MASK–LANGUAGE CYCLE CONSISTENCY FOR TEST-TIME ADAPTATION IN REASONING SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning segmentation aims to predict pixel-level masks from complex, open-ended language queries. Despite recent progress, vision–language models remain vulnerable to test-time distribution shifts, such as environmental changes and image corruptions, which can disrupt mask–language alignment and degrade segmentation quality. Existing test-time adaptation (TTA) methods for segmentation primarily rely on entropy minimization or self-training, treating segmentation as a one-way prediction problem while overlooking its bidirectional cross-modal structure. We propose Cycle-ML, a TTA framework based on mask–language cycle consistency. At inference time, Cycle-ML predicts a mask from the input query, generates a referring expression grounded in the predicted mask, and regrounds the generated expression to obtain a cycle-reconstructed mask. It then uses both mask-level and language-level consistency as self-supervised adaptation signals. To mitigate error amplification from incorrect yet self-consistent predictions, we further introduce consistency–reliability learning, which estimates sample reliability from joint mask and language consistency and adaptively weights the resulting supervision. Experiments on multiple reasoning and referring segmentation benchmarks under 15 synthetic corruptions show that Cycle-ML consistently outperforms strong TTA baselines, demonstrating the effectiveness of reliability-aware mask–language cycle consistency for robust vision–language grounding.

1 INTRODUCTION

Reasoning segmentation is a challenging vision–language grounding task that aims to localize pixel-level regions from complex natural-language queries Lai et al. (2024). Unlike semantic or instance segmentation, which predicts masks according to a predefined semantic label space Long et al. (2015); He et al. (2017), reasoning segmentation must interpret open-ended, context-dependent instructions that may only implicitly specify the target. This capability is particularly relevant to applications such as intelligent assistants and robotics, where systems must ground free-form user intent in fine-grained visual regions. Extending Referring Expression Segmentation (RES) Hu et al. (2016), reasoning segmentation places greater demands on the integration of visual perception, language–region alignment, and high-level reasoning. Accordingly, recent approaches have increasingly built on Multimodal Large Language Models (MLLMs) Liu et al. (2023), leveraging their capabilities in multimodal understanding and general-purpose reasoning.

Despite this progress, current reasoning segmentation models remain vulnerable to test-time distribution shifts, including environmental changes, variations in imaging conditions, and common image corruptions. Such shifts can not only degrade mask quality but also disrupt mask–language alignment, resulting in erroneous grounding. Without adaptation to unseen test conditions, model performance can deteriorate substantially, limiting reliability in real-world deployments. This motivates an important yet largely unexplored problem: test-time adaptation (TTA) for reasoning segmentation, which aims to restore mask–language alignment using only unlabeled test inputs.

Most existing TTA methods for segmentation rely on entropy minimization Wang et al. (2020); Niu et al. (2023) or self-training Wang et al. (2022); Chen et al. (2022) to derive self-supervised signals from unlabeled test data. However, these methods typically treat segmentation as a one-way visual prediction problem, overlooking the bidirectional cross-modal structure of reasoning segmentation. In vision–language grounding, masks and language are inherently coupled: a query should map to

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.

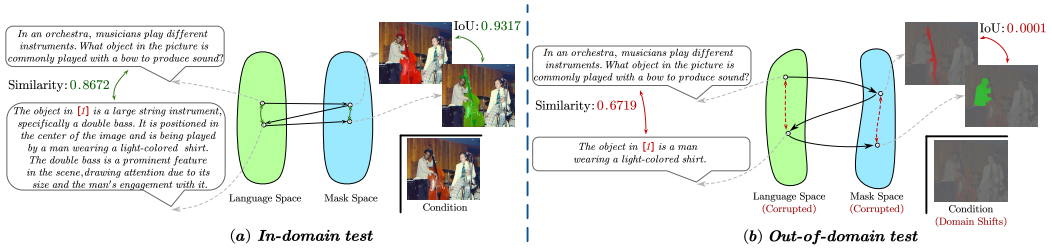


Figure 1: For in-domain condition (a), the mapping between mask and language satisfies cycle consistency. For out-of-domain condition (b), distribution shifts break this mask–language alignment, resulting in incorrect reasoning and degraded segmentation quality.

the corresponding mask, while the grounded mask should in turn support a semantically consistent referring description. As illustrated in Fig. 1, we therefore formulate reasoning segmentation as a bidirectional mask–language grounding problem. For in-domain inputs (Fig. 1(a)), the language-to-mask and mask-to-language mappings are expected to form a consistent cycle. Distribution shifts, however, can disrupt this alignment (Fig. 1(b)), resulting in incorrect grounding and degraded segmentation quality. This perspective suggests that mask–language cycle consistency offers a natural self-supervised signal for test-time adaptation.

While cycle consistency provides a natural self-supervised signal, naively enforcing it at test time can be unreliable. Under severe distribution shifts, a model may produce predictions that are internally consistent yet semantically incorrect, thereby reinforcing errors rather than correcting them. To mitigate this issue, we assess the reliability of cycle-induced supervision using consistency in both modalities (Fig. 2): mask consistency between the forward and cycle-reconstructed masks, and language consistency between the original query and the reconstructed description. These complementary signals jointly provide a more robust reliability estimate than either alone, allowing unreliable cycles to be down-weighted for more conservative test-time adaptation.

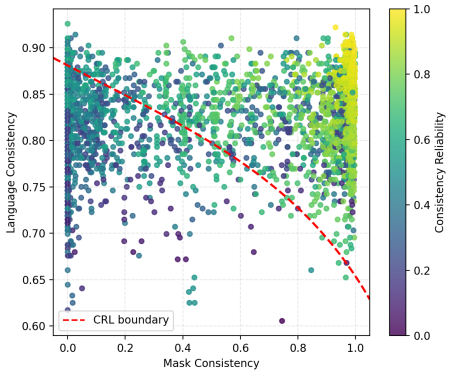


Figure 2: Motivation of the proposed reliability measure. Mask consistency is the IoU between forward and reconstructed masks, and language consistency is the similarity between the original query and reconstructed description. Color indicates a ground-truth-based reliability score for visualization only.

We propose **Cycle-ML**, a test-time adaptation framework that closes the **M**ask–**L**anguage loop at inference time for reasoning and referring segmentation. Cycle-ML comprises two components: mask–language consistency (MLC) and consistency–reliability learning (CRL). Given an input query, the model predicts a mask, generates a referring expression grounded in that mask, and regrounds the expression to obtain a cycle-reconstructed mask. MLC uses both mask-level and language-level consistency as cross-modal adaptation signals, while CRL estimates the reliability of cycle-induced supervision from joint mask and language consistency and adaptively reweights the consistency losses. Together, these components help restore mask–language alignment and improve robustness under distribution shifts.

Our contributions are as follows: (1) We view test-time adaptation for reasoning segmentation as the recovery of mask–language alignment and construct an explicit mask–language cycle to exploit its bidirectional cross-modal structure. (2) We propose Cycle-ML, a reliability-aware test-time adaptation framework that combines mask-level and language-level consistency as self-supervised signals, while adaptively down-weighting unreliable cycle-induced supervision. (3) Extensive experiments on reasoning and referring segmentation benchmarks, including evaluations across 15 synthetic corruptions, show that Cycle-ML consistently outperforms strong test-time adaptation baselines.

