DeOP-Light: Analyzing the Fine-Tuning Paradox in CLIP Visual Encoder Compression for Open-Vocabulary Semantic Segmentation

Junghun Kim and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon-si, Gyeonggi-do 16419, Republic of Korea
{hun0919,jpjeong}@skku.edu

Abstract. Open-vocabulary semantic segmentation (OVSS) relies on contrastive language-image pre-training (CLIP) visual encoders to classify *unseen* categories at inference time. Compressing these encoders for efficient deployment is practically important, yet the impact on the critical *seen/unseen* class distinction has not been studied. We present a diagnostic empirical study rather than a new architectural proposal of CLIP visual encoder compression within the decoupled one-pass (DeOP), applying floating-point (FP)16 quantization and low-rank adaptation (LoRA) on COCO-Stuff and Pascal VOC. We uncover a consistent asymmetric degradation: fine-tuning (FT) the CLIP visual encoder recovers *seen* class performance but simultaneously degrades *unseen* class performance, emerging within the first 1,000 iterations. By comparing against an uncompressed FT baseline, we show that FT itself not compression is the primary driver of this degradation. FT-free compression best preserves *unseen* performance, and post-hoc weight restoration partially recovers it without retraining. Our findings provide practical guidelines for model compression in resource-constrained deployment and highlight the need for *seen/unseen*-aware evaluation in OVSS.

Keywords: Open-vocabulary semantic segmentation · CLIP · Model compression · Catastrophic forgetting · Zero-shot generalization · LoRA

1 Introduction

Semantic segmentation has traditionally required training with a closed set of predefined categories. Open-vocabulary semantic segmentation (OVSS) breaks this limitation by leveraging pre-trained vision-language models (VLMs) such as contrastive language-image pre-training (CLIP) [1] to classify image regions into an arbitrary set of categories including those never *seen* during training, referred to as *unseen* classes. State-of-the-art OVSS methods [2–4] decouple the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 185–201, 2027.
https://doi.org/10.1007/978-3-032-30488-9_12

mask proposal and classification streams, keeping the CLIP visual encoder frozen to preserve its zero-shot generalization capability.

Despite impressive performance, deploying CLIP-based OVSS models at scale raises practical concerns, particularly for resource-constrained environments such as edge devices, autonomous systems, and real-time data analytics pipelines where both efficiency and generalization to novel categories are critical. The large CLIP vision transformer (ViT)-B/16 [5] backbone accounts for the majority of model parameters and inference cost. Model compression methods including floating-point (FP)16 quantization and low-rank adaptation (LoRA) [6] are natural candidates for reducing this overhead. However, a critical question remains unanswered: *does compressing the CLIP visual encoder affect seen* and *unseen* class performance differently?

This question is far from trivial.OVSS performance is conventionally reported as overall mean intersection-over-union (mIoU), which aggregates *seen* and *unseen* classes. An improvement or negligible drop in mIoU could mask a catastrophic drop in *unseen* performance compensated by improved *seen* performance undermining the core value proposition of OVSS. Prior work on CLIP fine-tuning (FT) for image classification has shown that adapting the model to a specific distribution improves in-distribution accuracy while degrading robustness to distribution shifts [7]. We hypothesize an analogous phenomenon in OVSS: compression combined with FT recovers *seen* performance (the training distribution) at the cost of *unseen* generalization (an out-of-distribution (OOD) challenge).

Crucially, OVSS presents a structurally distinct challenge compared to standard image classification. In classification, *seen* and *unseen* classes are evaluated on separate datasets, and the CLIP encoder serves as a feature extractor for one task at a time. In OVSS, however, the CLIP visual encoder must simultaneously serve as a shared feature backbone for both *seen* and *unseen* classes within the same image. This shared-encoder constraint means that any feature-space distortion caused by FT asymmetrically affects the two class groups, creating a degradation pattern that cannot be detected by aggregate metrics alone. In this paper, we confirm and characterize this hypothesis through a systematic study of CLIP visual encoder compression within the decoupled one-pass (DeOP) [4]. We refer to our experimental configurations collectively as DeOP-Light to denote DeOP variants with a compressed CLIP visual encoder. We emphasize that DeOP-Light is not a new architectural proposal but a diagnostic study and design principle: a controlled experimental setting designed to isolate and analyze the effects of compression and FT on the *seen/unseen* performance gap, and to inform principled deployment choices for CLIP-based OVSS systems.

Our main contributions of this study are summarized as follows:

– We apply FP16 quantization and LoRA to the CLIP visual encoder in the DeOP, and evaluate each on COCO-Stuff and Pascal VOC with separate *seen* and *unseen* metrics. While the general phenomenon of FT degrading OOD robustness is known, no prior work has quantified this effect in the *seen/unseen* split specific to OVSS.

- We define the FT Paradox as the phenomenon where FT the CLIP visual encoder consistently recovers *seen* performance while degrading *unseen* performance an asymmetric degradation that undermines the core open-vocabulary capability. Through a FT iteration ablation (no-FT, $1k$, $5k$ and $10k$), we demonstrate that this degradation emerges within the first 1,000 iterations and does not recover with continued training.
- By comparing compressed fine-tuned models against an uncompressed fine-tuned baseline (FP32 FT), we show that FT itself not compression is the primary driver of unseen degradation. Compression plays a secondary, modulatory role.
- FT-free compression best preserves *unseen* generalization. The paradox is more pronounced on COCO-Stuff (171 classes) than Pascal VOC (20 classes), indicating that dataset complexity amplifies the effect.
- Inspired by WiSE-FT, we investigate weight-space interpolation and find that unlike the image classification setting the optimal strategy in OVSS is full restoration of the original CLIP visual encoder weights ($\alpha = 0$), providing a training-free recovery of *unseen* performance.

