Rethinking Query-based Transformer for Continual Image Segmentation

Supplementary Material

In this supplementary material, we provide additional information regarding:

- Overall Workflow of our SimCIS with Pseudocode (In Sec. 1).
- More Dataset and Implementation Details (In Sec. 2).
- Comprehensive Experiments of Random Class Rrder (In Sec. 3).
- More Ablation Studies on the Stop-Gradient Strategy. (In Sec. 4).
- More Visualization Results of the Continual Semantic Segmentation task (In Sec. 6).
- More Visualization Results of Objectness Information (In Sec. 7).
- Discussion, Limitation and Future Work (In Sec. 9).

1. Pseudocode for our SimCIS

In this section, we present the overall workflow of our method in the pseudo-code Algo. 1. At the beginning, we define some modules, functions, and variables. For the current stage t and the previous stage $t-1$, we define the backbone modules f^t_{backbone} and $f^{t-1}_{\text{backbone}}$, the pixel decoder modules f^t_{pixel} and f^{t-1}_{pixel}, the prototypes $\mathcal{P}^t$ and $\mathcal{P}^{t-1}$ respectively. For clarity and readability of the pseudocode, some formulas introduced in the main text are encapsulated as functions. These include the select feature points function Φ (Eq. 4), the consistent selection loss function l_{csl} (Eq. 6), the calculate sample weights function g (Eq. 9), the virtual query bank $\mathcal{B}_{vq}$ update function $\mathcal{U}$ (Eq. 8), and the decoder layer with skip attention Θ(Eq. 11). We also define the input image for the current stage as x^t, the Virtual Query Bank $\mathcal{B}_{vq}$, and the total training iteration M. Specifically, our lazy **Query Pre-alignment** strategy is described in the line-4 and line-8-9, our **Consistent Selection Loss** strategy is described in the line-5-7, and our **Virtual Query** strategy is described in line-11-12, line-10-16. All model and code will be made publicly available.

2. More Dataset and Implementation Details

Dataset Information. Following previous works [3, 8, 12], we use ADE20k [18] to train and evaluate our model for both continual panoptic segmentation and continual semantic segmentation tasks. The ADE20K dataset contains $20,210$ training images and $2,000$ validation images, with each image averaging 19.5 instances and 10.5 classes. Compared with other datasets, such as VOC [7], which contains an average of 2.3 instances and 1.4 classes per image. ADE20K is a particularly challenging dataset that highlights our robustness during continual training stages.

Algorithm 1 Pseudocode for SimCIS

Input: Backbone f_{backbone}, pixel decoder f_{pixel} and prototype $\mathcal{P}$ at stage t and $t-1$;
Select feature points function Φ (Eq. 4);
Consistent selection loss function l_{csl}(Eq. 6);
Calculate sample weights function g (Eq. 9);
$\mathcal{B}_{vq}$ update function $\mathcal{U}$(Eq. 8);
Decoder layer with skip attention Θ (Eq. 11);
Image of current stage x^t;
Virtual Query Bank $\mathcal{B}_{vq}$;
Training iteration M.
Output: $\mathcal{M}^t$: model of current stage.

1: $\sigma \leftarrow$ Collect pseudo-distribution statistics
2: $\omega \leftarrow g(\sigma)$
3: **for** $i \leftarrow 1, \ldots, M$ **do**
4: $\quad F^t \leftarrow f_{\text{pixel}}(f_{\text{encoder}}(x^t))$
5: $\quad F^{t-1} \leftarrow f^{t-1}_{\text{pixel}}(f^{t-1}_{\text{encoder}}(x^t))$
6: $\quad \mathcal{I}^{t-1} \leftarrow \Phi(F^{t-1}, \mathcal{P}^{t-1})$
7: $\quad \mathcal{L}_{csl} \leftarrow l_{\text{cls}}(F^t, F^{t-1}, \mathcal{I}^{t-1}, \mathcal{P}^{t-1})$ ▷ Sec. 4.2 end.
8: $\quad \mathcal{I}^t \leftarrow \Phi(F^t, P^t)$
9: $\quad Q_N \leftarrow$ Object query on F^t by $\mathcal{I}^t$. ▷ Sec. 4.1 end.
10: $\quad Q_j \leftarrow$ Sample j virtual query from $\mathcal{B}_{vq}$ using ω.
11: $\quad Q_{N+j} \leftarrow \{Q_N, Q_j\}$
12: $\quad$ **for** $l \leftarrow 1, \ldots, L$ **do**
13: $\quad\quad Q_{N+j} \leftarrow \Theta(Q_{N+j})$
14: $\quad$ **end for**
15: $\quad Z_N \leftarrow$ Get Q_N's prediction results.
16: $\quad \mathcal{B}_{vq} \leftarrow \mathcal{U}(Z_N, Q_N, y)$ ▷ Sec. 4.3 end.
17: $\quad$ Calculate L_{class} using Q_{N+j}.
18: $\quad$ Calculate L_{mask} using Q_N.
19: $\quad L_{\text{total}} \leftarrow L_{\text{class}} + L_{\text{mask}} + L_{\text{csl}}$
20: $\quad$ Update parameters via backpropagation.
21: **end for**