2 Related Work

2.1 Reasoning Segmentation

Reasoning segmentation is a vision–language grounding task that aims to localize pixel-level regions from open-ended queries requiring implicit reasoning. Unlike semantic segmentation, which operates over predefined categories, and referring expression segmentation (RES) Kazemzadeh et al. (2014); Yu et al. (2016), which typically grounds explicit textual descriptions, reasoning segmentation requires stronger visual reasoning and language–region alignment. LISA Lai et al. (2024) introduced this task by equipping multimodal large language models (MLLMs) Liu et al. (2023) with pixel-level segmentation capabilities. Subsequent studies Zhang et al. (2023); Yan et al. (2024); Liu et al. (2025) have advanced this paradigm by integrating MLLMs Bai et al. (2025) with segmentation decoders Ravi et al. (2024), expanding training data, and extending the framework to broader dense prediction settings. Despite this progress, existing methods primarily focus on in-domain performance, while their robustness to test-time distribution shifts remains underexplored. Our work addresses this problem from a test-time adaptation perspective.

2.2 Test-Time Adaptation

Test-time adaptation (TTA) aims to improve model robustness under distribution shifts by adapting the model during inference without labeled target data. Representative methods update normalization statistics or selected model parameters using objectives such as entropy minimization Wang et al. (2020), self-training Wang et al. (2022), and contrastive learning Liu et al. (2021). Subsequent studies further improve adaptation stability and efficiency under challenging test streams Niu et al. (2023); Zhao et al. (2023); Ye et al. (2025); Shin et al. (2022). In semantic segmentation, most existing TTA methods formulate adaptation as a one-way visual prediction problem, typically assuming a predefined category space. Consequently, they do not directly exploit the open-ended, bidirectional cross-modal structure of reasoning segmentation. In contrast, our method uses mask–language cycle consistency as a self-supervised adaptation signal.

2.3 Cycle Consistency

Cycle consistency has been widely used as a structural constraint in tasks involving forward and reverse mappings, including visual tracking Sundaram et al. (2010), machine translation He et al. (2016), structure-from-motion Zach et al. (2010), dense correspondence learning Zhou et al. (2015), depth estimation Godard et al. (2017), and image translation Zhu et al. (2017). It has also been extended to cross-modal retrieval Wu et al. (2018) and temporal video alignment Dwibedi et al. (2019), where round-trip consistency provides supervision without direct annotations. Our work extends this principle to multimodal reasoning segmentation by constructing bidirectional mappings between masks and language. To our knowledge, Cycle-ML is the first framework to use mask–language cycle consistency as a core self-supervised signal for test-time adaptation.

3 Method

3.1 Preliminaries

Problem Formulation. We consider test-time adaptation for referring and reasoning segmentation under distribution shifts. Given a target-domain image $\mathcal{X}_i$ and a natural-language query q_i, a pretrained vision–language segmentation model $\mathcal{F}_{\mathrm{VL}}(\cdot;\theta_0)$ predicts a mask $\mathcal{M}_i \in [0,1]^{H\times W}$ corresponding to the region described by q_i. The model may additionally take visual prompts, such as points, boxes, or masks, depending on the underlying task and architecture.

Let $\mathcal{D}^{\mathrm{src}}$ denote the labeled source-domain data used to train the initial model parameters θ_0, and let $\mathcal{D}^{\mathrm{tgt}} = \{(\mathcal{X}_i, q_i)\}_{i=1}^{N}$ denote an unlabeled target-domain test stream. During adaptation, neither source-domain data nor target-domain annotations are available. Our goal is to update selected model parameters online using only the model's mask and language predictions, thereby improving segmentation performance and restoring mask–language alignment under distribution shifts.

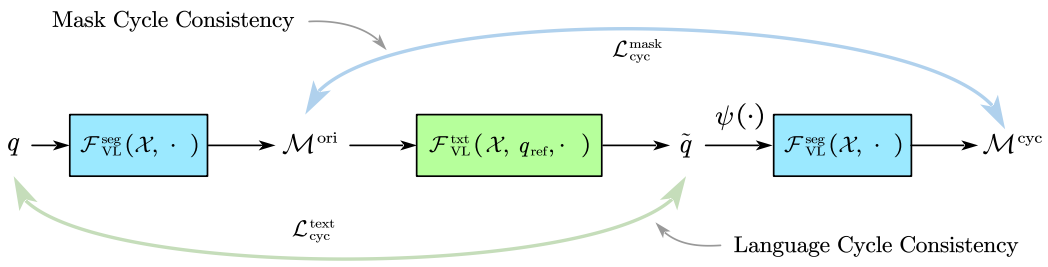


Figure 3: Illustration of Cycle-ML pipeline.

Baseline Architecture. We build upon UniPixel (Liu et al., 2025), a unified large multimodal model for pixel-level visual reasoning. Its architecture comprises four main components: (1) a ViT-based visual encoder that processes images at dynamic resolutions; (2) a language-model backbone, Qwen2.5-VL (Bai et al., 2025), that integrates visual and textual tokens; (3) a SAM2-based mask decoder (Ravi et al., 2024) for dense mask prediction; and (4) a prompt encoder and an object memory bank that support point, box, and mask prompts as well as mask-grounded reasoning.

At inference time, given $(\mathcal{X}, q)$, UniPixel encodes the image and query into multi-modal tokens, then decodes a textual response and inserts a special segmentation token `<SEG>` into the output stream. The hidden state of `<SEG>` is projected to the mask decoder, which predicts a mask $\mathcal{M}$ for the referred region. The object memory bank further allows the model to inject region-specific features into subsequent reasoning via special memory tokens `<MEM>`. In this work, we keep this architecture fixed and introduce a test-time adaptation procedure that constructs a mask–language cycle using only the model's own predictions, without requiring target-domain annotations.

3.2 Cycle-ML Framework

Building on the above architecture, we introduce Cycle-ML, a test-time adaptation framework that exploits a closed Mask–Language loop. Concretely, Cycle-ML (i) enforces *mask–language consistency* by cycling between segmentation and referring expression generation (Section 3.2.1), and (ii) performs *consistency–reliability learning* to selectively trust cycle-induced signals via an explicit reliability modeling (Section 3.2.2). These components are unified in a self-training objective for online model update at test time (Section 3.2.3), enabling progressive correction of mask–language alignment on unlabeled test images under domain shift. The pipeline is illustrated in Fig. 3.

3.2.1 Mask–Language Consistency

Forward segmentation. Given a test sample $(\mathcal{X}, q)$, we feed $(\mathcal{X}, q)$ into the vision–language model $\mathcal{F}_{\text{VL}}^{\text{seg}}$ (i.e., $\mathcal{F}_{\text{VL}}$ with its segmentation branch enabled) and obtain the segmentation probability map:

$$\mathcal{P}^{\text{ori}} = \sigma\big(\mathcal{F}_{\text{VL}}^{\text{seg}}(\mathcal{X}, q)\big) \in [0,1]^{H\times W}, \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function. The corresponding binary mask is:

$$\mathcal{M}^{\text{ori}} = \mathbb{I}\big[\mathcal{P}^{\text{ori}} > 0.5\big]\,. \tag{2}$$

Reverse referring generation. To close the cycle, we treat the predicted mask $\mathcal{M}^{\text{ori}}$ as a visual prompt for the *text* branch of the model. Specifically, we feed $(\mathcal{X}, q_{\text{ref}}, \mathcal{M}^{\text{ori}})$ into the vision–language model with its text-generation branch $\mathcal{F}_{\text{VL}}^{\text{txt}}$ and obtain a referring expression:

$$\tilde{q} = \mathcal{F}_{\text{VL}}^{\text{txt}}\big(\mathcal{X}, q_{\text{ref}}, \mathcal{M}^{\text{ori}}\big), \tag{3}$$

where q_{ref} is a mask-grounded instruction for generating a referring expression. This step effectively solves the reverse problem *mask* $\rightarrow$ *language* using the same model.