The remainder of this paper is organized as follows. Section 2 reviews related work; Sect. 3 describes the DeOP-Light framework; Sect. 4 presents the main experiments; Sect. 5 discusses practical guidelines and limitations; Sect. 6 outlines future directions; Sect. 7 conclude the paper.

2 Related Work

2.1 Open-Vocabulary Semantic Segmentation

OVSS aims to segment images into regions belonging to both *seen* (base) and *unseen* (novel) categories [8]. CLIP [1] is a VLM pre-trained on 400 million image-text pairs using contrastive learning, which learns to align visual and textual representations in a shared embedding space; this alignment enables zero-shot recognition of arbitrary categories described by text, making CLIP a natural backbone for OVSS. Early approaches such as ZS3Net [8] and SPNet [9] used generative or projection-based methods but were limited to simple datasets. OpenSeg [10] explored the use of ALIGN [11] for segment classification. ZegFormer [2] and SimBaseline [3] introduced the decoupled two-stream design, separating mask proposal from CLIP-based classification.

DeOP [4] our base framework further improved efficiency by enabling a single-pass through the CLIP encoder via its patch severance (PS) and classification anchor learning (CAL) modules. Subsequent methods have further advanced OVSS: SAN [12] introduced a side adapter network for efficient open-vocabulary segmentation, CAT-Seg [13] proposed cost aggregation for improved spatial alignment, and FC-CLIP [14] achieved strong performance with a single frozen CLIP backbone. More recently, training-free OVSS methods such as NACLIP [15] and ClearCLIP [16] have demonstrated that competitive dense prediction can be achieved without any FT of CLIP, which aligns with our finding that preserving the original CLIP weights is beneficial for *unseen* class performance.

2.2 CLIP Compression and Adaptation

Compressing large VLMs for efficient deployment is an active research area. Model compression approaches include quantization [17,18] and parameter-efficient FT (PEFT) such as LoRA [6] and prompt tuning [19]. TinyCLIP [20] demonstrated that CLIP can be compressed to 50% of its original size via affinity mimicking and weight inheritance while retaining much of its zero-shot accuracy. CLIP-KD [21] systematically studied various distillation losses for compressing CLIP models across different frameworks. Prompt learning methods such as CoOp [22] adapt CLIP to downstream tasks by learning continuous prompt vectors, achieving strong generalization with minimal parameter updates. Token reduction methods for ViTs [23] have demonstrated significant speedups on image classification, though their interaction with dense prediction frameworks remains underexplored. However, existing compression and adaptation methods are evaluated solely on overall accuracy, without distinguishing *seen* from unseen categories a critical distinction for OVSS.

2.3 Catastrophic Forgetting in Foundation Models

FT pre-trained models on specific tasks often leads to catastrophic forgetting of their original generalization capabilities [24]. For CLIP, Wortsman et al. showed that standard FT on ImageNet improves in-distribution accuracy but reduces out-of-distribution robustness, proposing weight-space ensembling-FT (WiSE-FT) as a mitigation [7]. Zheng et al. observed similar forgetting in continual learning of VLMs [25]. Akbarian Bafghi et al. demonstrated that selective LoRA FT reduces forgetting of zero-shot capabilities [26]. Our work extends these observations to the specific scenario of CLIP visual encoder compression in OVSS. While the general phenomenon of FT degrading OOD robustness is well established, we provide the first quantitative evidence of its asymmetric impact on *seen* versus *unseen* classes within the OVSS evaluation framework.

3 DeOP-Light Framework

DeOP-Light builds upon the DeOP [4] by applying compression exclusively to its CLIP visual encoder. This section describes the problem setting and the base framework on which all compression experiments are conducted. Figure 1 illustrates the overall pipeline of DeOP-Light. Following [3], the class set is partitioned into *seen* classes C_s (available during training) and *unseen* classes C_u (only at inference), with $C_s \cap C_u = \emptyset$. A model is trained on images annotated with C_s and evaluated on both C_s and C_u at test time. Performance is reported as *seen* IoU, *unseen* IoU, and their harmonic mean (hIoU), which equally weights both.

