value of mIoU (avg) evaluates the performance of the model throughout the continual learning process.

Implementation Details. For all experiments, as in previous work, we use Deeplabv3 (Chen et al. 2017) as the segmentation network with ResNet-101 (He et al. 2016) as the backbone, which is pre-trained on ImageNet (Deng et al. 2009). The feature distillation is used as PLOP (Douillard et al. 2021). For the Pascal VOC 2012 and ADE20k datasets, the model is trained with a crop size of 512×512 and a batch size of 12. The model is trained for 30 epochs on Pascal VOC 2012 and 60 epochs on ADE20k, respectively. For Cityscapes, the model is trained for 50 epochs with a crop size of 800×800. Empirically, λ_1 is set to 1 and λ_2 is set to 10 in experiments. We use the stochastic gradient descent (SGD) optimizer, where the base learning rate is 0.001 with a weight decay of 0.0001. We use the Text-to-Image model to generate 100 images for each class for the Pascal VOC 2012, 50 images for each class for ADE20k, and 50 images for each class for Cityscapes via text prompts.

Quantitative Evaluation

We compare the experimental results of TIKP with state-of-the-art methods: ILT (Michieli and Zanuttigh 2019), MiB (Cermelli et al. 2020), SDR (Michieli and Zanuttigh 2021a), PLOP (Douillard et al. 2021), RECALL (Maracani et al. 2021), REMINDER (Phan et al. 2022) and RICL (Zhang et al. 2022).

For the Pascal VOC 2012 dataset, Tab. 1 shows the results for the 19-1 (2 tasks), 15-5 (2 tasks) and 15-1s (6 tasks) settings. In the 19-1 setting, REMINDER and RCIL obtain promising results for all classes (74.38% and 74.52%, respectively), and our method obtains a significant improvement on the new classes (+5.05%) compared with the latter. For the 15-5 setup, adding 5 classes in one incremental step causes the model to severely forget the old classes. Compared with the state-of-the-art method, our method improves the mIoU on both new and all classes (+3.50% and +0.86%, respectively). For the longer continual learning process (15-1s), there is a significant performance degradation for each method, especially for the newly added class. In contrast, TIKP achieves a larger improvement (+14.91%) in the new classes and retains the old knowledge well (73.77%), thus improving the overall performance (all). In addition, Tab. 2

Method	10-10 (2 tasks)				10-5s (3 tasks)				10-1s (11 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT	70.82	63.52	67.34	73.94	55.59	47.67	51.82	66.45	16.98	7.27	3.77	5.60
MiB	70.51	63.73	67.28	73.91	56.99	51.47	54.36	68.28	20.02	20.11	20.06	39.14
SDR	70.60	63.99	67.45	74.00	56.96	51.41	54.32	68.75	32.42	17.20	25.17	42.86
PLOP	73.82	63.55	68.93	74.81	58.58	53.66	56.24	69.89	44.95	15.43	30.89	44.77
RECALL	65.00	58.40	63.10	-	60.80	52.90	58.40	-	59.50	**46.70**	54.80	-
RICL	73.98	65.34	69.87	75.22	61.11	55.74	58.55	71.36	55.44	15.03	36.20	47.37
TIKP (Ours)	**75.12**	**65.61**	**70.59**	**75.60**	**69.32**	**57.91**	**63.89**	**72.66**	**69.71**	43.48	**57.22**	**67.28**

Table 2: mIoU for different continual learning settings on the dataset Pascal VOC 2012.