Cycle segmentation. The generated expression $\tilde{q}$ may be noisy and stylistically diverse. We therefore apply a post-processing function $\psi(\cdot)$ that first extracts a content-preserving description (by removing memory tokens, indices, and boilerplate) and then converts it into a segmentation instruction. Feeding $(\mathcal{X}, \psi(\tilde{q}))$ into the segmentation branch $\mathcal{F}_{\text{VL}}^{\text{seg}}$ yields a second probability map:

$$\mathcal{P}^{\text{cyc}} = \sigma\big(\mathcal{F}_{\text{VL}}^{\text{seg}}(\mathcal{X}, \psi(\tilde{q}))\big) \in [0,1]^{H\times W}, \tag{4}$$

with its corresponding binary mask:

$$\mathcal{M}^{\text{cyc}} = \mathbb{I}[\mathcal{P}^{\text{cyc}} > 0.5]\,. \tag{5}$$

This closes the *mask* $\rightarrow$ *language* $\rightarrow$ *mask* cycle.

Mask consistency. To enforce mask-level consistency between the original prediction $\mathcal{P}^{\text{ori}}$ and the cycle prediction $\mathcal{P}^{\text{cyc}}$, we adopt the mask decoding loss used in SAM2 and apply it bidirectionally, using each binary mask to supervise the other probability map:

$$\mathcal{L}_{\text{cyc}}^{\text{mask}} = \mathcal{L}_{\textit{mask}}\big(\mathcal{M}^{\text{ori}}, \mathcal{P}^{\text{cyc}}\big) + \mathcal{L}_{\textit{mask}}\big(\mathcal{M}^{\text{cyc}}, \mathcal{P}^{\text{ori}}\big), \tag{6}$$

where $\mathcal{L}_{\textit{mask}}$ is the SAM2 mask decoding loss. This bidirectional supervision not only encourages pixel-level agreement between the two masks, but also couples the forward (*reasoning* / implicit description $\rightarrow$ mask) and reverse (*referring* / explicit description $\rightarrow$ mask) paths through a shared target. By repeatedly enforcing this cycle during test-time adaptation, the model is pushed to align its segmentation outputs with both implicit reasoning-style descriptions and explicit referring expressions.

Language consistency. While mask consistency captures mask-level alignment, it does not directly enforce semantic alignment in the language space. To this end, we further impose a cross-modal text consistency loss between the original user instruction q and the cycle-generated instruction $\psi(\tilde{q})$. Given the shared visual input $\mathcal{X}$, we construct two multi-modal inputs $(\mathcal{X}, q)$ and $(\mathcal{X}, \psi(\tilde{q}))$ and feed them into the vision–language model with its text-generation branch $\mathcal{F}_{\text{VL}}^{\text{txt}}$ (with SAM2 mask decoding disabled) to obtain the last-layer hidden states $\mathbf{H}_{\text{ori}}$, $\mathbf{H}_{\text{cyc}} \in \mathbb{R}^{L\times D}$. Then we retain only textual token embeddings and apply masked mean pooling to obtain language features $\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}} \in \mathbb{R}^{D}$. The language consistency loss is defined as:

$$\mathcal{L}_{\text{cyc}}^{\text{text}} = 1 - \cos\big(\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}}\big), \tag{7}$$

which encourages the model to produce semantically consistent descriptions for the same underlying object under different prompts.

In combination with the mask-consistency loss, this language-level alignment explicitly ties together *implicit* reasoning instructions and *explicit* referring expressions in a shared embedding space, and implicitly sharpens the correspondence between textual descriptions and pixel-level masks. As a result, Cycle-ML strengthens mask–language alignment from both mask-level and language-level perspectives during test-time adaptation.

3.2.2 Consistency–Reliability Learning

Although cycle consistency provides a self-supervised signal, not all test samples are equally reliable: under severe domain shift, the model may still produce *consistent yet wrong* masks and descriptions. To avoid adapting the model on such unreliable cycles, we explicitly quantify sample reliability and use these statistics to construct sample-wise weights for test-time optimization. Concretely, we derive two complementary reliability indicators in the mask and language spaces, and then build an online separability maximization and Gaussian weighting scheme in a low-dimensional consistency feature space.

Reliability indicators. We leverage the cycle defined in Sec. 3.2.1 to obtain, for each test sample, two scalar consistency scores: the mask-level consistency $\mathrm{C}_{\text{mask}} = \mathrm{IoU}\big(\mathcal{M}^{\text{ori}}, \mathcal{M}^{\text{cyc}}\big) \in [0, 1]$ and the language-level consistency $\mathrm{C}_{\text{text}} = \cos\big(\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}}\big) \in [-1, 1]$. These two indicators respond differently under typical failure modes: (1) When the reasoning-style instruction q is misinterpreted but the reverse referring path conditioned on $\mathcal{M}^{\text{ori}}$ correctly localizes the object, the two texts disagree while the masks remain close, yielding low C_{text} but high C_{mask}, so language consistency is more indicative of reliability. (2) Conversely, when the initial reasoning segmentation is correct but the segmentation conditioned on the referring expression is inaccurate, the two instructions can stay semantically similar while the masks diverge, leading to high C_{text} but low C_{mask}, so mask consistency better reflects reliability. These observations show that C_{mask} and C_{text} capture complementary aspects of reliability; we therefore combine them into a joint consistency feature and perform sample-adaptive separation and weighting in this space.

Online separability maximization. For each test sample, we form a 2D reliability vector $\mathbf{h} = \big[\mathrm{C}_{\text{mask}},\ \mathrm{C}_{\text{text}}\big]^{\top} \in \mathbb{R}^2$. We maintain a sliding buffer of recent K reliability vectors:

$$\mathcal{B} = \{\mathbf{h}_1, \ldots, \mathbf{h}_K\}, \quad \Phi = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{R}^{2\times K}. \tag{8}$$

Our goal is to automatically identify a subset of high-confidence cycles whose reliability vectors exhibit consistently high consistency scores, and to separate them from the remaining, potentially unreliable ones. To this end, we formulate a pairwise agreement objective as an approximate proxy for separating the buffered reliability vectors into two groups. Specifically, we seek a partition that aggregates samples with strong within-group affinity under the reliability-space inner product:

$$\max_{\mathbf{S}} \operatorname{Tr}\big(\mathbf{S}^\top \Phi^\top \Phi \mathbf{S}\big), \text{s.t.} \quad \mathbf{S} \in \{0,1\}^{K\times 2}, \sum_{c=1}^{2} S_{kc} = 1 \tag{9}$$

where $\mathbf{S}$ is a binary assignment matrix indicating the cluster membership of each sample in $\mathcal{B}$. Optimizing this combinatorial objective exactly is NP-hard, so we adopt a spectral relaxation Olsson et al. (2008). Let $\mathbf{G} = \Phi^\top \Phi \in \mathbb{R}^{K\times K}$ be the Gram matrix of reliability features, and let $(\lambda_i, \mathbf{u}_i)$ denote its eigenpairs sorted in ascending order. We form a 2D spectral embedding $\mathbf{V} \in \mathbb{R}^{K\times 2}$ by stacking the last two eigenvectors: $\mathbf{V} = [\mathbf{u}_{K-1},\ \mathbf{u}_K]$, where the k-th row $\mathbf{v}_k^\top$ corresponds to sample k. We then assign each sample to one of the two clusters according to the dominant coordinate in this embedding:

$$\text{cluster}(k) = \arg\max_{c\in\{1,2\}} \big|v_{k,c}\big|, \tag{10}$$

where $v_{k,c}$ is the c-th entry of $\mathbf{v}_k$. Since we do not know a priori which cluster corresponds to "reliable" cycles, we simply choose the cluster whose samples have higher mean mask consistency:

$$\mathcal{C}_{\text{rel}} = \arg\max_{c\in\{1,2\}} \frac{1}{|\mathcal{I}_c|} \sum_{k\in\mathcal{I}_c} \mathrm{C}^{(k)}_{\text{mask}}, \tag{11}$$

where $\mathcal{I}_c = \{k \mid \text{cluster}(k) = c\}$. In this way, samples with consistently higher mask-level and language-level consistency scores are automatically separated out as reliable cycles for subsequent reliability weighting.

