UNIEVO-RS: OMNI-PROMPT UNIFIED REMOTE SENSING SEGMENTATION WITH REPRESENTATIVE EXEMPLAR-DRIVEN PROTOTYPE EVOLUTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Prompt-driven vision-language models (VLMs) hold immense promise for accelerating dense remote sensing (RS) annotation, but static models suffer from severe performance degradation when deployed on novel scenes, unseen categories, or visually confusing backgrounds. Moreover, existing unified paradigms primarily rely on intra-image specific prompts, lacking flexible task routing to adapt to multi-intent operational workflows. In practical batch mapping, annotators typically refine a small set of representative samples before processing large datasets. Motivated by this practice, we propose UniEvo-RS, an omni-prompt unified RS segmentation framework equipped with representative exemplar-driven prototype evolution. First, we construct a multi-instruction prompt dataset that unifies text-driven and visual-driven prompts within a single architecture, establishing a dynamic task-routing mechanism for highly diverse RS annotation scenarios. Second, we introduce a representative feedback-driven, training-free prototype evolution mechanism. By contrasting manual annotations with initial predictions on exemplars, UniEvo-RS distills prediction errors into positive and negative prototypes. These prototypes enhance LLM query recall and suppress spatial background noise under a fixed-budget clustering memory. Extensive experiments show that UniEvo-RS unifies diverse prompting tasks, achieving state-of-the-art performance across most settings. Crucially, with minimal interaction on a few exemplars, it enables training-free, progressive accuracy enhancement on unseen categories during batch annotation.

1 INTRODUCTION

Semantic segmentation of remote sensing (RS) imagery plays a crucial role in Earth observation tasks such as urban planning and disaster assessment. However, due to the complexity of RS scenes, the highly specialized nature of geographical categories, and the strict requirement for high-fidelity delineation, segmentation models conventionally rely on time-consuming and expensive dense pixel-level annotations Waqas Zamir et al. (2019); Wang et al. (2023a). Recently, prompt-driven vision-language models (VLMs) Kirillov et al. (2023); Radford et al. (2021); Ravi et al. (2024); Zou et al. (2023a;b) have demonstrated remarkable zero-shot generalization capabilities in segmentation tasks. By incorporating textual instructions and visual prompts, such as points and bounding boxes, VLMs offer a novel paradigm for constructing initial annotations, enhancing interactive efficiency, and reducing annotation costs Lai et al. (2024); Zhang et al. (2024b); Ren et al. (2024).

Despite these advances, directly applying existing prompt-driven or unified segmentation models to complex RS annotation workflows remains challenging. First, current models primarily rely on intra-image-specific prompts and are typically designed for static, single-turn inference Kirillov et al. (2023); Ravi et al. (2024); Lai et al. (2024); Zhang et al. (2024b); Ren et al. (2024). Second, they lack flexible task-routing capabilities, making it difficult to switch among the diverse operational intents of human annotators. In practical batch-mapping scenarios, an annotator may need to alternate among referring segmentation, single-instance extraction, and cross-image multi-category segmentation based on a visual exemplar. Existing frameworks do not fully unify and route these heterogeneous instructions within a single architecture, limiting their applicability to comprehensive RS annotation workflows Wang et al. (2023c); Li et al. (2024); Zheng et al. (2025).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

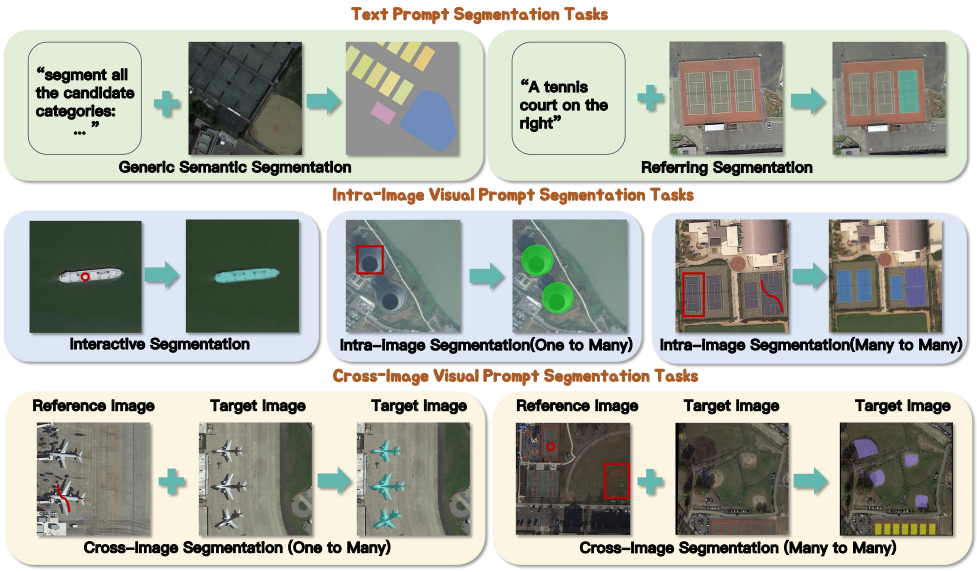


Figure 1: Overview of the segmentation tasks supported by UniEvo-RS. The unified architecture integrates text-prompted generic and referring segmentation with visual-prompted interactive, intra-image, and cross-image segmentation.

The limitations of static inference become particularly pronounced in practical batch annotation. Although VLMs possess open-vocabulary recognition capabilities, their performance often degrades when applied to RS scenes with novel textures, unseen categories, or confusing backgrounds, resulting in unreliable initial pre-annotations. In realistic batch mapping, human annotators typically inspect and refine a small set of representative samples, or exemplars, before applying the model to a larger image collection. While fine-tuning the model on these verified exemplars is a straightforward solution, repeatedly updating a large VLM introduces prohibitive computational overhead Lai et al. (2024); Zhang et al. (2024b); Ren et al. (2024); Jia et al. (2022); Hu et al. (2022); Chen et al. (2022). To avoid such repeated parameter updates, methods such as SegGPT Wang et al. (2023c) and OmniRIS Zheng et al. (2025) introduce cross-image prompting. However, these methods still process each target image independently and do not explicitly reuse the verified prediction errors obtained from representative exemplars. Directly caching all exemplar features would also introduce unnecessary memory and computational costs. Therefore, extracting reusable target and distractor priors from a small number of human-refined exemplars, while maintaining a fixed memory budget and avoiding parameter updates, remains an important challenge for RS batch annotation.

To address multi-instruction adaptability and batch annotation efficiency, we propose UniEvo-RS, an omni-prompt unified RS segmentation framework featuring representative exemplar-driven prototype evolution. First, we construct a multi-instruction dataset that unifies text and visual prompts. UniEvo-RS maps these heterogeneous prompts into a shared token space, establishing dynamic task routing for diverse annotation intents. Second, we introduce a training-free prototype evolution mechanism. By contrasting manual annotations with initial predictions on exemplars, it distills missed targets and false-positive backgrounds into positive and negative prototypes. Maintained via a fixed-budget clustering strategy, these prototypes enhance query recall and suppress spatial background noise during downstream cross-image inference without parameter updates. Evaluations across multiple benchmarks confirm that the unified architecture achieves state-of-the-art results, and the evolution mechanism yields training-free accuracy improvements on novel distributions. In summary, the main contributions of this paper are as follows:

(1) We propose UniEvo-RS, an omni-prompt unified RS segmentation framework, together with a comprehensive multi-instruction and multimodal-prompt RS segmentation dataset. It establishes a flexible task-routing mechanism that unifies text-prompted and visual-prompted tasks within a single architecture, supporting diverse RS segmentation and annotation intents.

(2) We propose a training-free prototype evolution mechanism for batch annotation. It distills verified errors on exemplars into positive and negative prototypes, which respectively enhance target recall and suppress background noise under a fixed-budget clustering memory.

(3) Extensive experiments demonstrate that UniEvo-RS achieves state-of-the-art results across most of the evaluated tasks. Furthermore, limited exemplar verification significantly enhances training-free cross-image pre-annotation for unseen categories and complex backgrounds.