Method	100-50 (2 tasks)				50-50s (3 tasks)				100-10s (6 tasks)			
	old	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*	*old*	*new*	*all*	*avg*
ILT	18.29	14.40	17.00	29.42	3.53	12.85	9.70	30.12	0.11	3.06	1.09	12.56
MiB	40.52	17.17	32.79	37.31	45.57	21.01	29.31	38.98	38.21	11.12	29.24	35.12
SDR	40.52	17.17	32.79	37.31	45.66	18.76	27.85	34.25	37.26	12.13	28.94	34.48
PLOP	41.76	14.52	32.74	37.73	47.33	20.27	29.41	38.75	38.59	14.21	30.52	34.48
REMINDER	41.55	19.16	34.14	38.43	47.11	20.35	29.39	39.26	38.96	**21.28**	33.11	36.97
RCIL	**42.30**	18.80	34.50	38.63	48.30	24.40	32.50	40.26	39.30	17.50	32.10	37.47
TIKP (Ours)	42.17	**20.21**	**34.90**	**38.90**	**48.75**	**25.86**	**33.56**	**40.84**	**40.96**	19.56	**33.79**	**38.61**

Table 3: mIoU for different continual learning settings on the dataset ADE20k.

Method	100-5s (11 tasks)			
	old	*new*	*all*	*avg*
ILT	0.08	1.31	0.49	7.83
MiB	36.01	5.66	25.96	32.69
SDR	33.02	10.63	25.61	33.07
PLOP	35.72	12.18	27.93	35.10
REMINDER	36.06	16.38	29.54	36.49
RCIL	38.50	11.50	29.60	36.61
TIKP (Ours)	**37.48**	**17.56**	**30.88**	**37.11**

Table 4: mIoU for the setting 100-5s on the dataset ADE20k.

Method	11-5	11-1s	1-1s
ILT	59.11	57.48	30.11
MiB	61.58	60.06	42.29
PLOP	63.55	62.17	45.22
RCIL	64.31	63.03	48.90
TIKP (Ours)	**65.51**	**65.06**	**50.88**

Table 5: mIoU for different settings on the dataset Cityscapes.

shows the results for the 10-10 (2 tasks), 10-5s (3 tasks), and 10-1s (11 tasks) settings. For more challenging settings, TIPK achieves the most advanced results. The mIoU of all classes is improved by +0.72%, +5.34% and +21.02% at 10-10, 10-5s and 10-1s settings, respectively.

For the ADE20k dataset, Tab. 3 shows the results for the 100-50 (2 tasks), 50-50s (3 tasks), and 100-10s (6 tasks) settings. For the 100-50 setting, our method improves the mIoU of all classes (+0.40%), compared with the advanced method RCIL. For the 50-50s setting, our method obtains an mIoU of 33.56% for all classes, which is 1.06% higher than the second-best result. Moreover, For the more difficult setting of 100-10s, our method preserves the old knowledge well in the CSS, improving +1.66% on the old classes compared with RCIL and again obtaining a state-of-the-art result of 33.79% for all classes. Tab. 4 compares the performance of the model for 11 tasks in the longer setup of 100-5s on ADE20k. In such settings, the model is highly susceptible to forgetting the old knowledge because of the high number of learning steps. As shown in the table, our method performs better than current top-performing methods on all classes and gets a mIoU of 30.88%.

The experimental results under different incremental settings on Cityscapes are presented in Tab. 5. Unlike other datasets, we treat each city as an increment rather than a class. As a result, existing methods perform better on this dataset. However, we observe an inevitable performance degradation as the number of training steps increases, which is due to the differences in the data distribution between different cities. Our proposed method, TIKP, outperforms the existing methods on this dataset, achieving the best results. These results demonstrate the effectiveness of our framework in preserving and utilizing the old knowledge to mitigate catastrophic forgetting even in a more complex and challenging setting such as incremental city-wise learning.

Ablation Study

We evaluate the impact of the proposed components and the experimental results are shown in Tab. 6. We use DeepLabv3 trained with the cross-entropy loss ($\mathcal{L}_{ce}$) and knowledge distillation loss ($\mathcal{L}_{kd}$) as the baseline (BL). First, we replace $\mathcal{L}_{kd}$ with $\mathcal{L}_{wl-kd}$, and the performances of both old and new classes are improved by +4.44% and +3.34%, respectively. After that, we add the Text-to-Image strategy (TI) to