DeOP adopts a two-stream framework that combines a class-agnostic mask proposal network with a CLIP-based mask classification stream. The proposal stream is based on MaskFormer [27] with a ResNet-101c backbone, whereas the

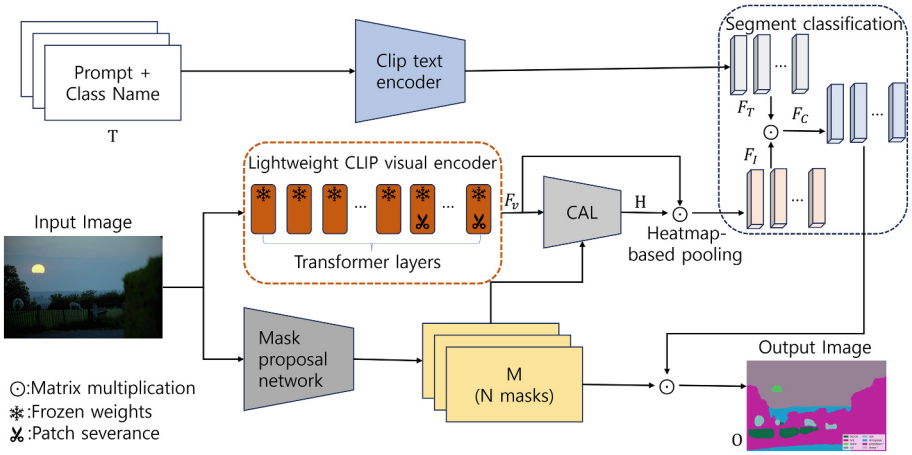

Fig. 1. Overview of the DeOP-Light framework. Compression is applied exclusively to the CLIP visual encoder (highlighted in red dashed border). All other components mask proposal network, CLIP text encoder, and CAL module remain unchanged from the original DeOP. (Color figure online)

classification stream uses a frozen CLIP ViT-B/16 visual encoder. Given an input image I, the mask proposal network generates N proposal masks M. The CLIP visual encoder equipped with PS produces a feature map F_V. The CAL module then generates heatmaps H from F_V and M, which are used for heatmap-based pooling to obtain segment embeddings $F_I = H \odot F_V$. Classification is performed by matching F_I against text embeddings F_T from the CLIP text encoder: $F_C = F_T \cdot F_I$. The final segmentation output is obtained by combining the class scores with the proposal masks: $O = F_C \cdot M$.

In DeOP-Light, the CLIP visual encoder is our sole target for compression. All other components (MaskFormer backbone, CAL module, text encoder and prompt learner) remain unchanged from the original DeOP. This design isolates the effect of visual encoder compression on the *seen/unseen* performance gap, which is the central question of this study. We apply two representative compression methods to the CLIP ViT-B/16 visual encoder within DeOP-Light. Importantly, these techniques target only the 158 parameter tensors (weights and biases) of the visual encoder, out of 308 total CLIP parameter tensors, leaving the text encoder and all other modules untouched. We convert all non-LayerNorm parameters in the CLIP visual encoder from FP32 to FP16 precision. LayerNorm parameters are kept in FP32 to maintain numerical stability. For FT variants, the model is trained with FP16 forward pass and FP32 optimizer state (standard mixed-precision training), and the checkpoint is saved in FP32 (which we then re-quantize to FP16 at evaluation time). We note that FP16 represents the most conservative form of quantization; more aggressive methods (e.g., INT8, INT4) are left for future work. We inject LoRA adapters [6] with rank $r = 4$ into the `out_proj` layers of the self-attention modules in the CLIP

visual encoder. LoRA freezes the pre-trained weights W_0 and learns low-rank updates $\Delta W = BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times d}$. With $d = 768$, $r = 4$, and 12 transformer layers, the total number of additional trainable parameters is $12 \times 2 \times r \times d = 12 \times 2 \times 4 \times 768 = 73{,}728 \approx 73.7k$, which is negligible compared to the $86M$-parameter visual encoder. We also considered layer pruning but excluded it because removing transformer layers disrupts the PS module, causing complete performance collapse.

4 Experiments

4.1 Experimental Setup

Datasets. We follow the open-vocabulary evaluation protocol of [3] on two benchmarks. COCO-Stuff-171 [28] uses 156 *seen* classes for training and 15 *unseen* (novel) classes for evaluation. We train on COCO-Stuff-$164k$ and evaluate on the $5k$ validation split. For cross-dataset generalization, we evaluate models trained on COCO-Stuff on Pascal VOC 2012 [29] (15 *seen* / 5 *unseen* classes) without any FT.

All experiments are conducted on a single NVIDIA A100 80GB GPU using the official DeOP codebase with Detectron2 [30]. The baseline model uses the published DeOP checkpoint. For FT experiments, we train for $1k$, $5k$, and $10k$ iterations with batch size 8 using AdamW (learning rate = 10^{-4}, weight decay = 10^{-4}).

4.2 Performance Metrics

We evaluate all models using three metrics that separately capture *seen* and *unseen* class performance. *seen* IoU and *unseen* IoU denote the mIoU computed over *seen* classes C_s and *unseen* classes C_u, respectively. hIoU is the harmonic mean of *seen* and *unseen* IoU, defined as:

$$hIoU = \frac{2 \times seenIoU \times unseenIoU}{seenIoU + unseenIoU}, \tag{1}$$

Which equally penalizes degradation in either category and serves as the primary metric for evaluating the *seen/unseen* trade-off.Frames per second (FPS) is measured on COCO-Stuff with batch size 1 to assess inference efficiency. Note that FP16 fine-tuned models show slightly lower FPS (7.99) than FP16 no-FT (8.33) because the fine-tuned checkpoints are saved in FP32 and re-quantized to FP16 at evaluation, introducing a small overhead from the mixed-precision conversion.

Table 1. DeOP-Light results on COCO-Stuff. All methods compress only the CLIP visual encoder. "no-FT" denotes no FT after compression. FPS, V.E. Mem., and Peak GPU are measured on an A100 80 GB at 512×512 input with batch 1.