Gaussian reliability weighting. Given the reliable cluster $\mathcal{C}_{\text{rel}}$, we further refine the contribution of each sample via a smooth, sample-specific weight $w \in (0,1]$. Let $\Phi_{\text{rel}} \in \mathbb{R}^{2\times|\mathcal{I}_{\text{rel}}|}$ denote the sub-matrix of Φ restricted to reliable samples, and let $\boldsymbol{\mu}, \boldsymbol{\sigma}^2 \in \mathbb{R}^2$ be the mean and variance of its columns. For the current sample with feature $\mathbf{h}$, we first clip each dimension: $\tilde{\mathbf{h}} = \min(\mathbf{h}, \boldsymbol{\mu})$, and then define an anisotropic Gaussian kernel in the reliability space as

$$w = \exp\Big(-(\tilde{\mathbf{h}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{h}} - \boldsymbol{\mu})\Big), \tag{12}$$

where $\boldsymbol{\Sigma} = \operatorname{diag}\big(\alpha\,\boldsymbol{\sigma}^2 + \varepsilon\big)$ is a diagonal covariance matrix, $\alpha > 0$ is a smoothing hyperparameter, and ε is a small constant for numerical stability. Samples near or above the reliability center $\boldsymbol{\mu}$ in both dimensions receive weights close to 1, while those farther away are smoothly down-weighted. Unlike hard thresholding on a single confidence score, this design exploits the complementary nature of mask-level and language-level consistency.

3.2.3 Model Optimization

We perform lightweight test-time adaptation by only updating a small subset of parameters while keeping the rest of the model frozen. Concretely, we fine-tune: (i) the language-to-mask and mask-to-language projectors (L $\to$ M and M $\to$ L), (ii) the mask decoder, and (iii) LoRA Hu et al. (2022) adapters on both the visual encoder and the LLM. All other parameters remain fixed to the source-pretrained weights.

For a batch of B test samples, the reliability-aware adaptation objective is

$$\mathcal{L}_{\text{adapt}} = \frac{1}{B} \sum_{i=1}^{B} w_i \Big(\lambda_{\text{mask}}\, \mathcal{L}^{\text{mask}}_{\text{cyc},i} + \lambda_{\text{text}}\, \mathcal{L}^{\text{text}}_{\text{cyc},i}\Big), \tag{13}$$

where w_i is the reliability weight in Eq. equation 12, and λ_{mask} and λ_{text} balance the contributions of mask- and language-level cycle consistency. The gradients of $\mathcal{L}_{\text{adapt}}$ are only back-propagated to the aforementioned tunable components, enabling efficient and stable test-time adaptation without overfitting the full model. The algorithm flow is detailed in Appendix B.

Table 1: Performance (gIoU) of different test-time adaptation methods under 15 synthetic corruptions on the ReasonSeg val/test splits.

Splits	Val						Test					
Corruption	No Adapt	Pseudo-label	TENT	CoTTA	SVDP	Cycle-ML	No Adapt	Pseudo-label	TENT	CoTTA	SVDP	Cycle-ML
Original	63.85	63.25	62.79	64.14	63.21	**64.82**	57.63	55.80	56.43	58.31	57.24	**59.07**
Gaussian Noise	55.89	56.28	57.16	57.03	57.34	**59.42**	52.25	52.78	52.78	54.34	53.41	**55.22**
Shot Noise	55.95	56.47	56.33	56.74	56.48	**57.72**	52.28	52.95	53.06	54.16	53.67	**56.00**
Impulse Noise	57.41	56.87	57.06	58.17	57.92	**60.66**	53.94	53.88	54.30	54.68	54.51	**56.83**
Defocus Blur	58.98	58.79	59.72	60.04	59.85	**60.60**	54.33	54.59	53.78	55.52	54.96	**57.41**
Glass Blur	57.90	58.03	58.47	59.58	58.61	**59.27**	53.68	53.85	52.95	55.00	54.12	**56.81**
Motion Blur	59.14	59.29	59.64	60.99	59.52	**59.99**	55.49	56.05	56.05	57.01	56.62	**58.05**
Zoom Blur	40.72	41.14	41.61	43.26	42.08	**47.82**	40.82	41.67	41.75	43.63	42.54	**46.70**
Snow	60.39	60.76	61.41	62.82	61.67	**63.92**	57.79	58.42	58.51	58.28	58.46	**60.49**
Frost	60.36	60.30	60.63	61.56	60.41	**61.43**	57.55	57.32	57.01	57.51	57.26	**58.55**
Fog	59.93	60.28	61.18	61.52	61.02	**62.86**	58.04	58.57	57.87	59.46	58.73	**59.96**
Brightness	58.89	59.27	59.87	61.30	60.03	**60.91**	56.02	55.99	56.23	55.90	56.11	**57.87**
Contrast	53.99	54.34	54.84	55.42	54.96	**58.27**	52.33	52.69	53.27	53.80	53.48	**55.71**
Elastic Trans.	60.24	60.46	61.71	62.08	61.48	**61.82**	56.58	57.57	57.42	58.27	57.91	**60.17**
Pixelate	59.89	60.03	59.41	60.31	59.76	**60.24**	55.44	55.72	55.57	56.61	56.03	**57.29**
JPEG Comp.	56.65	56.50	57.20	57.91	57.44	**61.68**	53.44	52.75	52.41	53.38	52.96	**55.50**
Average	57.09	57.25	57.75	58.58	57.90	**59.77**	54.00	54.32	54.19	55.17	54.72	**56.84**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate our method on two tasks: **(1) Referring Expression Segmentation.** We conduct experiments on three standard benchmarks: RefCOCO/+/g Kazemzadeh et al. (2014); Mao et al. (2016), which require segmenting the target object referred to by a natural language expression. **(2) Reasoning Segmentation.** To assess the capability of reasoning-aware segmentation, we adopt the ReasonSeg dataset Lai et al. (2024), which focuses on complex language instructions that require compositional reasoning or multi-step understanding. To the best of our knowledge, this is the first exploration of test-time adaptation (TTA) for reasoning segmentation. We draw inspiration from ImageNet-C Hendrycks & Dietterich (2019) and introduce 15 types of synthetic corruptions on the above datasets to simulate challenging test-time distribution shifts.

Evaluation Metrics. Following common practice in referring segmentation, we adopt two metrics: cIoU (the cumulative intersection over the cumulative union) and gIoU (the average of all per-image IoUs). Since cIoU is highly biased toward large-area objects and tends to fluctuate significantly, we primarily report gIoU as the main metric and provide cIoU as a supplementary reference.

TTA Baselines. We compare our method with four representative TTA approaches: **Pseudo-label** Lee et al. (2013): self-training with model-generated pseudo labels. **TENT** Wang et al. (2020): entropy minimization on test samples. **CoTTA** Wang et al. (2022): continual test-time adaptation with EMA teacher and stochastic restoration. **SVDP** Yang et al. (2024): sparse visual domain prompts for dense prediction test-time adaptation. All test-time adaptation baselines are built upon the same reasoning segmentation framework, UniPixel Liu et al. (2025), using the 3B version of Qwen2.5-VLBai et al. (2025) as the underlying VLM.

Implementation Details. We employ a batch size of 1 at test time and accumulate gradients over 8 steps to stabilize optimization. We use the AdamW optimizer with a learning rate of 2×10^{-6}. For adaptation, we insert LoRA adapters with rank $r = 32$ and scaling factor $\alpha_{\text{LoRA}} = 64$. We set the loss weight $\lambda_{\text{mask}} = 1.0$ and $\lambda_{\text{text}} = 0.05$. For Gaussian reliability weighting, we set the hyperparameter $\alpha = 1$. All experiments are conducted on a single NVIDIA RTX A6000 GPU.