2 Related Work

2.1 Unified Vision-Language Segmentation Models

Early unified segmentation models sought to accommodate heterogeneous vision tasks within a shared architecture through task-specific prompts or output interfaces Lu et al. (2022); Cheng et al. (2022); Kolesnikov et al. (2022). The SAM series Kirillov et al. (2023); Ravi et al. (2024) further established promptable segmentation using points, boxes, and masks, while SegGPT and Painter Wang et al. (2023c;b) cast visual tasks as in-context image transformation. X-Decoder and SEEM Zou et al. (2023a;b) subsequently aligned language and visual conditions in a shared semantic space. More recently, OmniRIS and its baseline OmniSegNet Zheng et al. (2025) extended referring segmentation to text and reference-image prompts, including one-to-many and many-to-many settings. Despite their broad task coverage, these models are primarily designed for natural images or intra-image interaction. Their prompt interfaces do not explicitly address the task routing and cross-scene appearance variation required by multi-intent RS annotation. In contrast, UniEvo-RS formulates text-driven and visual-driven RS segmentation as an omni-prompt multi-instruction learning problem, enabling diverse annotation intents to be dynamically routed within a shared architecture.

2.2 Interactive and Cross-Image RS Segmentation

Multimodal large language models have enabled segmentation systems to interpret increasingly complex language instructions Zhu et al. (2023); Bai et al. (2023). LISA, PSALM, and PixelLM Lai et al. (2024); Zhang et al. (2024b); Ren et al. (2024) connect language-model outputs with pixel-level decoders for referring, reasoning, and interactive segmentation. In remote sensing, GeoChat, EarthMarker, UniGeoSeg, SegEarth-R2, and SkyEyeGPT Kuckreja et al. (2024); Zhang et al. (2024a); Ni et al. (2025); Xin et al. (2025); Zhan et al. (2025) improve geospatial understanding through RS-specific instruction tuning and spatial grounding. However, most of these methods infer each target image from its current prompt and do not explicitly reuse human-verified prediction errors from representative samples. Although in-context models such as SegGPT and OmniRIS enable cross-image visual prompting, they merely treat historical exemplars as simple visual prompts, lacking a deep mining of the underlying information. In contrast, we propose a representative feedback-driven prototype evolution mechanism that deeply decouples verified errors on exemplars into reusable positive and negative prototypes.

2.3 Representative Feedback-Driven Prototype Adaptation

Test-time and continual adaptation methods update pretrained models using unlabeled target data or pseudo-labels Wang et al. (2021; 2022); Niu et al. (2023); Yuan et al. (2023). Although effective under certain distribution shifts, gradient-based adaptation can be computationally expensive for large vision-language models and sensitive to noisy predictions. Parameter-efficient alternatives adapt prompts or feature prototypes instead of the full network Rebuffi et al. (2017); Jia et al. (2022); Shu et al. (2022); Zhang et al. (2022), but they primarily emphasize positive class alignment or adaptation to individual samples. Meanwhile, conventional interactive segmentation typically treats human corrections as immediate prompts without converting verified errors into reusable cross-image knowledge. To bridge this gap, UniEvo-RS introduces a training-free approach. By explicitly storing verified feedback in a fixed-budget prototype memory, it transfers the verified knowledge to new target data through forward-pass inference.

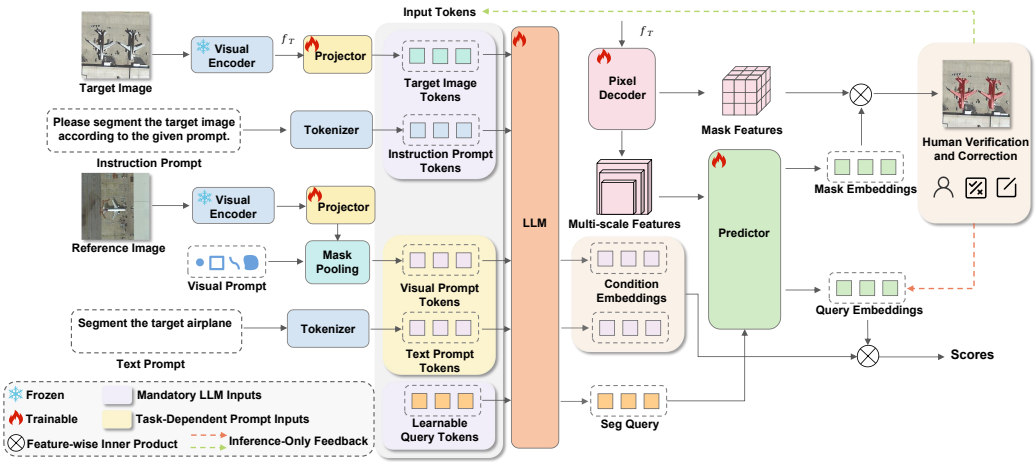


Figure 2: Overall architecture of UniEvo-RS.

3 METHODOLOGY

3.1 OVERALL WORKFLOW

To address multi-instruction adaptability and batch annotation efficiency, we propose UniEvo-RS. It is an omni-prompt unified RS segmentation framework equipped with representative exemplar-driven prototype evolution. Our method consists of two primary mechanisms.

First, we design a unified architecture to accommodate diverse annotation intents. As illustrated in Figure 2, UniEvo-RS processes multi-instruction inputs. Target images, reference features, language instructions, and visual prompts are mapped into a shared token space. An LLM contextualizes these inputs and dynamically routes them according to the current task instruction. A shared mask decoder then produces task-aware masks and matching scores. This formulation seamlessly integrates text-prompted and visual-prompted RS segmentation tasks.

Second, we introduce a training-free prototype evolution mechanism for practical batch annotation. UniEvo-RS compares the manual annotations of representative exemplars with their initial model predictions. It decomposes the prediction errors into missed target regions and incorrectly detected background distractors. Features from missed target regions are clustered into positive prototypes to guide the LLM toward recall-enhanced segmentation queries. False-positive background features form negative prototypes that drive spatial suppression in the mask predictor. A clustering strategy maintains the positive and negative prototype banks under a fixed memory budget. Consequently, the resulting exemplar feedback can be directly reused during downstream cross-image inference without updating any model parameters.

3.2 OMNI-PROMPT UNIFIED RS SEGMENTATION FRAMEWORK

UniEvo-RS projects heterogeneous instructions and multimodal prompts into a shared token space and routes them through a common mask-generation pathway. All tasks are formulated as:

$$\mathbf{Y} = \mathcal{F}(I^t, \mathcal{I}, \mathcal{P}^l, \mathcal{P}^v), \tag{1}$$

where I^t is the target image, $\mathcal{I}$ is the task instruction, $\mathcal{P}^l$ and $\mathcal{P}^v$ are optional language and visual prompts, and $\mathbf{Y}$ is the segmentation output.

Visual Feature Extraction via Image Encoder. A frozen pretrained vision transformer extracts multi-scale features $\mathbf{V}^t$ and $\mathbf{V}^p$ from the target image I^t and prompt image I^p, respectively. For intra-image prompting, $I^t = I^p$; otherwise, the two images are encoded with shared weights. These features are subsequently utilized for prompt encoding, mask decoding, and prototype extraction.

Multimodal Prompt Encoding. The multimodal prompt encoder maps heterogeneous prompt modalities into a unified token space. We adopt a shared token interface containing special tokens such as <image>, <region>, <seg>, <cls>, and <refer>. Text prompts are directly embedded as sequence tokens. For visual prompts, such as boxes, points, or scribbles, the encoded prompt-image feature map $\mathbf{V}^p$ is first mapped into the LLM embedding space by a trainable visual projector g_v. Mask pooling then samples and aggregates the projected features within each prompted region:

$$\mathbf{r}_k = \mathrm{Pool}(\mathrm{Sample}(g_v(\mathbf{V}^p), M_k^p)), \tag{2}$$

where M_k^p denotes the k-th prompt mask and $\mathbf{r}_k$ is its visual prompt token. A modality selector routes text tokens, visual tokens, or both into the LLM according to the task instruction. This design supports one-to-many and multi-region visual prompting under a unified formulation.

LLM Contextualization. The target-image tokens, selected prompt tokens, task instruction, and learnable segmentation queries are concatenated into a multimodal sequence $\mathbf{X}$ and fed into the LLM:

$$\mathbf{H} = \mathcal{F}_{\mathrm{llm}}(\mathbf{X}). \tag{3}$$

From $\mathbf{H}$, we extract contextualized segmentation queries at the <seg> positions and condition embeddings at the prompt-token positions, explicitly encoding task-aware reasoning and target conditions, respectively.