Method	FT Iter.	*seen*	*unseen*	*h*IoU	FPS	V.E. Mem(*MB*)	Peak GPU(*MB*)
DeOP (Baseline)		37.97	38.41	38.19	7.34	338.94	1808.59
DeOP-Light FP16 no-FT	0	37.00	35.12	36.03	8.33	169.55	1614.58
DeOP-Light FP16 FT	1*k*	36.31	31.08	33.49	7.99	169.55	1619.91
DeOP-Light FP16 FT	5*k*	38.23	31.83	34.74	7.99	169.55	1614.49
DeOP-Light FP16 FT	10*k*	36.53	29.80	32.82	7.99	169.55	1619.31
DeOP-Light LoRA (r=4) FT	1*k*	36.37	30.32	33.07	8.33	339.22	1811.24
DeOP-Light LoRA (r=4) FT	5*k*	38.23	32.13	34.92	8.33	339.22	1809.05
DeOP-Light LoRA (r=4) FT	10*k*	38.40	31.58	34.66	8.33	339.22	1807.75

4.3 Quantitative Analysis

Table 1 shows a clear *seen/unseen* trade-off. Fine-tuned variants recover or improve *seen* IoU (36.3–38.4) but consistently reduce *unseen* IoU (29.8–32.1) relative to the baseline (38.41), whereas FP16 no-FT preserves the highest *unseen* IoU among compressed models (35.12). LoRA 5*k* yields the best fine-tuned *h*IoU (34.92), but LoRA 10*k* already shows a decline in *unseen* IoU, indicating that longer FT further erodes zero-shot generalization.

The efficiency columns show that FP16 halves the CLIP visual encoder weight memory (338.94→169.55*MB*) and reduces peak inference memory by about 10.7%. LoRA adds only 73.7*k* adapter parameters and leaves inference memory nearly unchanged, so its main benefit is training-time parameter efficiency rather than deployment-time weight compression. To test cross-dataset generalization, we evaluate COCO-Stuff-trained models on Pascal VOC without VOC-specific FT. Table 2 shows the results. The VOC baseline *h*IoU (89.62) is higher than the original DeOP intra-dataset result because our setting follows [3]: training on COCO-Stuff and testing on VOC.

Table 2. Cross-dataset comparison on Pascal VOC. Models are trained on COCO-Stuff and evaluated on VOC without FT.

Method	FT Iter.	*seen*	*unseen*	*h*IoU
DeOP (Baseline)		90.31	88.94	89.62
DeOP-Light FP16 no-FT	0	90.31	88.93	89.61
DeOP-Light FP16 FT	5*k*	87.81	83.29	85.49
DeOP-Light LoRA (r=4) FT	5*k*	87.48	85.75	86.60

On VOC, FP16 no-FT is nearly identical to the baseline, while fine-tuned FP16 and LoRA degrade both *seen* and *unseen* IoU. The smaller drops compared with COCO-Stuff suggest that dataset complexity amplifies the *seen/unseen*

trade-off. To identify when degradation begins, we train FP16 and LoRA models for different iteration counts. Table 3 and Figs. 2–3 summarize the results.

Table 3. FT iteration ablation on COCO-Stuff. FP16 and LoRA are evaluated at multiple iteration counts.

Method	FT Iter.	*seen*	*unseen*	*h*IoU
FP16	no-FT	37.00	35.12	36.03
	$1k$	36.31	31.08	33.49
	$5k$	38.23	31.83	34.74
	$10k$	36.53	29.80	32.82
LoRA	$1k$	36.37	30.32	33.07
	$5k$	38.23	32.13	34.92
	$10k$	38.40	31.58	34.66

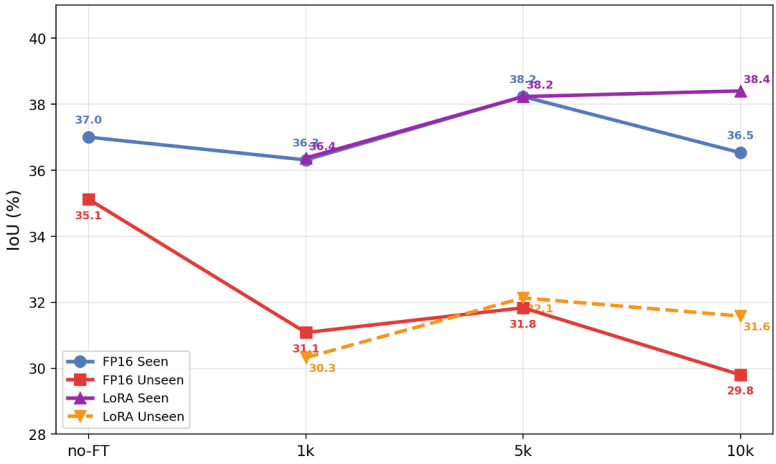

Fig. 2. *seen* and *unseen* IoU as a function of FT iterations on COCO-Stuff. *seen* IoU recovers with FT while *unseen* IoU consistently degrades for both FP16 and LoRA.

unseen degradation appears within the first $1k$ iterations: FP16 drops from 35.12 (no-FT) to 31.08, and LoRA follows the same pattern. Although LoRA is slightly more stable and peaks at $5k$, both methods lose *unseen* performance as FT proceeds. This confirms that the phenomenon is not method-specific; as shown in Table 4, it also occurs without compression.