Implementation Details. To ensure a fair comparison, we strictly follow previous works [2, 3, 12, 15]. In the initial training step, the learning rate is set up to 1e-4, and during the incremental learning phase, it is reduced to $5e$-5. The total training iteration is set to $160,000$ in the first step and $1,000$ iterations for each class in incremental steps. We utilize a multi-step strategy to dynamically adjust our learning rate for optimizing our model, with a decay factor set to 0.1. Following [2], there are two different experimental protocols: disjoint and overlap. In the disjoint setting, each task has its own exclusive image data, while the overlap setting allows different images to appear across tasks. We choose the more challenging overlap setting as our experimental

Random ID	Our SimCIS			ECLIPSE		
	1-100	*101-150*	*all*	*1-100*	*101-150*	*all*
1	41.2	28.9	37.1	33.4	20.4	29.1
2	42.2	30.2	38.2	32.1	23.0	29.1
3	41.1	29.8	37.3	32.2	23.3	29.3
4	42.2	29.6	38.0	30.4	18.0	26.3
5	41.2	30.5	37.6	32.2	22.8	29.1
6	41.7	27.5	37.0	28.5	24.3	27.1
7	41.9	28.8	37.6	34.3	18.8	29.2
8	40.0	29.9	36.6	30.4	22.7	27.9
9	42.0	28.7	37.6	32.7	22.2	29.2
10†	39.1	33.8	37.4	11.3	0.0	7.6
Origin	42.2	30.1	38.1	41.4	18.8	33.9

Table 1. **Continual Panoptic Segmentation with 10 random order** on the ADE20K 100-5 continual panoptic segmentation scenario. † means descending order. Origin means original ascending order.

protocol. Except for setting consistent select loss weight to 2.0, we follow Mask2former [6] to set other loss weights.

3. Continual Learning with Random Order

Experiment Details. As shown in Tab. 1, we conduct extensive experiments on our model and ECLIPSE [12] under the ten random orders (detailed orders shown in Tab. 4), where nine of them were completely randomly generated using the `random` module in `Numpy` without any manual selection. As ADE20k's classes are ranked by their total pixel ratios in the entire dataset, we deliberately set the last order to descending to evaluate the model's dependency on base categories. Specifically, the descending order forces the model first to learn rare categories, enabling us to assess its continual learning ability under such challenging conditions.

Comparison with ECLIPSE. The results are shown in Tab. 1. Our model achieves SOTAs across all 10 random orders. Overall, our model achieves an increase of 41.9% across all classes compared to ECLIPSE. Specifically, the average performance of old classes improves by $+11.5\%$ PQ, and new classes see an average improvement of $+10.2\%$ PQ. In the final experiment, where we set the categories in descending order, the performance of ECLIPSE is relatively dropped by 73.9%. This demonstrates that ECLIPSE's approach, which freezes other parameters and employs the VPT [11] strategy for model updates, strongly depends on the base class during continual learning. In contrast, our model remains stable even under this highly challenging setup.

4. More Ablation Study for Stop Gradient

As we mention in the main text, we apply stop gradient on selected object query Q_N after the QPA strategy, to ensure that the information in feature map F is not disrupted during training, keeping the objectness information stable across different stages. As shown in the Tab. 2. After using the stop gradient strategy, we achieve an increase of +2.1% PQ across all classes. All the experiments in the main text use this strategy unless otherwise specified.