Unified Mask Decoder. The unified mask decoder consists of a pixel decoder and a shared mask predictor. The pixel decoder transforms the target-image backbone features into multi-scale decoder features $\mathbf{F}_{\mathrm{multi}}$ and high-resolution mask features $\mathbf{F}_{\mathrm{mask}}$. The mask predictor uses the contextualized segmentation queries $\tilde{\mathbf{q}}_q$ to attend to $\mathbf{F}_{\mathrm{multi}}$, producing decoded query embeddings:

$$\mathbf{z}_q = \mathcal{D}_{\mathrm{pred}}\left(\tilde{\mathbf{q}}_q, \mathbf{F}_{\mathrm{multi}}\right). \tag{4}$$

Each decoded query is mapped to a mask embedding by a mask MLP ϕ_m, and the query-wise candidate mask logits are computed as

$$\hat{\mathbf{M}}_q = \langle \phi_m(\mathbf{z}_q), \mathbf{F}_{\mathrm{mask}} \rangle. \tag{5}$$

For region-conditioned instructions, the matching score between the k-th condition embedding and the q-th decoded query is

$$s_{k,q} = \phi_r(\mathbf{z}_q)^\top \mathbf{r}'_k, \tag{6}$$

where ϕ_r projects decoded queries into the shared region-query embedding space and $\mathbf{r}'_k$ is the projected condition embedding. This design keeps the computationally intensive mask-generation pathway shared across tasks.

Unified Multi-Task Optimization. During training, we adopt a batch-wise task alternation strategy, where each mini-batch is sampled from one task and tasks rotate across batches. All tasks share the mask loss $\mathcal{L}_{\mathrm{mask}}$ and Dice loss $\mathcal{L}_{\mathrm{dice}}$ for mask-quality supervision. Task-specific matching losses $\mathcal{L}_{\mathrm{prompt}}$, such as cross-entropy losses for semantic categories or referring expressions, are dynamically activated according to the instruction type:

$$\mathcal{L} = \lambda_{\mathrm{mask}}\mathcal{L}_{\mathrm{mask}} + \lambda_{\mathrm{dice}}\mathcal{L}_{\mathrm{dice}} + \lambda_{\mathrm{prompt}}\mathcal{L}_{\mathrm{prompt}}. \tag{7}$$

At inference time, a lightweight task-specific post-processing step parses the unified predictions. For referring segmentation, the mask with the highest matching score is selected. For interactive and generic segmentation, masks exceeding a predefined confidence threshold are retained, adapting the unified output to different RS task requirements.

3.3 Representative Exemplar-Driven Prototype Evolution

Although the unified architecture supports multiple annotation intents, its initial predictions may still degrade on unseen target appearances and confusing backgrounds in novel RS image collections. In practical batch annotation, an annotator can first verify a small set of representative exemplars before applying the model to the remaining images. We therefore introduce representative exemplar-driven, training-free prototype evolution to reuse this verified feedback without parameter updates.

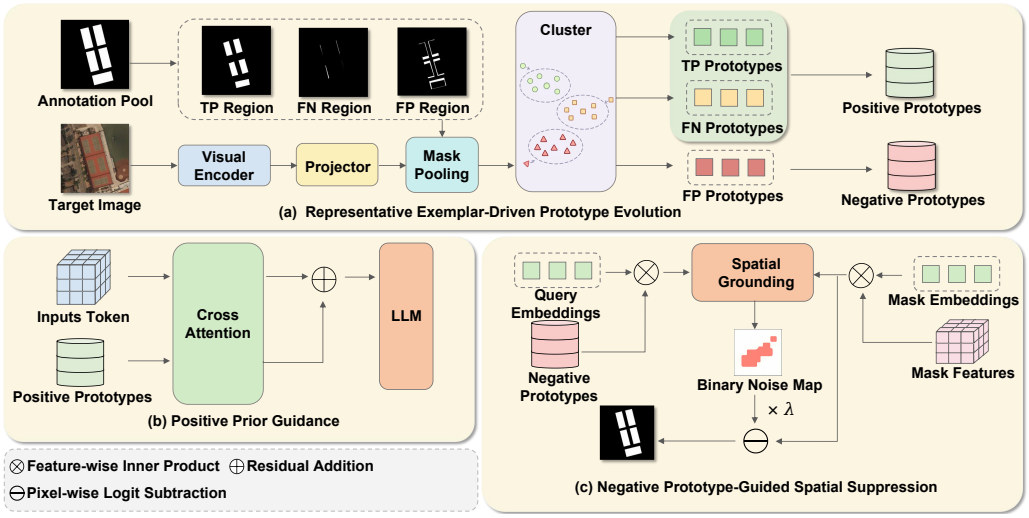


Figure 3: Representative exemplar-driven prototype evolution. Human-verified TP/FN regions form positive target memory, whereas FP regions form negative background memory. K-Means maintains fixed budgets for subsequent cross-image inference without parameter updates.

Prototype Memory Construction. For each representative exemplar, the model first predicts $\hat{\mathbf{Y}}$, after which an annotator verifies or corrects it to obtain $\mathbf{Y}_v$. Their comparison decomposes the result into true positives (TP), false negatives (FN), and false positives (FP). To simulate this interactive verification, ground-truth annotations serve as a surrogate for human feedback, revealed strictly after the initial prediction. We extract region-level prototype tokens from the projected target-image feature map $\bar{\mathbf{V}} = g_v(\mathbf{V})$, using the same visual projector g_v introduced above, where $\mathbf{V}$ denotes the target-image feature map produced by the image encoder. Specifically, a prototype $\mathbf{r} \in \mathbb{R}^d$ is generated through mask pooling:

$$\mathbf{r} = \mathrm{Pool}\big(\mathrm{Sample}(\bar{\mathbf{V}}, M)\big)\,. \tag{8}$$

where M denotes a decomposed region mask, such as M_{FN}. These tokens are stored in three isolated partitions, $\mathcal{B}_{\mathrm{TP}}$, $\mathcal{B}_{\mathrm{FN}}$, and $\mathcal{B}_{\mathrm{FP}}$. TP and FN prototypes serve as positive memory for target guidance, while FP prototypes serve as negative memory for background suppression.

Representative Exemplar-Driven Prototype Evolution. To keep the prototype memory exposed to the model within a fixed budget, we compress newly verified evidence and the previously retained centers through K-Means clustering. Let $\mathcal{R}_c^{(t)}$ denote the newly extracted region tokens of type $c \in \{\mathrm{TP}, \mathrm{FN}, \mathrm{FP}\}$ at representative step t. A temporary update pool is constructed as

$$\mathcal{U}_c^{(t)} = \mathcal{M}_c^{(t-1)} \cup \mathcal{R}_c^{(t)}. \tag{9}$$

K-Means is applied independently to the TP, FN, and FP update pools. Whenever newly verified tokens are added, the retained centers are recomputed from the temporary pools, allowing the memory to incorporate feedback from successively verified representative exemplars. The positive and negative memories at step t are defined as

$$\mathcal{M}_t^{+} = \left[\mathcal{C}_m\left(\mathcal{U}_{\mathrm{TP}}^{(t)}\right); \mathcal{C}_n\left(\mathcal{U}_{\mathrm{FN}}^{(t)}\right)\right], \tag{10}$$

$$\mathcal{M}_t^{-} = \mathcal{C}_r\left(\mathcal{U}_{\mathrm{FP}}^{(t)}\right), \tag{11}$$

where $\mathcal{C}_k(\cdot)$ extracts at most k cluster centers from the corresponding temporary pool. After clustering, only the centers are retained and the temporary pools are discarded. Thus, memory always contains at most $m + n$ positive prototypes and r negative prototypes. Retaining both TP and FN centers preserves stable target characteristics and difficult patterns associated with missed targets. The hyperparameters m and n control the allocation between stable and difficult target priors.

Positive Prior Guidance. Before LLM contextualization, positive memory $\mathcal{M}^+$ is injected into the current multimodal sequence $\mathbf{X}$. We compute a memory-enhanced representation $\tilde{\mathbf{X}}$ through residual cross-attention:

$$\mathbf{A}^+ = \mathrm{Softmax}\left(\frac{\mathbf{X}(\mathcal{M}^+)^\top}{\sqrt{d}}\right), \tag{12}$$

$$\tilde{\mathbf{X}} = \mathbf{X} + \alpha \mathbf{A}^+ \mathcal{M}^+, \tag{13}$$

where α modulates the injection strength. This operation enriches the input tokens with target characteristics and missed patterns extracted from verified representative exemplars, guiding the LLM queries toward relevant target regions in subsequent images.

Negative Prototype-Guided Spatial Suppression. Negative memory $\mathcal{M}^-$ is employed after query decoding to suppress spatial patterns associated with false-positive backgrounds identified from verified representative exemplars. Rather than directly generating masks, each negative prototype $\mathbf{p}_k^-$ and decoded query embedding $\mathbf{z}_q$ are projected into a shared region-query space. Their negative matching affinity is computed as

$$a_{kq}^- = \sigma\left(\frac{\left\langle \phi_n(\mathbf{p}_k^-), \phi_r(\mathbf{z}_q)\right\rangle}{\sqrt{d_r}}\right), \tag{14}$$

where ϕ_n and ϕ_r are projection functions, d_r is the matching dimension, $\sigma(\cdot)$ is the sigmoid function, and k and q index the negative prototypes and decoded queries, respectively.