4.2 EVALUATION ON BENCHMARK DATASETS

Results on ReasonSeg. As shown in Table 1, Cycle-ML achieves the highest average gIoU on both the validation and test splits under 15 synthetic corruptions. It improves the source-only model by 2.68 and 2.84 points on the validation and test splits, respectively, while outperforming the

Table 2: Performance (gIoU) of different test-time adaptation methods on RefCOCO/+/g under four selected corruptions.

	Method	RefCOCO			RefCOCO+			RefCOCOg		Avg
		val	testA	testB	val	testA	testB	val(U)	test(U)	
Original	No Adapt	80.58	83.02	78.34	76.80	80.33	71.62	77.08	77.67	78.18
	Pseudo-label	80.24	82.81	78.73	76.92	80.29	71.63	77.04	77.64	78.16
	TENT	80.10	83.52	79.02	76.66	80.44	71.60	77.63	77.60	78.32
	Cycle-ML	**81.12**	**83.06**	**79.89**	**78.13**	**81.40**	**73.53**	**78.37**	**79.04**	**79.32**
Impulse Noise	No Adapt	73.24	77.02	72.08	69.16	71.49	65.26	69.51	70.67	71.05
	Pseudo-label	72.86	77.41	73.41	68.98	72.19	66.40	69.89	70.60	71.47
	TENT	72.06	76.60	73.71	68.34	72.36	66.33	69.89	70.33	71.20
	Cycle-ML	**74.54**	**77.65**	**76.29**	**72.19**	**74.57**	**68.10**	**72.66**	**73.74**	**73.72**
Zoom Blur	No Adapt	59.13	58.41	58.40	59.04	59.25	56.66	59.85	60.79	58.94
	Pseudo-label	60.38	59.30	59.16	59.85	60.19	58.00	59.95	61.67	59.81
	TENT	59.71	59.07	59.68	60.58	60.36	57.91	59.51	60.30	59.64
	Cycle-ML	**61.25**	**61.36**	**62.70**	**61.08**	**62.79**	**60.12**	**61.28**	**62.76**	**61.67**
Brightness	No Adapt	77.93	81.31	76.66	74.72	76.66	73.32	75.32	77.40	76.66
	Pseudo-label	78.60	80.63	77.24	74.58	77.54	73.52	76.13	77.59	76.98
	TENT	78.50	80.74	77.36	74.90	77.52	73.57	76.46	77.45	77.06
	Cycle-ML	**80.46**	**81.49**	**78.86**	**75.68**	**79.01**	**74.10**	**78.35**	**78.87**	**78.35**
JPEG Comp.	No Adapt	72.65	75.18	71.87	69.11	72.07	67.46	71.13	71.78	71.41
	Pseudo-label	73.64	75.12	72.46	70.08	72.58	67.07	70.38	72.13	71.68
	TENT	73.07	75.80	71.91	70.25	72.37	67.77	72.07	72.11	71.92
	Cycle-ML	**73.96**	**76.65**	**73.54**	**71.16**	**73.64**	**68.81**	**73.14**	**74.27**	**73.14**
Average	No Adapt	70.74	72.98	69.75	68.01	69.87	65.67	68.95	70.16	69.52
	Pseudo-label	71.37	73.17	70.57	68.37	70.62	66.25	69.09	70.50	69.98
	TENT	70.84	73.05	70.66	68.52	70.65	66.39	69.48	70.05	69.95
	Cycle-ML	**72.55**	**74.38**	**72.85**	**70.03**	**72.50**	**67.78**	**71.36**	**72.41**	**71.72**

strongest TTA baseline by 1.19 and 1.67 points. The improvements are particularly pronounced under severe degradations such as noise, blur, adverse weather, and compression. Cycle-ML also improves performance on the clean "Original" setting, demonstrating that adaptation does not compromise the model's original capabilities. These results suggest that conventional strategies based on pseudo-labeling, entropy minimization, or generic test-time regularization are insufficient for robust pixel-level vision–language reasoning. In contrast, Cycle-ML closes the mask–language loop and incorporates reliability-aware cycle consistency, enabling more stable adaptation and stronger robustness across both clean and corrupted inputs.

Results on Referring Segmentation Benchmarks. Table 2 reports the results on RefCOCO, RefCOCO+, and RefCOCOg. Cycle-ML achieves the highest overall performance under both the clean setting and all four representative corruptions. Averaged across datasets and splits, it improves the strongest baseline by 1.14 points on clean inputs and by 2.20 points across all evaluated settings. Notably, Cycle-ML consistently performs best across all splits under each corruption. These results indicate that naive test-time adaptation strategies that merely optimize prediction confidence or treat language and masks independently are inadequate for open-vocabulary referring segmentation. By jointly exploiting segmentation and language feedback, Cycle-ML promotes mask–language alignment during adaptation and consequently improves robustness under domain shift.

4.3 Ablation Study and Discussion

Analysis of Mask–Language Consistency. We first analyze the impact of the two consistency terms introduced in Cycle-ML on ReasonSeg-Val with 15 synthetic corruptions. Table 3 shows that both cycle-consistency terms bring clear improvements over the setting without any mask/language consistency. Enforcing only mask-level cycle consistency $\mathcal{L}_{\text{cyc}}^{\text{mask}}$ yields the largest gain in gIoU (from 57.09 to 59.32), indicating that pixel-level self-supervision is particularly effective for enhancing overall segmentation quality under corruption. In contrast, using only language-level consistency $\mathcal{L}_{\text{cyc}}^{\text{text}}$ brings more balanced gains and achieves the best cIoU among single-term variants (56.64), suggesting that regularizing the semantic alignment between original queries and cycle-generated

Table 3: Ablation of mask/language-level consistency losses.

None	$\mathcal{L}_{\text{cyc}}^{\text{mask}}$	$\mathcal{L}_{\text{cyc}}^{\text{text}}$	gIoU (Avg.)	cIoU (Avg.)
✓			57.09	55.26
	✓		59.32	56.11
		✓	57.81	56.64
	✓	✓	**59.77**	**56.78**

Table 4: Ablation of consistency–reliability learning strategies.

	gIoU (Avg.)	cIoU (Avg.)
No Adapt	57.09	55.26
None	51.98	51.15
Hard-Assign	58.97	56.90
Gauss-Weight	59.77	56.78

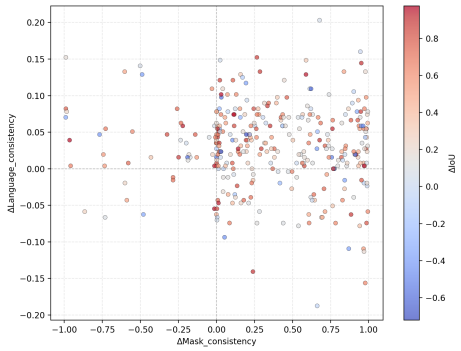


Figure 4: Per-sample changes in mask and language consistency on ReasonSeg-C before and after TTA with Cycle-ML. Points in the upper-right with warmer colors indicate that improved mask–language alignment is accompanied by higher IoU.

expressions mainly benefits query–mask coherence. Moreover, we observe that the improvements from the two terms are largely complementary: $\mathcal{L}_{\text{cyc}}^{\text{mask}}$ focuses on refining spatial details and robustness of predicted masks, while $\mathcal{L}_{\text{cyc}}^{\text{text}}$ primarily strengthens the cross-modal grounding. Combining both terms achieves the best overall trade-off (highest gIoU and competitive cIoU), demonstrating that jointly enforcing mask- and language-level consistency more effectively promotes cross-domain mask–language alignment than relying on either signal alone.