Psd	QPA	CSL	VQ	SG	**Panoptic 100-5** (11 tasks)		
					1-100	*101-150*	*all*
✓	✓	✓	✓		39.5	20.7	33.3
✓	✓	✓	✓	✓	42.1	21.9	35.4

Table 2. **Ablation Study on Stop Gradient.** Psd: pseudo label, QPA: lazy query pre-alignment, CSL: consistent selection loss, and SG: stop gradient.

5. Performance on COCO Ponaptic

To demonstrate the robustness of our approach across diverse datasets, we present its performance on the COCO panoptic segmentation dataset. As illustrated in Tabel 3, our method demonstrates strong adaptability across diverse datasets. Baseline means Mask2Former [6] with only pseudo label strategy [3].

Method	**Panoptic 83-5** (11 tasks)		
	1-83	84-133	all
Baseline	34.3	20.9	29.3
Our SimCIS	39.5	23.7	33.6

Table 3. **Continual Panoptic Segmentation.** Results on COCO [14] panoptic segmentation dataset where the total number of classes is 133 in PQ under the overlap setting.

6. More Visualization Results for CSS

As shown in Fig. 1, we additionally compare our SimCIS with BalConpas [5] in the 100-5 continual semantic segmentation task. In the first, second, and fourth row from Fig. 1, BalConpas encounters misclassification of the TV and lamps. In the fourth image, Balconpas fails to predict the building's accurate mask. While benefiting from the proper utilization of semantic priors in pixel feature and VQ strategy's ability to preserve class information, our SimCIS performs well in these cases.

7. Built-in Objectness Maintenance

Detailed clustering implementation. In the multi-scale feature generated by the pixel decoder, we choose the feature with the highest resolution for clustering. To evaluate the quality of objectness information contained in the features, we applied the K-means [9] algorithm for clustering. Regarding the hyperparameter settings, for the images shown in Fig. 2, we set the number of clustering centers from top to bottom as $[15, 10, 15, 15, 15, 15, 15]$.

SimCIS provides stable built-in objectness. Although pixel features can generally provide semantic priors across various methods, our observations indicate that they are still influenced by the continual learning process. In this section, we visually demonstrate that our SimCIS has the ability to maintain object information. As shown in Fig. 2, in the first image, the clustering results of Balconpas around the jeep exhibit significantly more noise. In the last image, Balconpas fails to capture the entire helicopter, while our feature successfully preserves the complete object information.

8. The Order of Attention Layers

In Mask2Former [6], the authors employ a cross then self-attention mechanism, as they argue that query features to the first self-attention layer are image-independent and do not have signals from the image, thus applying self-attention is unlikely to enrich information. However, in our proposed Lazy Query Pre-alignment strategy, the query features have rich information. Therefore, we revert to the conventional sequence of cross then self-attention. This modification, however, does not exhibit any significant impact on the experimental outcomes.

9. Discussion, Limitation and Future Work

Discussion of the choice of meta-architecture for image segmentation. To ensure a fair comparison, we adopt the same Mask2Former [6] as our meta-architecture for image segmentation. However, recent years have witnessed rapid advancements in transformer-based universal image segmentor [10, 13], which achieves a much stronger performance on the segmentation benchmark. We leave the investigation of other meta-architectures as future work.

Discussion of other common techniques/tricks in CIS. To maintain the simplicity and elegance of our SimCIS, we have discarded certain continual learning techniques/tricks commonly used in previous methods, such as model weight fusion across stages [16], specific initialization methods [1, 4, 17] for the classifier head, and freezing model parameters [8, 12]. Whether these techniques/tricks can further improve SimCIS's performance remains an open question for future work.