For each decoded query, we retain its strongest affinity with the negative memory:

$$w_q^- = \max_k a_{kq}^-. \tag{15}$$

The resulting confidence w_q^- measures how strongly the q-th query resembles any verified false-positive prototype.

Let $M_q(u, v)$ denote the candidate mask logit predicted by query q at spatial location (u, v). We spatially ground the query-level negative confidence onto the corresponding candidate mask and aggregate the responses over all decoded queries:

$$S^-(u, v) = \max_q \left[w_q^- \, \sigma\left(M_q(u, v)\right)\right], \tag{16}$$

where $S^-(u, v)$ denotes the negative spatial confidence map. This operation ensures that a candidate mask contributes strongly to the negative spatial response only when its corresponding query is highly similar to at least one negative prototype.

A binary noise map is then obtained through confidence thresholding:

$$N(u, v) = \mathbb{I}\left(S^-(u, v) > \tau\right), \tag{17}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function and τ is the confidence threshold.

Finally, the candidate mask logits are refined through explicit spatial subtraction:

$$\widetilde{M}_q(u, v) = M_q(u, v) - \lambda N(u, v), \tag{18}$$

where λ controls the spatial suppression strength. The same noise map is applied to all candidate masks, suppressing regions jointly supported by verified false-positive prototypes and their matched queries. This operation modifies only the spatial mask logits while leaving condition-query matching scores unchanged. Both guidance paths are used only during representative exemplar-driven batch inference, with all model parameters fixed.

Table 1: Comparison on text- and visual-prompted segmentation tasks. The upper and lower blocks report zero-shot and fine-tuned results, respectively. For text-prompted tasks, PQ is used for generic segmentation, while gIoU and cIoU are used for referring segmentation. For visual-prompted tasks, gIoU and cIoU are reported for interactive, intra-image 1:N, and cross-image 1:N segmentation. A dash indicates that the corresponding task is not supported. Best and second-best results within each block are shown in bold and underlined, respectively.

Method	Text-prompted			Visual-prompted					
	Generic	Referring		Interactive		Intra 1:N		Cross 1:N	
	PQ	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU	gIoU	cIoU
Zero-shot									
RemoteSAM	<u>25.67</u>	**66.67**	<u>42.81</u>	–	–	–	–	–	–
SEEM	**26.57**	20.35	18.36	15.63	3.80	–	–	–	–
PSALM	12.55	24.75	20.84	17.10	9.36	–	–	–	–
X-SAM	10.56	32.71	22.86	18.99	6.60	–	–	–	–
UniGeoSeg	–	<u>61.08</u>	**65.70**	30.41	<u>38.96</u>	–	–	–	–
SAM3	–	42.57	28.65	**61.60**	**56.55**	–	–	**38.16**	**37.71**
YOLOE	–	–	–	<u>43.10</u>	35.40	**45.46**	**49.40**	13.75	13.86
DINOv	19.16	–	–	36.49	29.78	<u>38.66</u>	<u>31.55</u>	30.35	20.43
COSINE	3.29	19.93	13.07	10.81	12.35	17.04	16.06	<u>32.27</u>	<u>31.21</u>
Fine-tuned on our dataset									
SegEarth-R2	–	34.99	37.94	–	–	–	–	–	–
RemoteSAM	10.74	**65.75**	<u>78.96</u>	–	–	–	–	–	–
COSINE	2.85	63.31	76.34	–	–	–	–	–	–
UniGeoSeg	–	<u>63.72</u>	**80.06**	54.23	59.34	–	–	–	–
SEEM	27.96	46.59	57.47	24.75	12.99	–	–	–	–
SAM3	–	47.10	49.37	**67.57**	<u>68.54</u>	–	–	–	–
OmniSegNet	–	34.09	48.08	–	–	48.10	47.50	41.58	38.68
YOLOE	–	–	–	47.27	41.57	45.00	41.94	12.58	9.78
DINOv	<u>42.81</u>	–	–	36.47	35.37	<u>61.30</u>	<u>58.98</u>	**74.97**	<u>74.78</u>
UniEvo-RS (Ours)	**62.60**	55.72	69.83	<u>64.71</u>	**78.75**	**66.37**	**65.53**	<u>74.49</u>	**79.25**

4 EXPERIMENTS

4.1 DATASETS AND METRICS

We construct a multi-instruction RS segmentation dataset from four public benchmarks: SIOR (Wang et al., 2023a), iSAID Waqas Zamir et al. (2019), NWPU-Refer Yang et al. (2025), and RRSIS-D Liu et al. (2024). This dataset encompasses text-prompted generic and referring segmentation, as well as visual-prompted interactive, intra-image one-to-many, and cross-image one-to-many segmentation. Dataset statistics, task construction, and prompt generation protocols are provided in the appendix.

We report PQ for generic segmentation, and gIoU and cIoU for referring, interactive, and one-to-many visual prompt segmentation. For the prototype-memory ablations, we additionally report mIoU to jointly measure foreground recovery and background suppression.

4.2 COMPARISON RESULTS

Table 1 benchmarks UniEvo-RS against general-purpose segmentation models, including SEEM (Zou et al., 2023b), PSALM (Zhang et al., 2024b), SAM3 (Carion et al., 2025), DINOv (Li et al., 2024), YOLOE (Wang et al., 2025), and X-SAM (Wang et al., 2026), as well as RS-specific methods such as UniGeoSeg (Ni et al., 2025), RemoteSAM (Yao et al., 2025), and SegEarth-R2 (Xin et al., 2025). We further include OmniSegNet (Zheng et al., 2025) for the one-to-many settings. The

Table 2: Ablation results of fixed-budget memory fusion on cross-image one-to-many segmentation.

Exp.	Method	gIoU	mIoU
1	Baseline	36.43	54.96
2	Previous-mask transfer	36.66	56.87
3	Positive memory concatenation	36.47	54.92
4	Positive prior guidance	**41.36**	57.39
5	Positive and negative prior guidance	41.24	**59.50**

upper blocks evaluate released checkpoints, whereas the lower blocks report applicable baselines fine-tuned on our dataset's training splits. All methods are evaluated under identical test splits and metrics.

As shown, fine-tuning generally improves applicable baselines. Crucially, UniEvo-RS supports all five settings and achieves state-of-the-art results on five of the nine metrics, including generic segmentation, intra-image one-to-many segmentation, and cIoU for interactive and cross-image one-to-many segmentation. While specialized RS models edge out UniEvo-RS in referring segmentation due to their heavily optimized task-specific language grounding, our framework remains best or highly competitive on most metrics, providing complete task coverage without sacrificing multi-task versatility.

Taken together, these results highlight the distinction between task specialization and unified task coverage. Existing methods typically perform strongly on the prompt setting for which they are designed, but many do not support other annotation intents. UniEvo-RS instead maintains a shared prompt-to-mask pathway across text and visual prompts, enabling different segmentation settings to be handled without switching architectures.

4.3 Ablation Study

We evaluate the proposed training-free prototype evolution mechanism in a practical batch annotation setup using the Vehicle category from SIOR. Specifically, prototypes are extracted from human-verified feedback on a small set of representative exemplars and subsequently applied to the remaining target batch. As shown in Table 2, transferring the previous mask or directly concatenating historical features provides limited benefit, indicating that feedback reuse alone is insufficient.

The comparison further suggests that the key is not simply retaining historical information, but converting verified feedback into structured guidance. Previous-mask transfer and feature concatenation reuse historical information without explicitly separating target evidence from recurring background distractors. In contrast, prototype evolution decouples these two types of feedback and introduces them through complementary guidance paths for more targeted refinement.

Positive prior guidance improves target recognition, while incorporating negative guidance further improves mIoU with comparable gIoU. This behavior is consistent with the intended complementary roles of positive prototypes in recovering targets and negative prototypes in suppressing recurring false positives. Additional analyses of prototype allocation, hyperparameter sensitivity, fixed-budget efficiency, and memory compression are provided in the appendix.