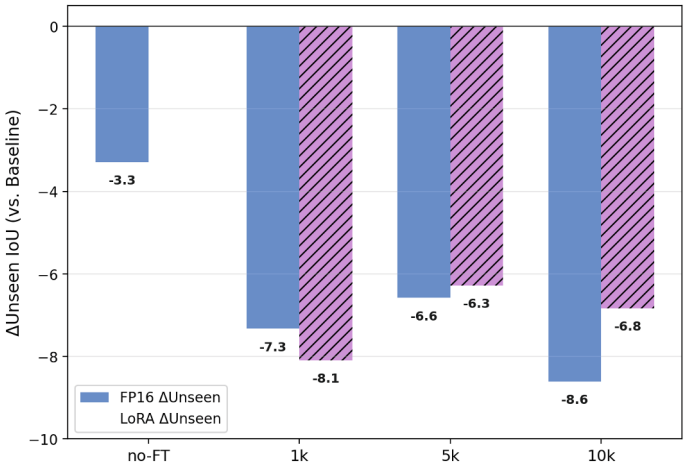

Fig. 3. *unseen* IoU degradation relative to the baseline (38.41). The drop accumulates progressively for both FP16 and LoRA across all iteration counts, confirming the method-agnostic nature of the FT Paradox.

Comparing COCO-Stuff and Pascal VOC, the degradation is stronger on the more complex COCO-Stuff benchmark. FP16 no-FT causes a −3.29 *unseen* drop on COCO-Stuff but only −0.01 on VOC, suggesting that a larger and semantically denser class space increases the risk of confusing *unseen* classes after visual-encoder adaptation. To separate the effects of compression and FT, we fine-tune the original DeOP model without compression (FP32 FT) using the same hyperparameters.

Table 4. Disentangling compression from FT on COCO-Stuff. FP32 FT applies no compression the CLIP visual encoder is fine-tuned in its original FP32 precision.

Method	FT Iter.	*seen*	*unseen*	*h*IoU
Baseline (no FT)		37.97	38.41	38.19
FP32 FT (no compression)	1k	36.04	31.34	33.53
FP32 FT (no compression)	5k	36.47	30.84	33.42
FP16 FT	1k	36.31	31.08	33.49
FP16 FT	5k	38.23	31.83	34.74

FP32 FT produces nearly the same *unseen* degradation as FP16 FT: at 1k, the *unseen* IoU values are 31.34 and 31.08, respectively. Thus, FT the CLIP visual encoder itself is the primary driver of *unseen* degradation, while compression plays a secondary role. This is consistent with Sect. 4.5, where fully restoring the original CLIP visual weights ($\alpha = 0$) gives the best recovery.

4.4 Qualitative Analysis

Figure 4 provides a visual illustration of the asymmetric degradation on a COCO-Stuff validation image containing both *seen* classes (bowl, dining table) and an *unseen* class (scissors). The baseline and FP16 no-FT models correctly identify all three objects, including the *unseen* scissors. After just $1k$ FT iterations, the scissors region shrinks and spurious *seen*-class labels (cardboard, metal) begin to appear. At $5k$ iterations, the scissors are completely lost, and the bowl boundary becomes irregular. At $10k$ iterations, the segmentation quality further degrades with a severely distorted bowl mask. This progression visually confirms the quantitative findings: FT progressively erodes the model's ability to recognize *unseen* classes while introducing spurious *seen*-class predictions, whereas compression without FT preserves the original segmentation quality.

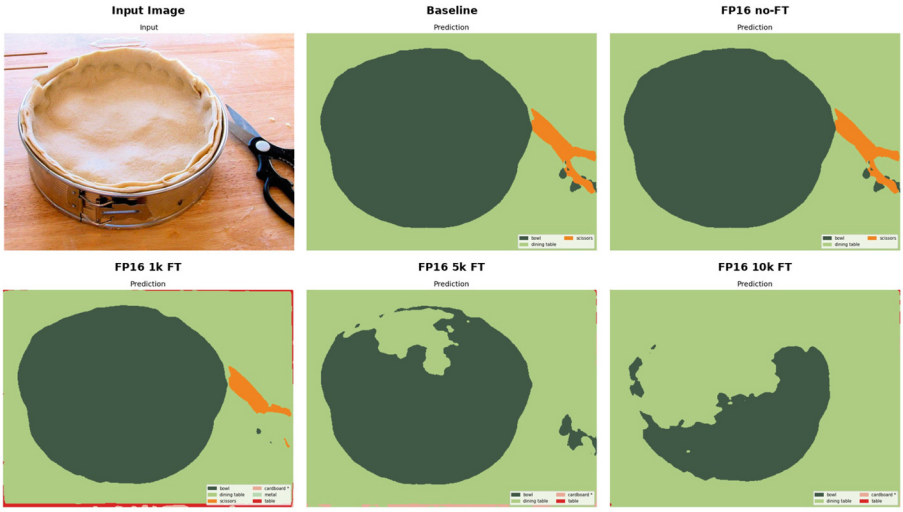

Fig. 4. Qualitative comparison on a COCO-Stuff validation image containing *seen* classes (bowl, dining table) and an *unseen* class (scissors). The Baseline and FP16 no-FT correctly detect all objects including the *unseen* scissors. With increasing FT iterations, the *unseen* class is progressively lost and spurious *seen*-class labels appear.