ID	Category Order
1	*[71, 135, 3, 60, 74, 1, 10, 40, 118, 91, 52, 50, 59, 146, 33, 42, 66, 148, 41, 78, 46, 14, 26, 57, 73, 96, 89, 55, 149, 84, 13, 2, 77, 54, 32, 138, 64, 81, 129, 104, 93, 86, 62, 130, 21, 125, 128, 136, 12, 65, 79, 43, 4, 134, 68, 145, 99, 15, 58, 29, 111, 51, 56, 11, 117, 102, 140, 105, 116, 131, 18, 120, 22, 19, 85, 28, 0, 123, 38, 95, 115, 17, 70, 61, 20, 112, 109, 67, 98, 133, 30, 76, 49, 8, 101, 47, 25, 48, 147, 132, 100, 44, 69, 6, 53, 126, 7, 75, 90, 83, 107, 106, 9, 113, 37, 122, 121, 143, 103, 137, 80, 144, 94, 142, 110, 63, 124, 87, 35, 24, 88, 39, 139, 27, 92, 23, 114, 119, 141, 108, 5, 45, 72, 31, 36, 127, 82, 16, 97, 34]*
2	*[11, 114, 103, 122, 48, 41, 85, 92, 113, 64, 3, 80, 110, 10, 112, 30, 96, 101, 102, 9, 7, 21, 17, 37, 93, 77, 73, 94, 59, 135, 2, 123, 98, 130, 49, 129, 25, 66, 50, 145, 76, 147, 83, 90, 63, 111, 27, 126, 1, 65, 75, 119, 12, 78, 5, 143, 15, 29, 71, 22, 89, 115, 84, 16, 120, 139, 38, 68, 146, 116, 35, 124, 97, 23, 39, 117, 13, 18, 108, 138, 33, 134, 141, 62, 105, 142, 40, 26, 8, 46, 144, 95, 131, 99, 104, 19, 60, 132, 6, 42, 4, 140, 128, 55, 32, 70, 118, 100, 125, 127, 87, 52, 45, 31, 81, 88, 44, 24, 20, 56, 82, 61, 28, 34, 148, 14, 53, 121, 47, 133, 57, 137, 67, 136, 106, 36, 58, 109, 107, 72, 91, 86, 43, 74, 69, 0, 149, 51, 79, 54]*
3	*[74, 149, 75, 46, 113, 67, 118, 89, 130, 7, 119, 33, 77, 39, 96, 81, 112, 37, 124, 1, 34, 105, 35, 80, 135, 13, 143, 53, 9, 101, 22, 57, 139, 138, 12, 123, 48, 63, 60, 69, 117, 71, 4, 65, 127, 84, 97, 59, 70, 91, 128, 142, 41, 99, 136, 32, 108, 120, 42, 145, 148, 104, 87, 132, 52, 5, 85, 61, 10, 121, 49, 44, 17, 115, 93, 134, 68, 3, 110, 36, 133, 102, 0, 16, 55, 90, 83, 54, 62, 94, 126, 6, 19, 18, 26, 51, 114, 31, 43, 45, 76, 131, 25, 66, 92, 29, 50, 40, 100, 58, 109, 20, 30, 98, 86, 14, 28, 107, 122, 11, 111, 64, 21, 72, 103, 137, 23, 88, 125, 140, 47, 146, 27, 116, 141, 78, 79, 24, 95, 2, 144, 38, 82, 56, 106, 129, 147, 73, 8, 15]*