5 Conclusion

In this paper, we present UniEvo-RS, an omni-prompt unified framework supporting five text-prompted and visual-prompted RS segmentation tasks, backed by a comprehensive multi-instruction dataset. Tailored for RS batch annotation, UniEvo-RS introduces a training-free prototype evolution mechanism driven by representative feedback. By contrasting manual annotations with initial predictions on a few exemplars, it decomposes errors into missed targets and false-positive distractors, forming fixed-budget positive and negative prototype memories. These memories guide LLM queries and suppress recurring background noise during cross-image inference without updating model parameters. Experiments demonstrate unified performance and confirm that limited human verification on representative exemplars substantially enhances downstream batch pre-annotation.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing and presentation, including improving readability and LaTeX formatting of tables and document structure. We did not use generative AI tools to generate experimental data or results, or to formulate the methodology, experimental design, or scientific conclusions of this work. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and J Qwen-VL Zhou. A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 6:3, 2023.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

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*, pp. 1290–1299, 2022.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

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 (ECCV)*, pp. 709–727, 2022.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4015–4026, 2023.

Alexander Kolesnikov, André Susano Pinto, Lucas Beyer, Xiaohua Zhai, Jeremiah Harmsen, and Neil Houlsby. Uvim: A unified modeling approach for vision with learned guiding codes. *Advances in Neural Information Processing Systems*, 35:26295–26308, 2022.

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 27831–27840, 2024.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9579–9589, 2024.

Feng Li, Qing Jiang, Hao Zhang, Tianhe Ren, Shilong Liu, Xueyan Zou, Huaizhe Xu, Hongyang Li, Jianwei Yang, Chunyuan Li, et al. Visual in-context prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12861–12871, 2024.

Sihan Liu, Yiwei Ma, Xiaoqing Zhang, Haowei Wang, Jiayi Ji, Xiaoshuai Sun, and Rongrong Ji. Rotated multi-scale interaction network for referring remote sensing image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26658–26668, 2024.

Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. *arXiv preprint arXiv:2206.08916*, 2022.

Shuo Ni, Di Wang, He Chen, Haonan Guo, Ning Zhang, and Jing Zhang. Unigeoseg: Towards unified open-world segmentation for geospatial scenes. *arXiv preprint arXiv:2511.23332*, 2025.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations (ICLR)*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2001–2010, 2017.

Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26374–26383, 2024.

Yang Shu, Xingjian Guo, Jialong Wu, Ximei Wang, Jianmin Wang, and Mingsheng Long. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Ao Wang, Lihao Liu, Hui Chen, Zijia Lin, Jungong Han, and Guiguang Ding. Yoloe: Real-time seeing anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24591–24602, 2025.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*, 2021.

Di Wang, Jing Zhang, Bo Du, Minqiang Xu, Lin Liu, Dacheng Tao, and Liangpei Zhang. Samrs: Scaling-up remote sensing segmentation dataset with segment anything model. *Advances in Neural Information Processing Systems*, 36:8815–8827, 2023a.

Hao Wang, Limeng Qiao, Zequn Jie, Zhijian Huang, Chengjian Feng, Qingfang Zheng, Lin Ma, Xiangyuan Lan, and Xiaodan Liang. X-sam: From segment anything to any segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 26187–26196, 2026.

Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7201–7211, 2022.

Xinlong Wang, Wen Wang, Yue Cao, Chunhua Shen, and Tiejun Huang. Images speak in images: A generalist painter for in-context visual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6830–6839, 2023b.

Xinlong Wang, Xiaosong Zhang, Yue Cao, Wen Wang, Chunhua Shen, and Tiejun Huang. Seggpt: Towards segmenting everything in context. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1130–1140, 2023c.

Syed Waqas Zamir, Aditya Arora, Akshita Gupta, Salman Khan, Guolei Sun, Fahad Shahbaz Khan, Fan Zhu, Ling Shao, Gui-Song Xia, and Xiang Bai. isaid: A large-scale dataset for instance segmentation in aerial images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 28–37, 2019.

Zepeng Xin, Kaiyu Li, Luodi Chen, Wanchen Li, Yuchen Xiao, Hui Qiao, Weizhan Zhang, Deyu Meng, and Xiangyong Cao. Segearth-r2: Towards comprehensive language-guided segmentation for remote sensing images. *arXiv preprint arXiv:2512.20013*, 2025.

Zhigang Yang, Huiguang Yao, Linmao Tian, Xuezhi Zhao, Qiang Li, and Qi Wang. A large-scale referring remote sensing image segmentation dataset and benchmark. *arXiv preprint arXiv:2506.03583*, 2025.

Liang Yao, Fan Liu, Delong Chen, Chuanyi Zhang, Yijun Wang, Ziyun Chen, Wei Xu, Shimin Di, and Yuhui Zheng. Remotesam: Towards segment anything for earth observation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 3027–3036, 2025.

Longhui Yuan et al. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

Yang Zhan, Zhitong Xiong, and Yuan Yuan. Skyeyegpt: Unifying remote sensing vision-language tasks via instruction tuning with large language model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 221:64–77, 2025.

Renrui Zhang, Rongyao Wei, Jianwei Fang, et al. Tip-adapter: Training-free image-text model adaptation. In *European Conference on Computer Vision (ECCV)*, 2022.

Wei Zhang, Miaoxin Cai, Tong Zhang, Yin Zhuang, Jun Li, and Xuerui Mao. Earthmarker: A visual prompting multimodal large language model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 63:1–19, 2024a.

Zheng Zhang, Yeyao Ma, Enming Zhang, and Xiang Bai. Psalm: Pixelwise segmentation with large multi-modal model. In *European Conference on Computer Vision*, pp. 74–91. Springer, 2024b.

Qiancheng Zheng, Yunhang Shen, Gen Luo, Baiyang Song, Xing Sun, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Omni-referring image segmentation. *arXiv preprint arXiv:2512.06862*, 2025.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

Xueyan Zou, Zi-Yi Dou, Jianwei Yang, Zhe Gan, Linjie Li, Chunyuan Li, Xiyang Dai, Harkirat Behl, Jianfeng Wang, Lu Yuan, et al. Generalized decoding for pixel, image, and language. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15116–15127, 2023a.

Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *Advances in neural information processing systems*, 36:19769–19782, 2023b.

A Appendix

A.1 Omni-Prompt Unified Remote Sensing Segmentation

A.1.1 Omni-Prompt Multi-Instruction Dataset

To support diverse remote sensing annotation intents within a single architecture, we reformat task-specific splits derived from SIOR, iSAID, NWPU-Refer, and RRSIS-D into a unified instruction-following segmentation dataset. Each constructed sample contains a target image, a task instruction, an optional textual or visual prompt, and the corresponding segmentation target. For cross-image one-to-many segmentation, the visual prompt is provided by a region in a separate source image.

The task instruction is present for all samples and specifies the current annotation intent. The optional prompt determines whether additional textual prompt tokens or visual prompt tokens are activated. This unified sample representation allows heterogeneous segmentation tasks to share the same multimodal token space, LLM contextualization pathway, and mask-generation interface while being dynamically routed according to the task instruction.

As summarized in Table 3, the reconstructed dataset contains two text-prompted settings and three visual-prompted settings. The reported train and test statistics denote the numbers of constructed prompt–target pairs rather than unique images.

Table 3: Composition of the reconstructed omni-prompt remote sensing segmentation dataset. Train and test statistics denote the numbers of constructed prompt–target pairs rather than unique images.

Prompt Modality	**Task Setting**	**Prompt Form**	**Source Dataset(s)**	**Train**	**Test**
Text	Generic	Semantic task instruction	SIOR	6,034	1,492
	Referring	Natural-language expression	NWPU-Refer, RRSIS-D	23,698	2,639
Visual	Interactive	Point, box, scribble, mask	SIOR, iSAID	44,374	1,757
	Intra-image 1:N	Point, box, scribble, mask	SIOR	67,269	19,011
	Cross-image 1:N	Point, box, scribble, mask	SIOR	67,269	19,291

A.1.2 Text-Prompted Task Construction

Generic semantic segmentation. We reorganize SIOR into instruction–mask pairs. The language instruction specifies the semantic segmentation intent, while the dense semantic annotations of the target image provide pixel-level supervision. This setting evaluates whether the shared architecture can perform category-level dense prediction from a task-level language instruction without requiring an additional spatial prompt.

Referring segmentation. We use NWPU-Refer and RRSIS-D, which associate natural-language referring expressions with spatially corresponding object masks. Each expression is used as the textual prompt, and the associated mask is used as the segmentation target. This setting evaluates whether the model can jointly resolve linguistic attributes, object identity, and spatial references in complex remote sensing scenes.

A.1.3 Visual-Prompted Task Construction

Interactive segmentation. We construct visual prompts from annotated foreground regions in SIOR and iSAID. Points, bounding boxes, scribbles, and region masks are used as alternative prompt forms, while the mask of the prompted instance provides supervision. These prompt types expose the model to different levels of spatial specificity and support conventional single-instance interaction.