Analysis of Consistency-Reliability Learning. We further ablate the proposed consistency–reliability learning (CRL) by comparing four variants on ReasonSeg-Val with 15 synthetic corruptions: no adaptation, cycle consistency without any reliability modeling, a hard assignment scheme, and our Gaussian reliability weighting based on CRL clustering. Table 4 shows that directly applying cycle losses to all samples without reliability modeling (*None*) significantly degrades performance below the no-adaptation baseline, indicating that blindly trusting all cycle signals is harmful under domain shift. Introducing CRL-based hard assignment (*Hard-Assign*), which selects samples solely based on clustering, not only recovers but surpasses the baseline, confirming the importance of distinguishing reliable from unreliable cycles. Our Gaussian weighting (*Gauss-Weight*) further improves gIoU while maintaining strong cIoU, suggesting that soft reliability weighting in the consistency space is more effective than hard selection for exploiting high-quality cycles and suppressing noisy updates.

Visualization of Mask–Language Alignment. Figure 4 visualizes per-sample changes in mask and language consistency before and after TTA on ReasonSeg-Val within selected corruptions, with the color indicating the corresponding change in IoU. Most points lie in the upper-right region, showing that Cycle-ML tends to simultaneously improve mask- and language-level consistency on corrupted inputs. Moreover, points with larger consistency gains are generally colored in warmer tones, suggesting that enhancing mask–language alignment is closely associated with performance improvements, rather than coming from isolated changes in either modality alone.

5 Conclusion

We present Cycle-ML, a test-time adaptation framework for reasoning and referring segmentation that preserves mask–language alignment under distribution shift. Cycle-ML closes the mask–language cycle at inference time by coupling mask prediction, language reconstruction, and regrounding, and uses mask-level and language-level consistency as adaptation signals. To avoid reinforcing erroneous predictions, it estimates sample reliability and selectively exploits more reliable supervision. With lightweight parameter updates and no extra annotations, Cycle-ML consistently improves robustness across benchmarks and corruption types. These results highlight cycle consistency and reliability-aware adaptation as effective principles for robust vision–language grounding.

AI USE STATEMENT

Generative AI tools were used solely for language editing and coding assistance. Specifically, they were used to improve the clarity, grammar, and presentation of the manuscript and to assist with code implementation and debugging. They were not used to formulate the research problem, develop the core methodology, conduct experiments, analyze results, or draw scientific conclusions. All AI-assisted text was carefully reviewed and revised by the authors, and all AI-assisted code was inspected, tested, and validated for correctness. The authors take full responsibility for the final content of this work, including all text, claims, results, and code produced with the assistance of generative AI.

REFERENCES

Gustavo A Vargas Hakim, David Osowiechi, Mehrdad Noori, Milad Cheraghalikhani, Ali Bahri, Moslem Yazdanpanah, Ismail Ben Ayed, and Christian Desrosiers. Clipartt: Adaptation of clip to new domains at test time. In *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 7092–7101, 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 295–305, 2022.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1801–1810, 2019.

Ky Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39(1):42–47, 1953.

Ky Fan. A generalization of tychonoff's fixed point theorem. *Mathematische Annalen*, 142(3): 305–310, 1961.

Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 270–279, 2017.

Zenghao Guan, Yucan Zhou, Wu Liu, and Xiaoyan Gu. Statistics caching test-time adaptation for vision-language models. *Advances in Neural Information Processing Systems*, 38:6616–6639, 2026.

Jisu Han, Jaemin Na, and Wonjun Hwang. Ranked entropy minimization for continual test-time adaptation. *arXiv preprint arXiv:2505.16441*, 2025.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. Dual learning for machine translation. *Advances in neural information processing systems*, 29, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Ronghang Hu, Marcus Rohrbach, and Trevor Darrell. Segmentation from natural language expressions. In *European conference on computer vision*, pp. 108–124. Springer, 2016.

Yihao Hu, Congyu Qiao, Xin Geng, and Ning Xu. Selective label enhancement learning for test-time adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yu Huang, Zelin Peng, Yichen Zhao, Piao Yang, Xiaokang Yang, and Wei Shen. Medseg-r: Reasoning segmentation in medical images with multimodal large language models. *arXiv preprint arXiv:2506.10465*, 2025.

Adilbek Karmanov, Dayan Guan, Shijian Lu, Abdulmotaleb El Saddik, and Eric Xing. Efficient test-time adaptation of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14162–14171, 2024.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

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.

Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, pp. 896. Atlanta, 2013.

Taeckyung Lee, Sorn Chottananurak, Junsu Kim, Jinwoo Shin, Taesik Gong, and Sung-Ju Lee. Test-time adaptation with binary feedback. *arXiv preprint arXiv:2505.18514*, 2025.

Jixiang Lei and Franz Pernkopf. Promptcal: Entropy-calibrated and prompt-tuned test-time adaptation for semantic segmentation. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 862–869. IEEE, 2025.

Kaiyu Li, Zepeng Xin, Li Pang, Chao Pang, Yupeng Deng, Jing Yao, Guisong Xia, Deyu Meng, Zhi Wang, and Xiangyong Cao. Segearth-r1: Geospatial pixel reasoning via large language model. *arXiv preprint arXiv:2504.09644*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Ye Liu, Zongyang Ma, Junfu Pu, Zhongang Qi, Yang Wu, Ying Shan, and Chang Wen Chen. Unipixel: Unified object referring and segmentation for pixel-level visual reasoning. *arXiv preprint arXiv:2509.18094*, 2025.

Yingnan Liu, Rui Qiao, Mong-Li Lee, and Wynne Hsu. Test-time adaptation by causal trimming. *Advances in Neural Information Processing Systems*, 38:6515–6546, 2026.

Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1215–1224, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

Mehrdad Noori, David Osowiechi, Gustavo Vargas Hakim, Ali Bahri, Moslem Yazdanpanah, Sahar Dastani, Farzad Beizaee, Ismail Ayed, and Christian Desrosiers. Test-time adaptation of vision-language models for open-vocabulary semantic segmentation. *Advances in Neural Information Processing Systems*, 38:74741–74778, 2026.

Carl Olsson, Anders P Eriksson, and Fredrik Kahl. Improved spectral relaxation methods for binary quadratic optimization problems. *Computer Vision and Image Understanding*, 112(1):3–13, 2008.

David Osowiechi, Mehrdad Noori, Gustavo A Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail B Ayed, and Christian Desrosiers. Watt: Weight average test time adaptation of clip. *Advances in neural information processing systems*, 37:48015–48044, 2024.

Boyang Peng, Sanqing Qu, Tianpei Zou, Fan Lu, Ya Wu, Kai Chen, Siheng Chen, Yong Wu, and Guang Chen. Ood-barrier: Build a middle-barrier for open-set single-image test time adaptation via vision language models. *Advances in Neural Information Processing Systems*, 38:150541–150560, 2026.

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.

Mona Schirmer, Dan Zhang, and Eric Nalisnick. Temporal test-time adaptation with state-space models. *arXiv preprint arXiv:2407.12492*, 2024.

Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16928–16937, 2022.

Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European conference on computer vision*, pp. 438–451. Springer, 2010.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

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*, pp. 7201–7211, 2022.

Lin Wu, Yang Wang, and Ling Shao. Cycle-consistent deep generative hashing for cross-modal retrieval. *IEEE Transactions on Image Processing*, 28(4):1602–1612, 2018.

Cilin Yan, Haochen Wang, Shilin Yan, Xiaolong Jiang, Yao Hu, Guoliang Kang, Weidi Xie, and Efstratios Gavves. Visa: Reasoning video object segmentation via large language models. In *European Conference on Computer Vision*, pp. 98–115. Springer, 2024.