4	*[60, 110, 89, 119, 147, 123, 116, 35, 22, 1, 36, 99, 58, 17, 43, 11, 109, 130, 113, 138, 65, 94, 74, 8, 106, 12, 29, 118, 24, 136, 140, 21, 6, 93, 142, 9, 71, 135, 54, 114, 121, 77, 16, 105, 117, 5, 67, 86, 61, 97, 20, 76, 18, 84, 103, 46, 96, 0, 141, 100, 63, 131, 31, 45, 81, 73, 13, 124, 79, 48, 40, 132, 102, 112, 107, 44, 27, 49, 134, 85, 144, 66, 83, 104, 75, 88, 101, 82, 19, 47, 87, 122, 125, 115, 72, 137, 7, 128, 78, 15, 90, 51, 145, 39, 2, 126, 64, 139, 41, 55, 34, 26, 3, 129, 69, 68, 120, 98, 92, 57, 59, 70, 23, 80, 148, 10, 149, 52, 38, 42, 53, 108, 127, 91, 50, 95, 146, 56, 33, 30, 111, 25, 62, 32, 4, 37, 14, 143, 133, 28]*
5	*[77, 20, 111, 65, 117, 53, 43, 90, 28, 79, 134, 45, 116, 98, 92, 105, 137, 10, 6, 59, 67, 34, 44, 99, 55, 147, 1, 80, 122, 54, 56, 12, 31, 49, 37, 61, 108, 133, 143, 130, 70, 95, 132, 2, 115, 118, 81, 47, 51, 121, 14, 3, 8, 21, 22, 62, 78, 72, 39, 25, 23, 142, 149, 50, 83, 11, 52, 141, 129, 113, 4, 148, 144, 136, 91, 146, 35, 114, 46, 138, 97, 16, 69, 84, 131, 64, 66, 5, 24, 13, 68, 9, 102, 104, 139, 106, 74, 126, 19, 0, 58, 60, 96, 32, 41, 94, 7, 48, 93, 30, 119, 75, 42, 15, 57, 38, 127, 120, 124, 100, 135, 123, 63, 33, 103, 71, 128, 17, 145, 26, 86, 29, 107, 82, 88, 73, 110, 112, 85, 89, 27, 125, 109, 40, 76, 87, 36, 101, 18, 140]*
6	*[54, 27, 42, 13, 38, 94, 134, 97, 95, 109, 130, 26, 117, 67, 107, 96, 69, 78, 141, 113, 4, 147, 129, 108, 144, 145, 49, 44, 128, 115, 148, 104, 19, 58, 114, 89, 98, 21, 106, 39, 138, 63, 43, 7, 12, 17, 81, 84, 103, 45, 120, 5, 23, 142, 143, 14, 102, 56, 116, 112, 136, 60, 50, 92, 65, 82, 127, 139, 8, 91, 10, 93, 131, 83, 73, 74, 85, 75, 121, 105, 40, 25, 123, 149, 118, 52, 29, 88, 126, 51, 110, 1, 122, 133, 47, 99, 137, 80, 55, 57, 62, 71, 125, 140, 32, 20, 2, 61, 132, 30, 111, 37, 76, 64, 15, 77, 79, 28, 33, 100, 31, 124, 72, 119, 9, 6, 90, 36, 16, 68, 22, 59, 86, 18, 0, 70, 53, 3, 34, 41, 46, 35, 24, 135, 146, 101, 66, 87, 11, 48]*
7	*[87, 70, 74, 1, 60, 111, 0, 26, 59, 35, 57, 128, 55, 24, 20, 53, 108, 49, 140, 29, 54, 6, 84, 10, 101, 5, 94, 32, 79, 63, 15, 9, 31, 107, 110, 104, 38, 33, 77, 132, 43, 149, 72, 119, 37, 56, 112, 114, 124, 13, 51, 58, 47, 83, 69, 45, 11, 145, 127, 123, 52, 97, 98, 8, 73, 95, 117, 86, 46, 89, 65, 93, 62, 61, 129, 28, 39, 125, 78, 67, 133, 120, 14, 99, 21, 141, 121, 7, 136, 42, 88, 17, 146, 19, 131, 96, 102, 4, 34, 44, 30, 22, 50, 90, 142, 137, 81, 82, 16, 118, 130, 100, 103, 64, 18, 113, 135, 41, 12, 85, 2, 115, 147, 134, 80, 76, 66, 68, 36, 109, 3, 105, 106, 92, 75, 138, 148, 27, 126, 71, 40, 48, 25, 139, 91, 122, 116, 23, 143, 144]*