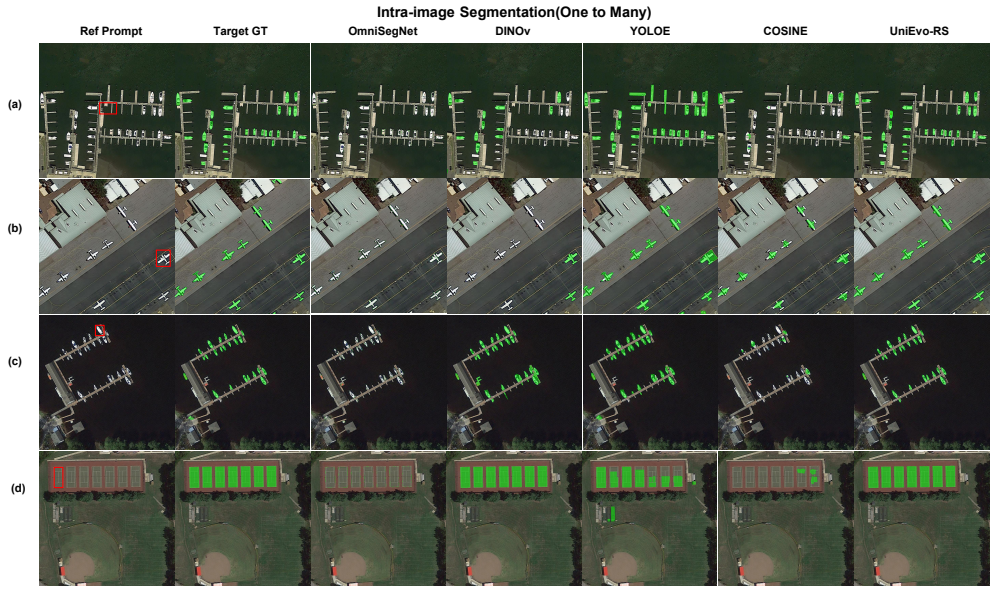


Figure 4: Qualitative comparisons on intra-image one-to-many segmentation. The reference prompt and target instances are located in the same image. The red bounding box specifies one reference instance, and the model is required to segment all instances belonging to the same semantic category. Green overlays denote the target ground-truth or predicted masks.

Intra-image one-to-many segmentation. Given a visual prompt sampled from one instance in the target image, the model is required to segment all instances belonging to the same semantic category in that image. This task extends conventional single-instance interaction to category-level extraction within a single scene.

Cross-image one-to-many segmentation. A visual prompt is sampled from a source image and paired with a different target image containing instances of the same semantic category. The segmentation target contains all matching instances in the target image. This task evaluates whether region-level semantics can be transferred across images despite variations in appearance, scale, orientation, spatial layout, object density, and background context.

A.1.4 Qualitative Results for One-to-Many Segmentation

We provide additional qualitative comparisons for the two one-to-many visual-prompted settings. In each example, the region enclosed by the red bounding box serves as the reference prompt. The model is required to identify and segment all instances sharing the prompted semantic category. Green overlays denote the target ground-truth or predicted masks.

Figure 4 illustrates the intra-image one-to-many setting. In the shown examples, OmniSegNet fails to propagate the prompted category semantics to several corresponding instances. DINOv and YOLOE recover more targets but still miss some small, densely distributed, or appearance-shifted instances. COSINE occasionally produces incomplete masks or responses on visually similar background structures.

Across the displayed cases, UniEvo-RS produces more complete target coverage and fewer fragmented responses. The improvement is particularly visible for densely distributed aircraft and ships, as well as repeated structures such as sports courts, where instances exhibit substantial variation in scale, orientation, and surrounding context.

Figure 5 further evaluates semantic transfer across different images. Compared with intra-image prompting, this setting is more challenging because the reference and target instances may differ substantially in appearance, scale, orientation, density, spatial arrangement, and imaging context.

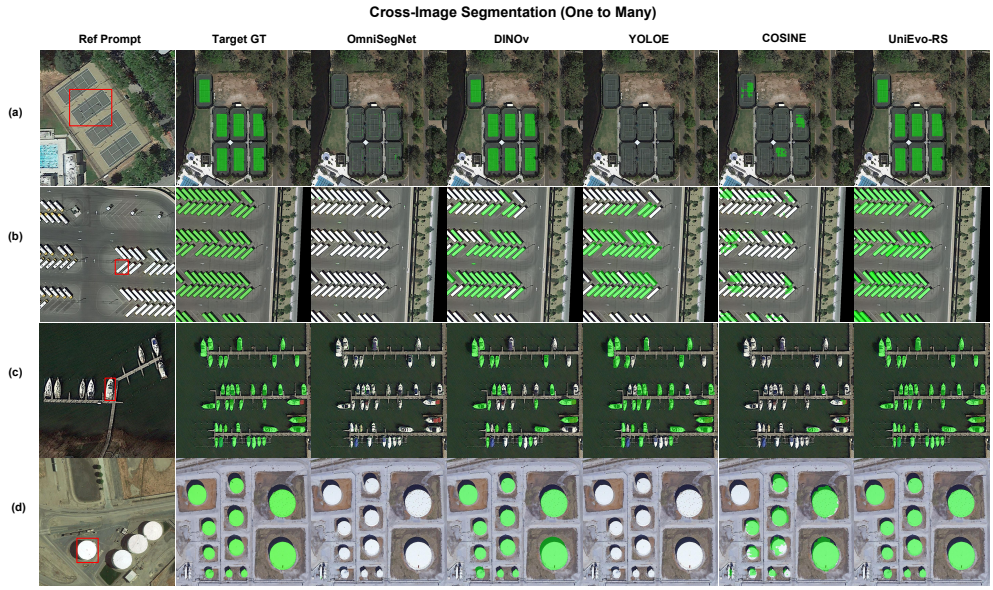


Figure 5: Qualitative comparisons on cross-image one-to-many segmentation. A reference instance enclosed by the red bounding box is selected from a source image, while all semantically corresponding instances must be segmented in a different target image. Green overlays denote the target ground-truth or predicted masks.

In the displayed examples, OmniSegNet identifies only a subset of the corresponding targets. DINOv and YOLOE recover more matching instances but still exhibit missed objects or incomplete masks. COSINE produces meaningful responses in several cases but also introduces fragmented predictions and false-positive regions.

UniEvo-RS more consistently transfers the region-level semantics of the reference prompt to the target image. It recovers a larger proportion of matching instances for densely parked vehicles, differently oriented ships, multi-scale storage tanks, and repeated sports-field structures. These results support the effectiveness of the unified visual-prompt interface for category-level correspondence rather than local appearance matching alone.

A.1.5 Qualitative Capability Study for Many-to-Many Segmentation

The unified visual-prompt interface also supports multi-region prompting. In the many-to-many setting, multiple reference regions are provided simultaneously, potentially corresponding to different semantic categories, and the model is required to segment all target regions associated with the active prompts.

As shown in Figure 6, UniEvo-RS can jointly process multiple visual prompts and recover multiple corresponding target regions within a single inference process. In the intra-image examples, the model segments regions associated with several active prompts in the same scene. In the cross-image examples, it transfers the semantics of multiple prompted regions to a different image despite changes in appearance, scale, layout, and background context.These examples indicate that the omni-prompt design is not structurally restricted to one-to-many interaction.

A.2 Representative Exemplar-Driven Prototype Evolution

A.2.1 Sequential Batch-Annotation Protocol

The prototype evolution mechanism is evaluated under a sequential human-in-the-loop batch-annotation protocol. Representative exemplars refer to the small subset of samples inspected and corrected by the annotator before the remaining image collection is processed. UniEvo-RS does not require a learned exemplar-selection module.

