Learning Yourself: Class-Incremental Semantic Segmentation with Language-Inspired Bootstrapped Disentanglement

Supplementary Material

1. Model Details

1.1. Visual Encoder

Since the original version of CLIP [1, 4] was trained on classification tasks at the image level, it cannot be directly applied to segmentation tasks. To address this, we synthesized insights from existing methods and implemented the following improvements (all encoders are based on the transformer architecture):

1. Following MaskCLIP [6], we removed the average pooling in the last layer of the CLIP visual encoder ViT, which allows us to obtain dense features.
2. Following ClearCLIP [3], we directly removed the feedforward neural network and residual connections from the last layer of ViT. Additionally, we replaced the attention mechanism in the final layer with v-v attention.
3. Inspired by the concept of multi-scale feature extraction [2], we first extracted features from different layers of the CLIP visual encoder (specifically, the 4th, 6th, 8th, and 12th layers), concatenated them along the feature dimension, and then used convolution operations to restore the previous dimensions. This feature was then used as input to the decoder.

1.2. Text Encoder

To obtain class templates, we first extracted the corresponding language features from multiple text descriptions containing the class information and then computed the average of the multiple features for each class. The descriptions we used include:

- A photo of a {}.
- A snapshot of a {}.
- A bad photo of the {}.
- A clean origami {}.
- A photo of the large {}.
- A {} in a video game.
- Art of the {}.
- A photo of the small {}.
- A {} in the scene.

2. Analysis of Computational Cost

In the domain of Continual Learning (CL), model efficiency is as crucial as performance. To provide a clear perspective on the computational overhead of our proposed Language-inspired Bootstrapped Disentanglement (LBD) method, we conduct a comparative analysis against DenseCLIP [5], a strong baseline that adapts the CLIP model for dense prediction tasks. This analysis is crucial for contextualizing the performance gains documented in the main paper.

Table 1. Computational and performance comparison. Our LBD method significantly outperforms DenseCLIP with only a minor increase in computational cost. Notably, key components of LBD are training-only and do not affect inference speed.

Method	DenseCLIP (Zero-shot)	DenseCLIP (Continual-train)	LBD (Ours)	Joint
VOC 15-1 All	61.2	68.7	78.1	83.0
Params (M)	105.3	105.3	121.1	-
GFLOPs	143.8	143.8	148.2	-

Our evaluation, summarized in Table 1, focuses on three key metrics: performance (mIoU on VOC 15-1 All), model size (Parameters), and computational load (GFLOPs). We assess DenseCLIP in both its zero-shot capacity and after being continually trained on the same CISS task protocol as our LBD. The results reveal that LBD achieves a mIoU of 78.1, substantially outperforming the continually-trained DenseCLIP (68.7). Regarding the computational budget, LBD exhibits only a marginal increase in complexity. The GFLOPs increase from 143.8 to 148.2, a modest rise of approximately 3%. This slight overhead is primarily attributed to the learnable prompts and the lightweight adapter module. The increase in parameters from 105.3M to 121.1M similarly reflects the inclusion of these task-specific components.

Crucially, it is important to note that our core architectural innovations, such as the Language-guided Prototypical Disentanglement (LPD) module, are designed to operate **exclusively during the training phase**. These components guide the model's feature space to form a disentangled semantic structure but are detached for inference. Consequently, they introduce no additional computational burden at deployment time. Given the substantial performance improvements, especially in challenging multi-step CISS scenarios, we conclude that the minor increase in training computation is a well-justified trade-off.

3. Exploration of PEFT

The advent of large-scale pre-trained models has spurred the development of Parameter-Efficient Fine-Tuning (PEFT) methods, which aim to adapt these models to downstream tasks by updating only a small fraction of their parameters. To assess the feasibility of this paradigm for Class-

Incremental Semantic Segmentation (CISS), we conducted an ablation study investigating different PEFT strategies within our LBD framework.

While our primary experiments configure the visual encoder (CLIP-ViT) as fully trainable to maximize adaptation, integrating PEFT is indeed a feasible alternative. Our study, presented in Table 2, explores the impact of selectively training different components: ❶ the learnable prompts introduced in Section 3.2, ❷ a convolution-based adapter module placed after the encoder, and ❸ the full image encoder itself.

The results yield a clear insight: while PEFT approaches show promise, they currently do not match the performance of full fine-tuning for the demanding task of CISS. Training only the prompts (❶) or the adapter (❷) results in mIoU scores of 64.8 and 66.9, respectively. Combining these two PEFT techniques (❶+❷) improves the score to 72.1. However, this is still considerably lower than the 78.1 mIoU achieved when the visual encoder is fully trained (❶+❷+❸).

This performance gap suggests that adapting the vision-language model to a dense, pixel-level prediction task like semantic segmentation requires more than just peripheral modifications. The supervised signal from pixel-level annotations appears crucial for fundamentally reshaping the features within the visual backbone, an adaptation that cannot be fully achieved when the encoder is frozen. We conclude that while PEFT offers a promising avenue for reducing the training cost of CISS, future work is needed to develop more sophisticated methods that can bridge this performance gap.

Table 2. Ablation study on integrating PEFT methods within our framework on Pascal VOC 15-1 *All*. We evaluate training different combinations of: ❶ Prompts, ❷ Adapter, and ❸ the full Image Encoder. Full fine-tuning of the encoder remains essential for achieving top performance.

Reference	❶ Prompts (Sec.3.2) ❷ Adapter (after encoder) ❸ Image Encoder (CLIP-ViT)				
Trainable	❶	❷	❶❷	❶❸	❶❷❸
VOC 15-1 All	64.8	66.9	72.1	77.4	78.1

4. Limitations

Our method relies on explicit class names, and when only images and numeric labels are available in the dataset, we are unable to leverage textual information. Moreover, due to the limitations of CLIP's pretraining data, CLIP fails to capture the semantic relationships between rare concepts and other classes, thus restricting the effectiveness of our method. Future work could focus on text supervision methods more suitable for incremental learning and cross-modal feature interaction.

References

[1] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, 2022. 1

[2] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:1904–1916, 2014. 1

[3] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. Clearclip: Decomposing CLIP representations for dense vision-language inference. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part XLVII*, pages 143–160. Springer, 2024. 1

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, pages 8748–8763. PMLR, 2021. 1

[5] Yongming Rao, Wenliang Zhao, Guangyi Chen, Yansong Tang, Zheng Zhu, Guan Huang, Jie Zhou, and Jiwen Lu. Denseclip: Language-guided dense prediction with context-aware prompting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18061–18070. IEEE, 2022. 1

[6] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII*, pages 696–712. Springer, 2022. 1