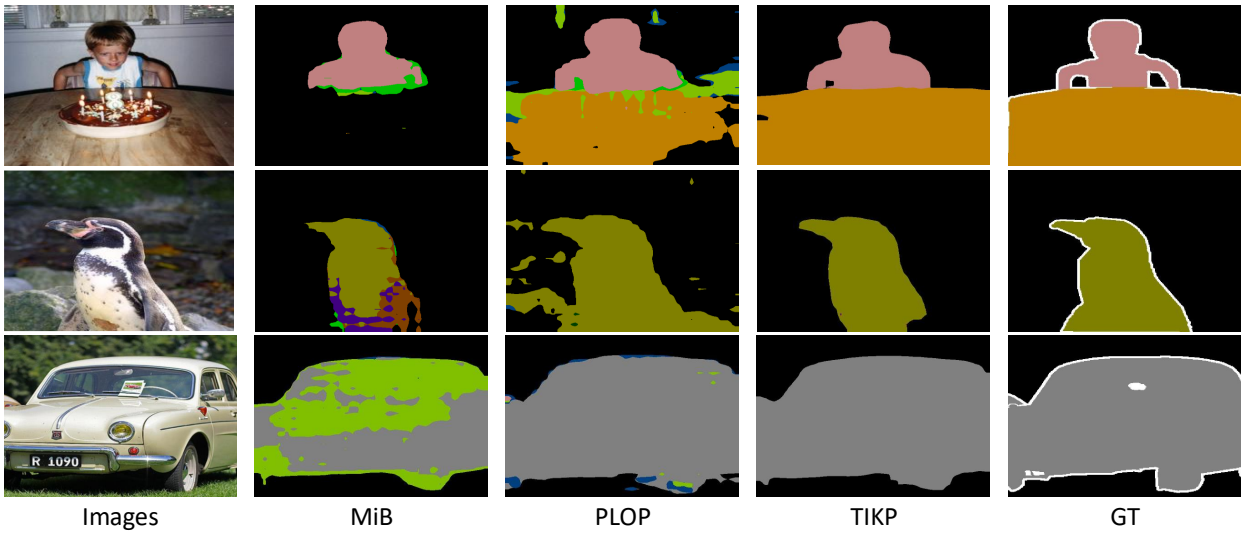

Figure 3: Visualization results of MiB, PLOP and TIKP for some test images on the Pascal VOC 2012 in 15-1s setting. TIKP has less confusion between the background and foreground classes, compared with PLOP and MiB. GT is the hand-annotated labels.

BL	$\mathcal{L}_{wl-kd}$	TI	$\mathcal{L}_{CTA}$	*old*	*new*	*all*
✓				63.41	26.76	54.68
	✓			67.85	30.10	58.86
	✓	✓		70.55	41.81	63.71
	✓	✓	✓	**71.67**	**50.29**	**66.58**

Table 6: Ablation study on the 15-1s setting of the Pascal VOC 2012 dataset. TI means adding the Generated Image in incremental training. BL means the baseline.

Figure 4: Comparisons of the heuristic images with the original images.

generate heuristic images for the old classes, and the old knowledge on the heuristic images is transferred by both the pseudo label and knowledge distillation strategies. With the help of this strategy, the performance is greatly improved (63.71%). Finally, we add the content adaption loss ($\mathcal{L}_{CTA}$) to the framework. The mIoU on the old classes is further improved (+1.12%) and the best mIoU (66.58%) for all classes is obtained. These experiments demonstrate the effectiveness of the proposed components.

Qualitative Evaluation

Fig. 3 illustrates the predictions of MiB, PLOP, and TIKP on the 15-1s setting of Pascal VOC 2012. Both MiB and PLOP have serious errors between foreground classes and the background, as shown in rows 1 and 2. Row 3 of the figure reveals the problem of confusion between foreground classes due to the forgetting during the incremental learning. In contrast, TIKP greatly alleviates this problem by transferring the old knowledge of the heuristic images.

In Fig. 4, we present a partial comparison of the heuristic with the original images. Although the generated heuristic image contains the same classes as the prompt, there are inconsistencies in terms of image style when compared with the original image. Therefore, we introduce the content adaption loss, which is designed to tackle this problem.