4.5 Post-hoc Weight Restoration

Having identified the asymmetric *seen/unseen* degradation, we investigate whether it can be mitigated without retraining. Inspired by WiSE-FT [7], we interpolate between the original zero-shot CLIP visual encoder weights W_0 and the fine-tuned compressed weights W_{ft}:

$$W_\alpha = (1 - \alpha) \cdot W_0 + \alpha \cdot W_{\text{ft}}, \tag{2}$$

where $\alpha \in [0, 1]$. At $\alpha = 0$, the model uses the original CLIP visual encoder from the DeOP checkpoint, whereas at $\alpha = 1$, it uses the fine-tuned CLIP visual encoder. The interpolation is restricted to the 158 CLIP visual encoder parameter tensors. All other modules, including the MaskFormer proposal network, CAL module, prompt learner, and CLIP text encoder, are kept fixed and identical across all α values.

Table 5. Weight interpolation comparison on COCO-Stuff (DeOP-Light FP16 5k). $\alpha = 0$: original CLIP visual encoder weights; $\alpha = 1$: fully fine-tuned weights. Interpolation is applied only to the CLIP visual encoder.

α	*seen*	*unseen*	hIoU
0.0 (original)	38.77	33.04	35.68
0.3	38.65	32.58	35.35
0.5	38.57	32.38	35.20
1.0 (fine-tuned)	38.23	31.83	34.74

Table 5 shows a monotonic trend: moving from the fine-tuned weights ($\alpha = 1$) toward the original CLIP weights ($\alpha = 0$) improves both *seen* and *unseen* IoU. The best result is obtained by full restoration, which recovers 1.21 *unseen* points (31.83→33.04) and 0.54 *seen* points (38.23→38.77) over the fully fine-tuned model. This differs from WiSE-FT in image classification, where an intermediate interpolation often balances in-distribution and OOD performance. In OVSS, the CLIP visual encoder is shared by *seen* and *unseen* classes within the same dense prediction task, so any FT-induced bias toward *seen* classes can harm *unseen* recognition. The appropriate mitigation is therefore closer to weight restoration than weight interpolation. However, full restoration still remains below the original uncompressed baseline (*unseen* 38.41), indicating that the fine-tuned non-visual modules have become partially mismatched with the restored CLIP features. FP16 no-FT also preserves *unseen* IoU better than restoration (35.12 *vs.* 33.04), reinforcing our main conclusion that avoiding visual-encoder FT is the safest strategy for preserving zero-shot generalization.

4.6 Training-Time Mitigation: Feature Distillation

While post-hoc weight restoration (Sect. 4.5) provides a training-free mitigation, we also investigate whether training-time regularization can prevent the asymmetric degradation from occurring in the first place. Inspired by recent work on preserving CLIP's zero-shot capabilities through knowledge distillation [21], we add a feature-level distillation loss that encourages the compressed (student) CLIP visual encoder to remain close to the original (teacher) encoder during FT:

$$L_{\text{total}} = L_{\text{DeOP}} + \lambda \cdot L_{\text{KD}}, \quad L_{\text{KD}} = 1 - \cos(F_{\text{student}}, F_{\text{teacher}}), \tag{3}$$

where F_{student} and F_{teacher} are the spatial feature maps from the compressed and original CLIP visual encoders, respectively. The teacher is a frozen deep copy of the original CLIP visual encoder, requiring no additional training.

Table 6. Feature distillation results on COCO-Stuff. The teacher is the original frozen CLIP visual encoder. All models use FP16 compression. "None" denotes standard FT without distillation.

Mitigation	λ	*seen*	*unseen*	*h*IoU
None (FP16 FT 5k)	–	38.23	31.83	34.74
Feature KD (5k)	0.5	21.10	16.85	18.74
Feature KD (5k)	0.01	27.98	22.30	24.82
Feature KD (10k)	0.01	17.74	14.77	16.12

Table 6 reports the results. Contrary to our expectation, feature distillation degrades both *seen* and *unseen* performance compared to standard FT across all tested configurations. With $\lambda = 0.5$, the distillation loss dominates the total loss (accounting for 97% of the gradient signal), effectively preventing the model from learning the segmentation task. Reducing λ to 0.01 restores a balanced loss ratio (KD $\approx$ 54% of classification loss), but performance remains substantially below the non-distilled baseline. Moreover, continued training to 10k iterations worsens performance further (*h*IoU: 24.82→16.12), indicating that the distillation objective actively interferes with task learning over time.

We attribute this failure to a fundamental gradient conflict: the segmentation loss L_{DeOP} requires the visual features to adapt to the downstream task, while the distillation loss L_{KD} constrains them to remain fixed. These two objectives produce opposing gradients on the same parameters, preventing the model from satisfying either goal effectively. This conflict is particularly severe in DeOP's framework, where the PS module modifies self-attention patterns and the CAL module learns spatial heatmaps on top of the visual features both of which require feature adaptation to function properly. This negative result carries an important implication: naive feature-level distillation is not a viable strategy for preserving zero-shot generalization during OVSS compression. The success of post-hoc weight restoration (Sect. 4.5) over training-time distillation suggests that decoupling the two objectives temporally first adapting, then restoring is more effective than forcing them to coexist in a single optimization.