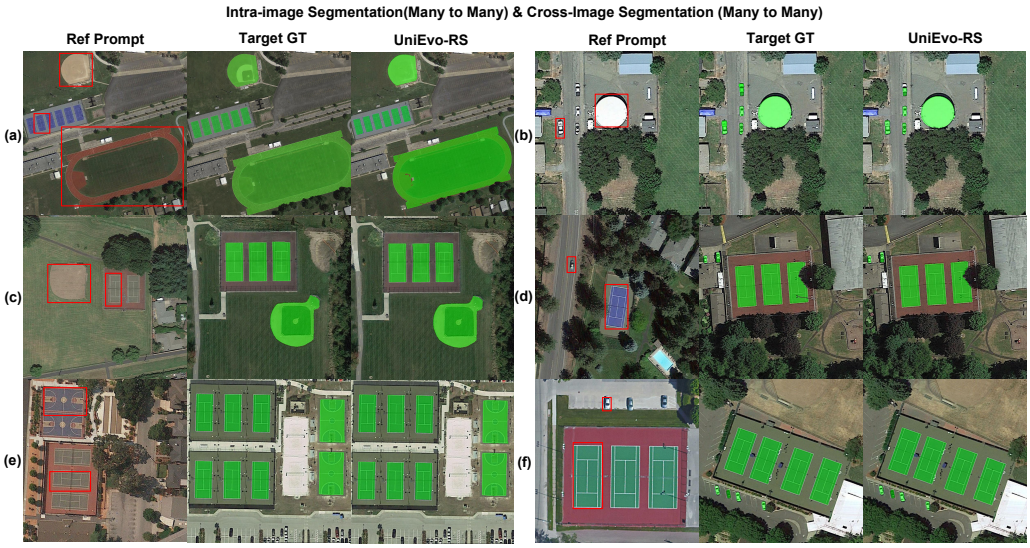


Figure 6: Qualitative demonstrations of many-to-many visual-prompted segmentation with UniEvo-RS. Panels (a) and (b) show intra-image many-to-many segmentation, where multiple reference prompts and their corresponding targets appear in the same image. Panels (c)–(f) show cross-image many-to-many segmentation, where multiple prompts are selected from a source image and their semantically corresponding regions are segmented in a different target image. Red boxes denote the prompted reference regions, while green overlays visualize the union of the corresponding target masks.

For the t-th image in an annotation sequence, prediction uses only the prototype memory accumulated from previously verified samples:

$$\hat{Y}_t = \mathcal{F}\left(I_t, \mathcal{I}_t, P_t; \mathcal{M}_{t-1}\right), \tag{19}$$

where I_t is the current target image, $\mathcal{I}_t$ is the task instruction, P_t denotes the active textual or visual prompt, and $\mathcal{M}_{t-1}$ contains only historical prototype memory.

The annotation or corrected mask Y_t^v is revealed only after $\hat{Y}_t$ has been produced. The comparison between $\hat{Y}_t$ and Y_t^v yields TP, FN, and FP regions, which are then used to update the memory:

$$\mathcal{M}_t = \text{Update}\left(\mathcal{M}_{t-1}, \text{Decompose}\left(\hat{Y}_t, Y_t^v\right)\right). \tag{20}$$

Therefore, the annotation of the current target image never contributes to its own prediction. In our experiments, human verification is simulated by revealing the dataset annotation only after the current prediction has been generated. This sequential protocol prevents target-mask leakage and matches the intended practical workflow.

TP and FN features form positive target memory, while FP features form negative background memory. K-Means independently compresses the three historical banks into a fixed number of prototype centers. Only these centers are retained after each update; the complete annotation history is not cached or injected into subsequent images.

In the following figures, *Static Pred* denotes inference using the current task instruction and prompt without historical prototype memory. *Evolved Pred* uses the same current input together with the memory accumulated from previously verified representative exemplars.

A.2.2 Generalization to Unseen Instances of Seen Categories

In this setting, the evaluated semantic categories are included in task-specific model training, while the target images and object instances are unseen. The representative exemplars used to construct the historical memory are separated from the target images on which the qualitative results are reported.

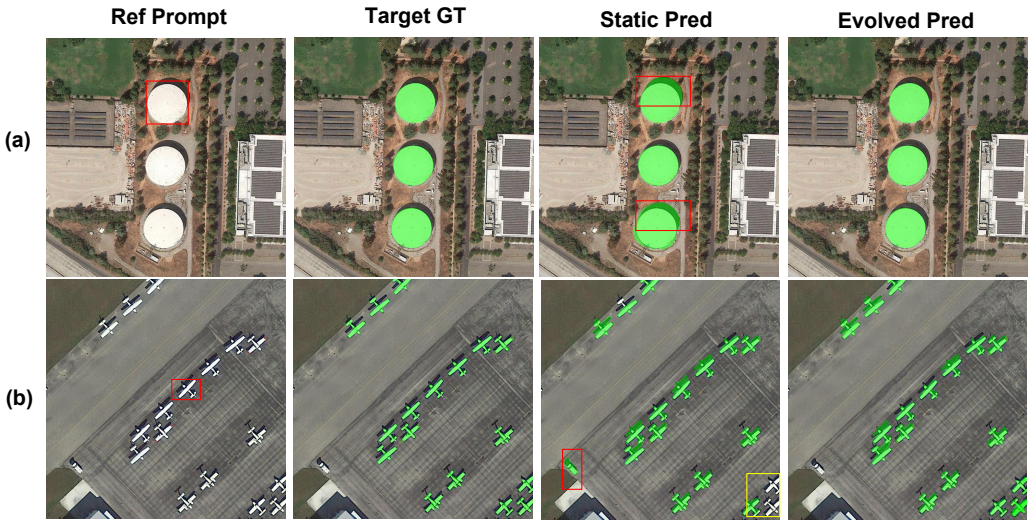


Figure 7: Qualitative comparison of representative exemplar-driven prototype evolution on unseen instances of seen categories. From left to right, the columns show the current reference prompt, target ground truth, static prediction without historical memory, and prediction with evolved prototype memory. The displayed categories are Storage Tank and Aircraft. Green overlays denote target or predicted masks, while colored boxes highlight representative errors in the static predictions.In the static predictions, red boxes highlight falsely predicted target pixels, corresponding to false-positive regions, whereas yellow boxes indicate missed target pixels, corresponding to false-negative regions.

As shown in Figure 7, static inference already captures the coarse semantics of Storage Tank and Aircraft but still produces incomplete target boundaries, missed regions, and responses to visually confusing structures.

After representative feedback is accumulated, the evolved predictions provide more complete storage-tank masks and more consistent aircraft coverage. TP prototypes preserve stable target characteristics, FN prototypes provide additional evidence for difficult or previously missed target patterns, and FP prototypes suppress recurrent distractors identified from earlier verified samples.

These examples indicate that the clustering-compressed prototype memory is not tied to individual representative samples. Instead, it provides reusable target and background priors for subsequent unseen images and instances of the corresponding semantic category.

A.2.3 Training-Free Adaptation to Held-Out Categories

We further evaluate whether representative feedback can improve batch annotation for categories excluded from task-specific model training. In this controlled setting, the model is trained on the original task data containing categories such as Storage Tank and Aircraft, whereas Wind Turbine and Vehicle are treated as held-out categories.

At test time, a small number of representative samples from each held-out category are first predicted and subsequently verified. Their TP, FN, and FP regions are used to construct category-relevant prototype memory. The resulting memory is then applied to the remaining images of the same held-out category without gradient-based fine-tuning or parameter updates.

Figure 8 shows that the static model can obtain initial responses for held-out categories through the current visual prompt, but still suffers from missed regions and recurrent false positives. After prototype evolution, the positive memory improves target recovery, while the negative memory suppresses several background responses that recur across the annotation batch.

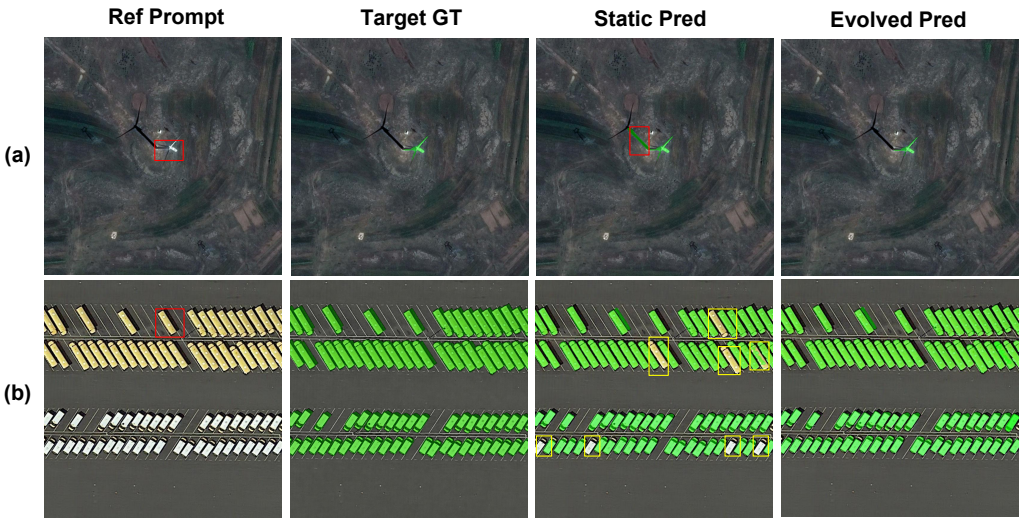


Figure 8: Qualitative comparison of training-free prototype evolution on categories held out from task-specific model training. For each held-out category, representative samples are first verified to construct the historical prototype memory, which is then applied to the remaining target images without updating model parameters. From left to right, the columns show the current reference prompt, target ground truth, static prediction, and evolved prediction. The displayed categories are Wind Turbine and Vehicle.In the static predictions, red boxes highlight falsely predicted target pixels, corresponding to false-positive regions, whereas yellow boxes indicate missed target pixels, corresponding to false-negative regions.