8	*[22, 119, 103, 67, 40, 38, 95, 43, 72, 34, 54, 88, 132, 94, 0, 107, 91, 104, 71, 21, 133, 16, 1, 27, 48, 125, 139, 144, 35, 75, 129, 25, 53, 82, 117, 7, 140, 124, 128, 147, 120, 23, 70, 122, 108, 106, 93, 12, 90, 73, 149, 99, 52, 47, 146, 28, 61, 55, 37, 87, 76, 136, 112, 148, 29, 57, 49, 45, 65, 100, 13, 32, 68, 78, 58, 69, 56, 2, 9, 130, 110, 51, 116, 123, 111, 118, 101, 19, 138, 59, 109, 4, 85, 98, 17, 141, 131, 50, 92, 8, 81, 30, 6, 41, 79, 97, 46, 74, 126, 115, 31, 11, 15, 3, 33, 5, 63, 105, 83, 62, 64, 134, 39, 137, 113, 36, 42, 10, 18, 114, 145, 80, 84, 66, 60, 77, 86, 89, 14, 127, 24, 96, 121, 142, 20, 143, 26, 44, 135, 102]*
9	*[83, 53, 93, 75, 14, 89, 54, 2, 115, 80, 110, 24, 56, 124, 62, 113, 1, 30, 100, 107, 86, 82, 87, 95, 129, 149, 0, 130, 143, 103, 43, 122, 29, 106, 19, 34, 5, 17, 74, 90, 6, 97, 44, 139, 51, 31, 35, 135, 96, 9, 72, 18, 66, 33, 40, 126, 125, 91, 23, 145, 94, 77, 3, 78, 49, 27, 7, 50, 63, 28, 41, 55, 84, 73, 123, 42, 38, 8, 102, 109, 112, 119, 65, 121, 144, 88, 133, 132, 25, 114, 134, 105, 92, 10, 11, 120, 79, 26, 47, 16, 46, 137, 71, 141, 117, 48, 20, 101, 142, 15, 104, 21, 127, 136, 147, 140, 128, 32, 108, 70, 57, 98, 69, 45, 22, 111, 12, 99, 59, 60, 36, 52, 116, 58, 13, 68, 76, 4, 131, 146, 67, 39, 148, 37, 138, 64, 118, 85, 61, 81]*
10*	*[149, 148, 147, 146, 145, 144, 143, 142, 141, 140, 139, 138, 137, 136, 135, 134, 133, 132, 131, 130, 129, 128, 127, 126, 125, 124, 123, 122, 121, 120, 119, 118, 117, 116, 115, 114, 113, 112, 111, 110, 109, 108, 107, 106, 105, 104, 103, 102, 101, 100, 99, 98, 97, 96, 95, 94, 93, 92, 91, 90, 89, 88, 87, 86, 85, 84, 83, 82, 81, 80, 79, 78, 77, 76, 75, 74, 73, 72, 71, 70, 69, 68, 67, 66, 65, 64, 63, 62, 61, 60, 59, 58, 57, 56, 55, 54, 53, 52, 51, 50, 49, 48, 47, 46, 45, 44, 43, 42, 41, 40, 39, 38, 37, 36, 35, 34, 33, 32, 31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0]*