Conclusions

In this paper, we addressed the catastrophic forgetting in CSS by introducing the Text-to-Image Knowledge Preservation (TIKP) framework. TIKP leverages text prompts to retain old knowledge. These prompts are cost-effective as they consist of text and can be easily maintained across steps. The images generated from the prompts provide valuable information for preserving the old knowledge. Moreover, to mitigate performance degradation caused by different data distributions, we put forward a content adaption loss to measure similarity with the original data. Additionally, we proposed a knowledge distillation loss based on wrong labels to balance learning between old and new classes. Extensive experiments on benchmark datasets demonstrate that TIKP achieves state-of-the-art performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62172385 and 62072428), and the Innovation Program for Quantum Science and Technology (No. 2021ZD0302900).

References

Brock, A.; Donahue, J.; and Simonyan, K. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations*.

Cermelli, F.; Mancini, M.; Bulo, S. R.; Ricci, E.; and Caputo, B. 2020. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9233–9242.

Cha, S.; Yoo, Y.; Moon, T.; et al. 2021. SSUL: Semantic Segmentation with Unknown Label for Exemplar-based Class-Incremental Learning. *Advances in Neural Information Processing Systems*, 34: 10919–10930.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, 801–818.

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34: 8780–8794.

Douillard, A.; Chen, Y.; Dapogny, A.; and Cord, M. 2021. Plop: Learning without forgetting for continual semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4040–4050.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, 86–102. Springer.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2): 303–338.

He, J.; Mao, R.; Shao, Z.; and Zhu, F. 2020. Incremental learning in online scenario. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13926–13935.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 831–839.

Iscen, A.; Zhang, J.; Lazebnik, S.; and Schmid, C. 2020. Memory-efficient incremental learning through feature adaptation. In *European conference on computer vision*, 699–715. Springer.

Karras, T.; Laine, S.; Aittala, M.; Hellsten, J.; Lehtinen, J.; and Aila, T. 2020. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 8110–8119.

Kingma, D.; Salimans, T.; Poole, B.; and Ho, J. 2021. Variational diffusion models. *Advances in neural information processing systems*, 34: 21696–21707.

Liu, Y.; Parisot, S.; Slabaugh, G.; Jia, X.; Leonardis, A.; and Tuytelaars, T. 2020. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *European Conference on Computer Vision*, 699–716. Springer.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3431–3440.

Maracani, A.; Michieli, U.; Toldo, M.; and Zanuttigh, P. 2021. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 7026–7035.

Michieli, U.; and Zanuttigh, P. 2019. Incremental learning techniques for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Michieli, U.; and Zanuttigh, P. 2021a. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1114–1124.

Michieli, U.; and Zanuttigh, P. 2021b. Knowledge distillation for incremental learning in semantic segmentation. *Computer Vision and Image Understanding*, 205: 103167.

Phan, M. H.; Phung, S. L.; Tran-Thanh, L.; Bouzerdoum, A.; et al. 2022. Class Similarity Weighted Knowledge Distillation for Continual Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16866–16875.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2001–2010.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695.

Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; and Hadsell, R. 2016. Progressive neural networks. *NIPS 2016 Deep Learning Symposium*.

Shang, C.; Li, H.; Meng, F.; Wu, Q.; Qiu, H.; and Wang, L. 2023. 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*, 7214–7224.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, 2256–2265. PMLR.

Xiao, J.-W.; Zhang, C.-B.; Feng, J.; Liu, X.; van de Weijer, J.; and Cheng, M.-M. 2023. Endpoints Weight Fusion for Class Incremental Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7204–7213.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, 418–434.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34: 12077–12090.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 3987–3995. PMLR.

Zhang, C.-B.; Xiao, J.-W.; Liu, X.; Chen, Y.-C.; and Cheng, M.-M. 2022. Representation Compensation Networks for Continual Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7053–7064.

Zhou, B.; Zhao, H.; Puig, X.; Fidler, S.; Barriuso, A.; and Torralba, A. 2017. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 633–641.

Zhu, L.; Chen, T.; Yin, J.; See, S.; and Liu, J. 2023. Continual semantic segmentation with automatic memory sample selection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3082–3092.