Senqiao Yang, Jiarui Wu, Jiaming Liu, Xiaoqi Li, Qizhe Zhang, Mingjie Pan, Yulu Gan, Zehui Chen, and Shanghang Zhang. Exploring sparse visual prompt for domain adaptive dense prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16334–16342, 2024.

Yanyu Ye, Wei Wei, Lei Zhang, Chen Ding, and Yanning Zhang. Domain consistency learning for continual test-time adaptation in image semantic segmentation. *Pattern Recognition*, 165:111585, 2025.

Chang'an Yi, Haotian Chen, Yifan Zhang, Yonghui Xu, Yan Zhou, and Lizhen Cui. From question to exploration: Can classic test-time adaptation strategies be effectively applied in semantic segmentation? In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 10085–10094, 2024.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15922–15932, 2023.

Christopher Zach, Manfred Klopschitz, and Marc Pollefeys. Disambiguating visual relations using loop constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 1426–1433. IEEE, 2010.

Ao Zhang, Yuan Yao, Wei Ji, Zhiyuan Liu, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*, 2023.

Bowen Zhao, Chen Chen, and Shu-Tao Xia. Delta: degradation-free fully test-time adaptation. *arXiv preprint arXiv:2301.13018*, 2023.

Tinghui Zhou, Yong Jae Lee, Stella X Yu, and Alyosha A Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1191–1200, 2015.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, 2017.

A JUSTIFICATION OF THE RELIABILITY-SPACE TRACE OBJECTIVE

In Sec. 3.2.2, our goal is to partition the buffered reliability features into two groups such that samples within the same group exhibit strong consistency in the 2D reliability space. We first show that minimizing within-cluster dispersion leads exactly to a normalized trace maximization objective. We then explain how the unnormalized objective used in Eq. 9 preserves the same pairwise affinity structure and motivates a practical spectral estimation procedure.

A.1 FROM INTRA-CLUSTER COMPACTNESS TO NORMALIZED TRACE MAXIMIZATION

Let $\Phi = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{R}^{2\times K}$ contain the buffered reliability features, and let $\mathbf{S} \in \{0,1\}^{K\times 2}$ be the binary assignment matrix satisfying $\sum_{c=1}^{2} S_{kc} = 1$. We assume that both clusters are nonempty. Let $\boldsymbol{\mu}_c \in \mathbb{R}^2$ denote the mean reliability feature of cluster c. The within-cluster dispersion is

$$\mathcal{L}(\mathbf{S}) = \sum_{k=1}^{K}\sum_{c=1}^{2} S_{kc}\,\|\mathbf{h}_k - \boldsymbol{\mu}_c\|_2^2\,. \tag{14}$$

For a fixed assignment matrix, the cluster means can be written as

$$\mathbf{M} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2] = \Phi\mathbf{S}(\mathbf{S}^\top\mathbf{S})^{-1}. \tag{15}$$

The reconstruction obtained by replacing each sample with its assigned cluster mean is therefore

$$\widehat{\Phi} = \mathbf{M}\mathbf{S}^\top = \Phi\mathbf{P}, \qquad \mathbf{P} = \mathbf{S}(\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top. \tag{16}$$

Because $\mathbf{P}$ is symmetric and idempotent, $\mathbf{P}^\top = \mathbf{P}$ and $\mathbf{P}^2 = \mathbf{P}$. Consequently, Eq. 14 is equivalent to the projection error

$$\mathcal{L}(\mathbf{S}) = \|\Phi - \Phi\mathbf{P}\|_F^2\,. \tag{17}$$

Expanding the Frobenius norm gives

$$\begin{aligned}\mathcal{L}(\mathbf{S}) &= \mathrm{Tr}\big((\Phi - \Phi\mathbf{P})^\top(\Phi - \Phi\mathbf{P})\big)\\ &= \|\Phi\|_F^2 - 2\,\mathrm{Tr}(\Phi^\top\Phi\mathbf{P}) + \mathrm{Tr}(\mathbf{P}^\top\Phi^\top\Phi\mathbf{P})\\ &= \|\Phi\|_F^2 - \mathrm{Tr}(\Phi^\top\Phi\mathbf{P}),\end{aligned} \tag{18}$$

where the last equality follows from the symmetry and idempotence of $\mathbf{P}$, together with the cyclic invariance of the trace.

Since $\|\Phi\|_F^2$ is independent of $\mathbf{S}$, minimizing the within-cluster dispersion is exactly equivalent to maximizing

$$\begin{aligned}\mathrm{Tr}(\Phi^\top\Phi\mathbf{P}) &= \mathrm{Tr}\big(\Phi^\top\Phi\mathbf{S}(\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\big)\\ &= \mathrm{Tr}\big((\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\Phi^\top\Phi\mathbf{S}\big)\,.\end{aligned} \tag{19}$$

For binary one-hot assignments,

$$\mathbf{S}^\top\mathbf{S} = \mathrm{diag}(n_1, n_2), \tag{20}$$

where n_c is the number of samples assigned to cluster c. Hence, the exact discrete objective induced by within-cluster compactness is

$$\begin{aligned}\max_{\mathbf{S}} \quad & \mathrm{Tr}\big((\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\Phi^\top\Phi\mathbf{S}\big)\,,\\ \text{s.t.} \quad & \mathbf{S}\in\{0,1\}^{K\times 2}, \qquad \sum_{c=1}^{2} S_{kc} = 1,\\ & \sum_{k=1}^{K} S_{kc} \geq 1, \qquad c = 1, 2.\end{aligned} \tag{21}$$

Let $\mathcal{I}_c = \{k : S_{kc} = 1\}$. The normalized objective can also be written as

$$\begin{aligned}&\mathrm{Tr}\big((\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\Phi^\top\Phi\mathbf{S}\big)\\ &\quad= \sum_{c=1}^{2}\frac{1}{n_c}\left\|\sum_{k\in\mathcal{I}_c}\mathbf{h}_k\right\|_2^2 = \sum_{c=1}^{2}\frac{1}{n_c}\sum_{i,j\in\mathcal{I}_c}\mathbf{h}_i^\top\mathbf{h}_j.\end{aligned} \tag{22}$$

Thus, minimizing within-cluster dispersion is equivalent to maximizing the cluster-size-normalized pairwise agreement within each group.

A.2 Practical surrogate and spectral relaxation

Directly optimizing Eq. 21 over binary assignments is combinatorial. In practice, we use the simpler objective

$$\begin{aligned}\max_{\mathbf{S}}\quad &\mathrm{Tr}\big(\mathbf{S}^\top\Phi^\top\Phi\mathbf{S}\big),\\ \text{s.t.}\quad &\mathbf{S}\in\{0,1\}^{K\times 2},\qquad \sum_{c=1}^{2}S_{kc}=1.\end{aligned} \tag{23}$$

This unnormalized objective satisfies

$$\begin{aligned}\mathrm{Tr}\big(\mathbf{S}^\top\Phi^\top\Phi\mathbf{S}\big) &= \sum_{c=1}^{2}\left\|\sum_{k\in\mathcal{I}_c}\mathbf{h}_k\right\|_2^2\\ &= \sum_{c=1}^{2}\sum_{i,j\in\mathcal{I}_c}\mathbf{h}_i^\top\mathbf{h}_j.\end{aligned} \tag{24}$$

It therefore encourages samples assigned to the same group to have large aggregate inner-product similarity.