Table 4. **Random orders.**

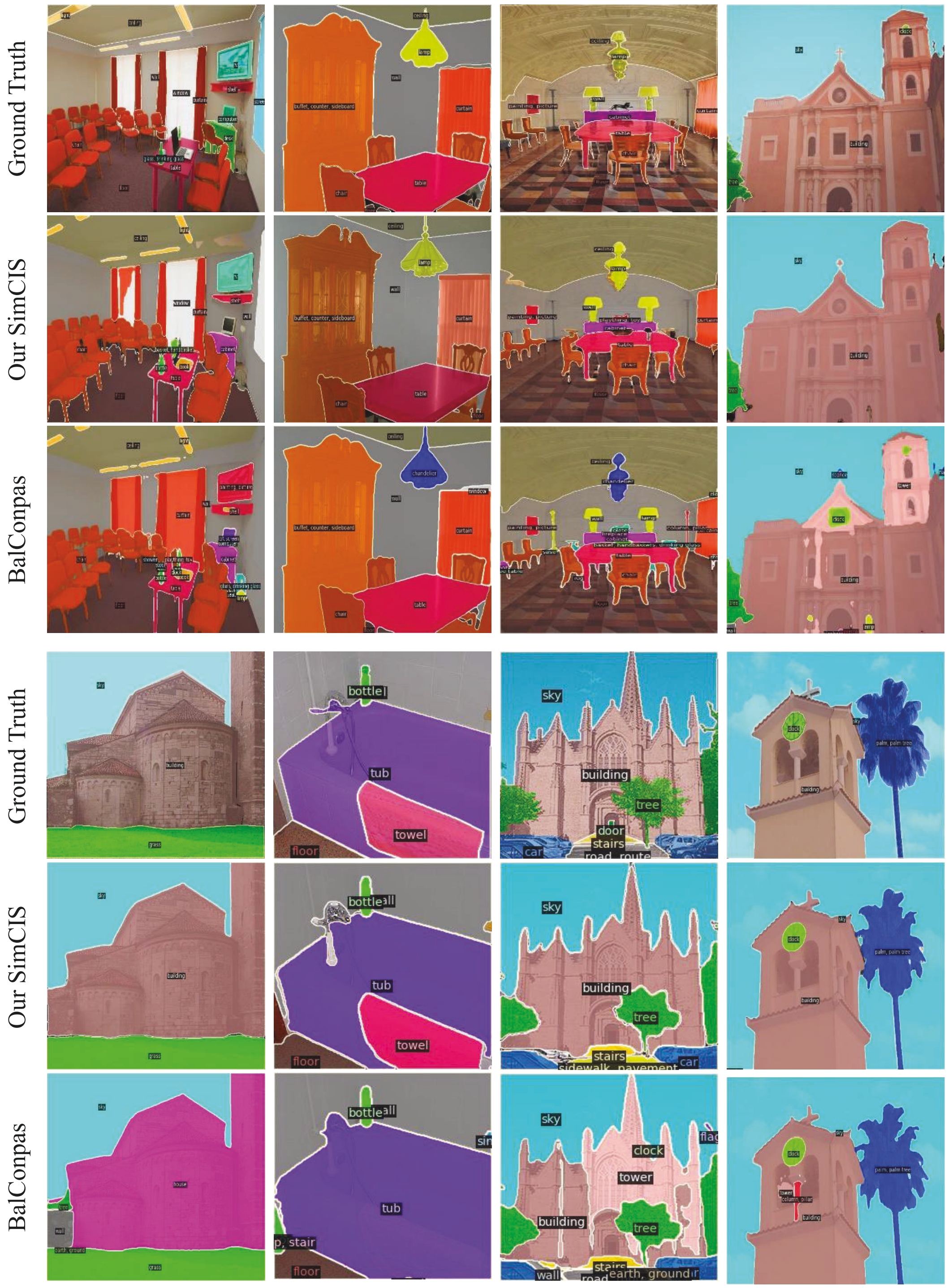

Figure 1. **Qualitative comparisons** between SimCIS and BalConpas [5] on the ADE20K 100-5 continual semantic segmentation.