5 Discussion and Limitations

5.1 Practical Guidelines

Based on our analysis, we offer the following practical guidelines for compressing CLIP-based OVSS models, particularly relevant to edge deployment and

resource-constrained inference pipelines. When *unseen* performance is the priority, FT-free compression is the recommended choice: DeOP-Light FP16 no-FT achieves 8.33 FPS (*vs.* 7.34 baseline) with only 3.3-point *unseen* degradation. When *seen/unseen* balance is required and FT is necessary, LoRA at rank r=4 with a limited budget ($5k$ iterations) is preferable, achieving the best fine-tuned hIoU (34.92) and degrading *unseen* IoU less than FP16; FT beyond this point progressively erodes *unseen* generalization, so iterations should be limited and *unseen* performance monitored throughout training. When FT has already been performed, post-hoc weight restoration (Sect. 4.5) provides training-free recovery of *unseen* performance. Finally, regardless of the compression method chosen, we strongly recommend evaluating *seen* and *unseen* IoU separately (alongside their harmonic mean), as overall mIoU can mask catastrophic *unseen* degradation behind improved *seen* performance.

5.2 Limitations

Our study has several limitations that should be acknowledged. All experiments are conducted on a single OVSS framework (DeOP). While the consistency across two different compression methods and two datasets suggests generality, we cannot rule out that DeOP's specific architectural choices (PS, CAL) may amplify or attenuate the observed effects. In particular, DeOP's PS and CAL modules are tightly coupled to the CLIP visual encoder's feature space: PS modifies self-attention patterns in the last transformer layer, and CAL learns spatial heatmaps conditioned on these features. In frameworks without such tight coupling (e.g., SimBaseline [3], which uses independent mask-and-crop passes, or OVSeg [31]), the asymmetric degradation may be less severe, as the visual encoder features are consumed more independently. Validating this hypothesis on additional OVSS frameworks is an important direction for future work.

Our compression study is intentionally restricted to FP16 quantization and LoRA at rank $r = 4$ in order to isolate the FT effect under minimally perturbative compression. While this design choice is essential to our diagnostic objective, the conclusions may not directly generalize to more aggressive and practically relevant settings such as INT8/INT4 quantization, structured pruning, or higher LoRA ranks, which we treat as orthogonal axes of investigation.

All reported numbers are single-run point estimates without variance estimates across multiple training runs. While the observed effect sizes (e.g., 7–8 IoU points of *unseen* degradation under FT) are substantially larger than typical run-to-run variance reported for semantic segmentation models, formal statistical analysis with multiple seeds and confidence intervals would further strengthen the reliability of our conclusions.

6 Future Directions

We demonstrated that restoring the original CLIP visual encoder weights can partially recover *unseen* performance (Sect. 4.5), but the recovered model still

underperforms the uncompressed baseline. Our feature distillation experiment (Sect. 4.6) showed that naive training-time regularization is insufficient, as the gradient conflict between task adaptation and feature preservation actively degrades performance. This suggests that future mitigation strategies should avoid forcing these two objectives to compete within a single optimization loop. Promising directions include two-stage approaches that first train non-visual modules with frozen CLIP weights and then apply compression with targeted adaptation; *seen/unseen*-aware regularization using proxy text embeddings from classes outside the training set to maintain alignment without accessing *unseen* class labels; and framework-aware distillation that accounts for the structural dependencies introduced by modules such as PS and CAL, rather than applying generic feature matching. The weight restoration results also suggest that training the non-visual modules (CAL, prompts) with original CLIP weights while deploying with compressed weights could reduce the distribution mismatch observed in our experiments.

Beyond mitigation strategies, several directions for empirical extension would strengthen the generalizability of our findings. First, multi-run experiments with variance reporting would provide a formal statistical foundation for the observed asymmetric degradation, particularly for boundary cases where the *seen/unseen* gap is small. Second, validating the FT Paradox on additional OVSS frameworks such as SimBaseline [3], OVSeg [31], SAN [12], and CAT-Seg [13] would clarify whether the effect is specific to DeOP's tightly coupled PS/CAL design or a general property of CLIP-based OVSS systems. Third, extending the compression spectrum to more aggressive methods such as INT8/INT4 quantization, structured pruning, and higher LoRA ranks would characterize the trade-off between compression aggressiveness and zero-shot preservation, which is essential for practical deployment in resource-constrained environments.

Our findings also have implications beyond static deployment. The FT Paradox observed here where adapting CLIP to *seen* classes degrades *unseen* performance closely parallels the catastrophic forgetting phenomenon studied in continual and lifelong learning of foundation models [24,25]. In a continual OVSS setting, where new categories are incrementally introduced over time, naively FT the visual encoder on each new task would compound the asymmetric degradation we observe, potentially eroding the model's open-vocabulary capability across the learning trajectory. Our weight restoration result ($\alpha = 0$ being optimal) suggests that maintaining a snapshot of the original CLIP weights and selectively adapting only the non-visual modules could serve as a principled starting point for continual OVSS an avenue we leave for future investigation.