A.2.4 Positive Prototype Allocation

The positive memory contains both TP and FN prototypes. TP prototypes preserve stable characteristics of correctly recognized targets, whereas FN prototypes capture difficult patterns associated with targets missed by static inference. We keep the total positive-memory capacity fixed at eight and vary its allocation between TP and FN prototypes.

As shown in Table 4, FN-dominant allocations are generally more effective for the two evaluated challenging categories. For Storage Tank, the 1:7 allocation obtains the highest gIoU, whereas 2:6 obtains the highest mIoU. For Vehicle, 2:6 performs best on both metrics.

The results indicate that allocating sufficient capacity to FN prototypes is important because they preserve hard target patterns that are absent from static predictions. We use $m = 2$ TP prototypes and $n = 6$ FN prototypes in the remaining experiments because this setting provides the most consistent overall behavior across the two evaluated categories.

A.2.5 Hyperparameter Sensitivity

We evaluate the positive-prior injection strength α and the negative spatial suppression strength λ. The two parameter studies are conducted sequentially. We first vary α while retaining the original suppression coefficient $\lambda = 5.0$. After selecting $\alpha = 1.0$, we fix α and refine λ within a narrower range. Because the fixed value of λ differs between the two blocks, their absolute metric values should not be directly compared.

As shown in Table 5, $\alpha = 1.0$ provides the best gIoU and mIoU in the positive-guidance study. With α fixed to 1.0, the results remain stable when λ varies from 0.8 to 1.1. We choose $\lambda = 1.1$ because it achieves the highest gIoU while maintaining an mIoU comparable to the best-performing settings.

Table 4: Ablation study of different TP-to-FN prototype allocations under a fixed positive-memory budget of eight. The best result for each metric is shown in bold.

(a) Storage Tank			
Exp.	**TP:FN**	**gIoU**	**mIoU**
1	1:7	**56.54**	75.87
2	2:6	56.28	**76.30**
3	4:4	40.95	67.52
4	6:2	54.53	71.99
5	7:1	54.42	71.90
(b) Vehicle			
Exp.	**TP:FN**	**gIoU**	**mIoU**
1	1:7	40.20	59.29
2	2:6	**41.24**	**59.50**
3	4:4	40.39	58.61
4	6:2	40.17	59.35
5	7:1	32.25	57.21

Table 5: Sensitivity analysis of the positive prototype injection strength α and negative spatial suppression strength λ.

Parameter	**Value**	**gIoU**	**mIoU**
Positive prototype injection strength α ($\lambda = 5.0$)			
α	0.8	49.54	74.66
α	0.9	48.56	73.44
α	1.0	**49.83**	**74.91**
α	1.1	49.38	73.95
Negative suppression strength λ ($\alpha = 1.0$)			
λ	0.8	51.89	78.18
λ	0.9	51.91	**78.20**
λ	1.0	51.92	**78.20**
λ	1.1	**51.94**	78.19

The relatively narrow variation within each controlled block indicates that prototype evolution does not depend on a highly specific parameter value. Unless otherwise stated, all remaining experiments use $\alpha = 1.0$ and $\lambda = 1.1$.

A.2.6 Fixed-Budget Performance–Efficiency Analysis

A central design objective of prototype evolution is to reuse historical feedback without causing unbounded memory growth as more samples are verified. We therefore evaluate two factors: the number of retained positive and negative prototypes and the annotation-sequence length.

Table 6 varies the positive and negative prototype budgets while keeping the annotation sequence fixed. The 8/8 configuration achieves the highest mIoU. Increasing the retained budget to 16/16 or 32/32 does not provide further accuracy gains, suggesting that retaining too many prototypes may cause the memory to overfit instance-specific details rather than capturing generalizable category-level semantics, thereby hindering cross-image generalization.

Peak GPU memory remains virtually unaffected by the prototype budget, as the prototype tensors introduce minimal overhead compared to the heavy memory requirements of the visual encoder, LLM,

Table 6: Performance–efficiency analysis under different positive and negative prototype-memory budgets. Memory and latency are reported in GB and ms, respectively.

Pos./Neg.	Mem.	Lat.	FPS	mIoU
4/4	8.34	410.44	2.44	77.29
8/8	8.34	408.61	2.45	78.19
16/16	8.34	391.82	2.55	77.85
32/32	8.34	392.14	2.55	77.37

Table 7: Performance–efficiency analysis under different annotation sequence lengths. Memory and latency are reported in GB and ms, respectively.

Seq. Len.	Mem.	Lat.	FPS	mIoU
50	7.79	424.72	2.35	77.06
100	7.79	402.88	2.48	76.67
500	8.34	399.20	2.51	77.71
1000	8.34	431.95	2.32	77.12

and mask decoder. Latency and FPS remain within a similar range and do not exhibit systematic growth as the retained prototype budget increases.

Table 7 fixes the retained prototype budget and increases the annotation-sequence length from 50 to 1000. Across this twenty-fold increase, memory usage remains within a narrow range and latency does not increase with sequence length.

The memory increases from 7.79 GB to 8.34 GB when the prototype banks reach their configured capacity, after which the model-facing memory remains bounded. This behavior is consistent with the fixed-budget update mechanism: newly verified region embeddings are temporarily combined with the previously retained centers, compressed through K-Means, and discarded after the updated centers have been obtained. Consequently, the model receives a bounded number of prototype tokens rather than the complete annotation history.

Table 8: Comparison of memory compression strategies under a fixed budget. Positive memory uses eight prototypes (two TP and six FN), while negative memory uses eight FP prototypes.

Exp.	Compression Strategy	gIoU	mIoU
1	Random sampling	39.87	58.28
2	Importance-based selection	18.46	54.56
3	Similarity-based merging	31.28	57.06
4	K-Means clustering	**41.24**	**59.50**

A.3 Additional Implementation Details

A.3.1 Model Configuration

Following PSALM, UniEvo-RS is initialized from its pretrained weights and uses a Phi-based LLM, a Swin-B visual encoder, an MSDeformAttn pixel decoder, and a Mask2Former-style predictor. The mask predictor has a hidden dimension of 256, 100 object queries, 8 attention heads, and 9 transformer decoder layers, resulting in 10 prediction stages when the initial query prediction is included.

The visual encoder is frozen during training. The multimodal projector, LLM, pixel decoder, and mask predictor are fine-tuned jointly. Visual prompts are represented as points, bounding boxes, scribbles, or region masks.

A.3.2 MULTI-TASK TRAINING CONFIGURATION

We train UniEvo-RS for 10 epochs using a base learning rate of 1×10^{-6}, a cosine learning-rate schedule, and a warmup ratio of 0.03. Batch-wise alternating multi-task training is used: each mini-batch is sampled from one task setting, and the task source alternates across successive mini-batches.

The per-device batch size is 4, and the maximum multimodal sequence length is 2048. The same shared architecture and mask-generation pathway are used for all five quantitative task settings.

A.3.3 PROTOTYPE-EVOLUTION CONFIGURATION

The positive memory contains $m = 2$ TP prototypes and $n = 6$ FN prototypes, while the negative memory contains $r = 8$ FP prototypes. The TP, FN, and FP update pools are clustered independently.

Whenever newly verified region embeddings are added, the previously retained centers and the new embeddings are temporarily combined and K-Means is rerun. Only the updated cluster centers are retained after each evolution step. The full historical feature pool is discarded.

We use a positive-prior injection strength of $\alpha = 1.0$, a negative confidence threshold of $\tau = 0.7$, and a spatial suppression strength of $\lambda = 1.1$. Prototype guidance is activated only during representative exemplar-driven batch inference, and all model parameters remain fixed.

A.4 MEMORY COMPRESSION STRATEGY

Table 8 compares four strategies under the same fixed memory budget. K-Means clustering performs best on both metrics, suggesting that preserving representative feature modes is more effective than sample-level selection or merging in this setting. This result supports its use for compressing the evolving prototype memory while keeping the model-facing memory size fixed.