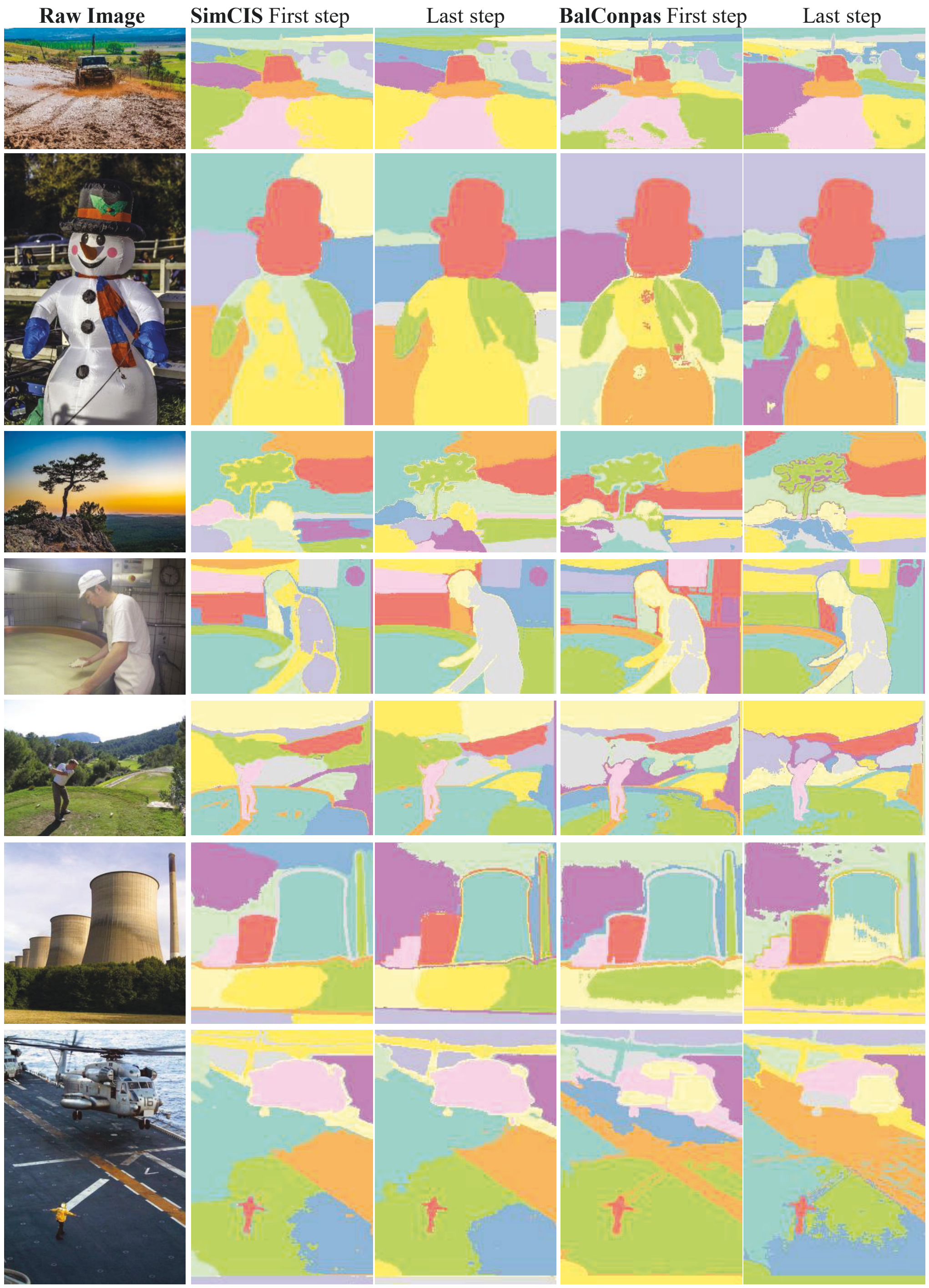

Figure 2. **Clustering results comparison between SimCIS and BalConpas.** Our SimCIS maintains the semantic priors in the pixel feature.

References

[1] Donghyeon Baek, Youngmin Oh, Sanghoon Lee, Junghyup Lee, and Bumsub Ham. Decomposed knowledge distillation for class-incremental semantic segmentation. *Advances in Neural Information Processing Systems*, 35:10380–10392, 2022. 4

[2] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulo, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9233–9242, 2020. 2

[3] Fabio Cermelli, Matthieu Cord, and Arthur Douillard. Comformer: Continual learning in semantic and panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3010–3020, 2023. 2, 3

[4] Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. *Advances in neural information processing systems*, 34:10919–10930, 2021. 4

[5] Jinpeng Chen, Runmin Cong, Yuxuan Luo, Horace Ho Shing Ip, and Sam Kwong. Strike a balance in continual panoptic segmentation, 2024. 4, 6

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 3, 4

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010. 2

[8] Yizheng Gong, Siyue Yu, Xiaoyang Wang, and Jimin Xiao. Continual segmentation with disentangled objectness learning and class recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3848–3857, 2024. 2, 4

[9] JA Hartigan. Clustering algorithms. *John Wiley google schola*, 2:25–47, 1975. 4

[10] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 4

[11] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer, 2022. 3

[12] Beomyoung Kim, Joonsang Yu, and Sung Ju Hwang. Eclipse: Efficient continual learning in panoptic segmentation with visual prompt tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3346–3356, 2024. 2, 3, 4

[13] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3041–3050, 2023. 4

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[15] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2

[16] Jia-Wen Xiao, Chang-Bin Zhang, Jiekang Feng, Xialei Liu, Joost van de Weijer, and Ming-Ming Cheng. Endpoints weight fusion for class incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7204–7213, 2023. 4

[17] Zhengyuan Xie, Haiquan Lu, Jia-wen Xiao, Enguang Wang, Le Zhang, and Xialei Liu. Early preparation pays off: New classifier pre-tuning for class incremental semantic segmentation. In *European Conference on Computer Vision*, pages 183–201. Springer, 2025. 4

[18] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 2