7 Conclusion

We present DeOP-Light, a systematic study of how CLIP visual encoder compression affects *seen* and *unseen* class performance in OVSS. Across two representative compression methods (FP16, LoRA) and two datasets (COCO-Stuff, Pascal VOC), we consistently observe an asymmetric degradation pat-

tern: FT the CLIP visual encoder recovers *seen* class performance but persistently degrades *unseen* class performance, with the degradation emerging within the first 1,000 iterations. Crucially, by comparing against an uncompressed FT baseline, we demonstrate that FT itself not compression is the primary driver of this degradation, with compression playing a secondary role. Post-hoc weight restoration partially recovers *unseen* performance, while training-time feature distillation fails due to gradient conflicts. Our key practical finding is that keeping the CLIP visual encoder frozen best preserves the zero-shot generalization that defines OVSS. We advocate for *seen/unseen*-separated evaluation as a standard practice when FT or compressing CLIP-based OVSS models.

Acknowledgments. This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP)-ICT Creative Consilience Program grant funded by the Korea government (MSIT) (IITP-2026-RS-2020-II201821), This work was supported by the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government under the Industrial Innovation Talent Growth Support Project (KIAT-2026-RS-2026-25532917).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML), pp. 8748–8763 (2021)
2. Ding, J., Xue, N., Xia, G.S., Dai, D.: Decoupling zero-shot semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11583–11592 (2022)
3. Xu, M., et al.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022. LNCS, vol. 13689, pp. 736–753. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19818-2_42
4. Han, C., Zhong, Y., Li, D., Han, K., Ma, L.: Open-vocabulary semantic segmentation with decoupled one-pass network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1086–1096 (2023)
5. Dosovitskiy, A., et al.: An image is worth 16x16 words: transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021)
6. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. In: International Conference on Learning Representations (ICLR) (2022)
7. Wortsman, M., et al.: Robust fine-tuning of zero-shot models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7959–7971 (2022)
8. Bucher, M., Vu, T.H., Cord, M., Pérez, P.: Zero-shot semantic segmentation. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 32 (2019)

9. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero- and few-label semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8256–8265 (2019)
10. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022. LNCS, vol. 13696, pp. 540–557. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20059-5_31
11. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML), pp. 4904–4916 (2021)
12. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2945–2954 (2023)
13. Cho, S., Shin, H., Hong, S., Arnab, A., Seo, P.H., Kim, S.: CAT-seg: cost aggregation for open-vocabulary semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4113–4123 (2024)
14. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Convolutions die hard: open-vocabulary segmentation with single frozen convolutional CLIP. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 36 (2023)
15. Hajimiri, S., Ben Ayed, I., Dolz, J.: Pay attention to your neighbours: training-free open-vocabulary semantic segmentation. In: Proceedings of the Winter Conference on Applications of Computer Vision (WACV), pp. 5061–5071 (2025)
16. Lan, M., Chen, C., Ke, Y., Wang, X., Feng, L., Zhang, W.: ClearCLIP: decomposing CLIP representations for dense vision-language inference. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024. LNCS, vol. 15105, pp. 143–160. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72970-6_9
17. Nagel, M., van Baalen, M., Blankevoort, T., Welling, M.: Data-free quantization through weight equalization and bias correction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1325–1334 (2019)
18. Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D.: OPTQ: accurate post-training quantization for generative pre-trained transformers. In: International Conference on Learning Representations (ICLR) (2023)
19. Jia, M., et al.: Visual prompt tuning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision - ECCV 2022. LNCS, vol. 13693, pp. 709–727. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19827-4_41
20. Wu, K., et al.: TinyCLIP: CLIP distillation via affinity mimicking and weight inheritance. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 21970–21980 (2023)
21. Yang, C., et al.: CLIP-KD: an empirical study of CLIP model distillation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 15952–15962 (2024)
22. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. Int. J. Comput. Vision **130**(9), 2337–2348 (2022)
23. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: DynamicViT: efficient vision transformers with dynamic token sparsification. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 13937–13949 (2021)

24. Zhai, Y., et al.: Investigating the catastrophic forgetting in multimodal large language model fine-tuning. In: Conference on Parsimony and Learning (CPAL), vol. 234, pp. 202–227. PMLR Proceedings of Machine Learning Research (2024)
25. Zheng, Z., Ma, M., Wang, K., Qin, Z., Yue, X., You, Y.: Preventing zero-shot transfer degradation in continual learning of vision-language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 19125–19136 (2023)
26. Bafghi, A., Bagwell, R., Ravichandran, C., Shrivastava, A., Raissi, A., M.: Fine tuning without catastrophic forgetting via selective low rank adaptation. arXiv preprint arXiv:2501.15377 (2025)
27. Cheng, B., Schwing, A.G., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 34, pp. 17864–17875 (2021)
28. Caesar, H., Uijlings, J., Ferrari, V.: COCO-stuff: thing and stuff classes in context. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1209–1218 (2018)
29. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL visual object classes (VOC) challenge. Int. J. Comput. Vision **88**(2), 303–338 (2010)
30. Wu, Y., Kirillov, A., Massa, F., Lo, W.Y., Girshick, R.: Detectron2 [Computer software]. Facebook AI Research (2019). https://github.com/facebookresearch/detectron2
31. Liang, F., et al.: Open-vocabulary semantic segmentation with mask-adapted CLIP. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7061–7070 (2023)