Both the exact normalized objective and the practical unnormalized objective are constructed from the same Gram matrix

$$\mathbf{G} = \Phi^\top\Phi, \qquad G_{ij} = \mathbf{h}_i^\top\mathbf{h}_j. \tag{25}$$

Their difference lies in the cluster-size normalization. In particular, if $n_1 = n_2 = n$, then

$$\mathrm{Tr}(\mathbf{S}^\top\mathbf{G}\mathbf{S}) = n\ \mathrm{Tr}\big((\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\mathbf{G}\mathbf{S}\big). \tag{26}$$

Thus, they are proportional under balanced assignments. For general cluster sizes, however, they are not exactly equivalent. We therefore interpret Eq. 23 as a practical pairwise-agreement surrogate that preserves the Gram-matrix affinity structure of the exact compactness objective.

To derive a tractable spectral representation, define the normalized indicator matrix

$$\mathbf{Y} = \mathbf{S}(\mathbf{S}^\top\mathbf{S})^{-1/2}. \tag{27}$$

It satisfies

$$\mathbf{Y}^\top\mathbf{Y} = \mathbf{I}_2, \tag{28}$$

and the exact normalized objective becomes

$$\mathrm{Tr}(\mathbf{Y}^\top\mathbf{G}\mathbf{Y}) = \mathrm{Tr}\big((\mathbf{S}^\top\mathbf{S})^{-1}\mathbf{S}^\top\mathbf{G}\mathbf{S}\big). \tag{29}$$

Dropping the discrete normalized-indicator structure of $\mathbf{Y}$, while retaining column orthonormality, gives the continuous relaxation

$$\begin{aligned}\max_{\mathbf{Q}\in\mathbb{R}^{K\times 2}}\quad &\mathrm{Tr}(\mathbf{Q}^\top\mathbf{G}\mathbf{Q}),\\ \text{s.t.}\quad &\mathbf{Q}^\top\mathbf{Q}=\mathbf{I}_2.\end{aligned} \tag{30}$$

Let the eigenvalues of the symmetric positive semidefinite matrix $\mathbf{G}$ be ordered as $\lambda_1 \le \cdots \le \lambda_K$, with corresponding orthonormal eigenvectors $\{\mathbf{u}_i\}_{i=1}^K$. By the Ky Fan maximum principle (Fan,

1953; 1961), the solution to Eq. 30 is any orthonormal basis of the eigenspace associated with the two largest eigenvalues. One convenient choice is

$$\mathbf{Q}^\star = [\mathbf{u}_{K-1}, \mathbf{u}_K], \tag{31}$$

with optimal value

$$\mathrm{Tr}\big((\mathbf{Q}^\star)^\top \mathbf{G}\mathbf{Q}^\star\big) = \lambda_{K-1} + \lambda_K. \tag{32}$$

Equation 30 is therefore an exact continuous relaxation of the normalized compactness objective. For the unnormalized objective used in practice, the same top-eigenvector construction should be interpreted as a spectral surrogate based on the shared Gram affinity matrix, rather than as an exact relaxation of the unnormalized discrete problem.

A.3 Spectral embedding and practical estimation

We use

$$\mathbf{V} = \mathbf{Q}^\star = [\mathbf{u}_{K-1}, \mathbf{u}_K] \in \mathbb{R}^{K\times 2} \tag{33}$$

as the spectral representation of the buffered samples. The k-th row $\mathbf{v}_k^\top$ represents sample k in the two dominant spectral directions. A binary partition is then obtained using the dominant-coordinate rule

$$\mathrm{cluster}(k) = \arg\max_{c\in\{1,2\}} |v_{k,c}|, \qquad k = 1, \ldots, K. \tag{34}$$

The absolute value makes the rule invariant to the sign ambiguity of each eigenvector.

This rounding step is a lightweight practical estimator. Ky Fan's theorem guarantees that the top-two eigenspace solves the continuous problem in Eq. 30, but it does not imply that Eq. 34 recovers a global optimizer of the original discrete objective. Moreover, when the leading eigenvalues are repeated or nearly repeated, the basis within the leading eigenspace may not be unique, and the resulting coordinate-wise assignment may depend on the basis returned by the eigensolver. We therefore interpret the rounding rule as an efficient implementation choice rather than an optimality-preserving discretization.

Overall, minimizing within-cluster dispersion leads exactly to the normalized trace objective in Eq. 21. The unnormalized objective in Eq. 9 retains the same pairwise reliability affinities and serves as a practical surrogate. The top-two eigenvectors of $\mathbf{G} = \Phi^\top\Phi$ provide a principled continuous representation through the Ky Fan characterization, while the rounding rule in Eq. 34 converts this representation into the two reliability groups used by our method.

B Implementation Details and Algorithms

In this subsection we summarize the practical implementation of Cycle-ML with consistency–reliability learning (CRL). We focus on three components that are sufficient to reproduce our method: (i) the CRL-based reliability weighting (Alg. 1), (ii) the per-sample cycle-consistent training step (Alg. 2), and (iii) the overall test-time adaptation loop (Alg. 3).

CRL-based reliability weight (Alg. 1). For each test sample, Sec. 3.2.2 defines a 2D reliability vector

$$\mathbf{h} = \big[\mathrm{C}_{\text{mask}},\, \mathrm{C}_{\text{text}}\big]^\top \in \mathbb{R}^2, \tag{35}$$

where $\mathrm{C}_{\text{mask}} = \mathrm{IoU}\big(\mathcal{M}^{\text{ori}}, \mathcal{M}^{\text{cyc}}\big) \in [0,1]$ and $\mathrm{C}_{\text{text}} = \cos\big(\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}}\big) \in [-1,1]$. We maintain a sliding buffer of recent K reliability vectors and apply the spectral relaxation of Sec. 3.2.2 followed by the Gaussian weighting of Eq. 12 to obtain a sample-wise weight $w \in (0,1]$.

This algorithm is the online separability maximization and Gaussian reliability weighting procedure described in Sec. 3.2.2, implemented in the 2D consistency feature space $\mathbf{h} = [\mathrm{C}_{\text{mask}}, \mathrm{C}_{\text{text}}]^\top$.

Per-sample Cycle-ML training step (Alg. 2). We next summarize the three-stage cycle and its associated losses for a single test sample $(\mathcal{X}, q)$. All symbols are consistent with Sec. 3.2.1 and Sec. 3.2.2.

Algorithm 1 CRL-based reliability weight for current sample

Require: current reliability vector $\mathbf{h} = [\mathrm{C}_{\text{mask}}, \mathrm{C}_{\text{text}}]^\top$; buffer $\mathcal{B} = \{\mathbf{h}_k\}_{k<t}$ with size $\leq K$; Gaussian smoothing parameter $\alpha > 0$; numerical constant $\varepsilon > 0$.
Ensure: reliability weight $w \in (0, 1]$ for the current sample.
append $\mathbf{h}$ to buffer $\mathcal{B}$
if $|\mathcal{B}| < 2$ **then**
  return $w \leftarrow 1.0$ {not enough history}
end if
stack all vectors into $\Phi = [\mathbf{h}_1, \ldots, \mathbf{h}_K] \in \mathbb{R}^{2\times K}$
compute Gram matrix $\mathbf{G} \leftarrow \Phi^\top \Phi \in \mathbb{R}^{K\times K}$
compute eigenvalues and eigenvectors of $\mathbf{G}$, sorted in ascending order: $\{(\lambda_i, \mathbf{u}_i)\}_{i=1}^{K}$
form 2D spectral embedding $\mathbf{V} \leftarrow [\mathbf{u}_{K-1}, \mathbf{u}_K] \in \mathbb{R}^{K\times 2}$
for each sample index $k = 1, \ldots, K$ **do**
  let $\mathbf{v}_k^\top$ be the k-th row of $\mathbf{V}$
  assign

$$\text{cluster}(k) \leftarrow \arg\max_{c\in\{1,2\}} \big|v_{k,c}\big|$$

end for
collect sample indices $\mathcal{I}_c \leftarrow \{k \mid \text{cluster}(k) = c\}$ for $c \in \{1, 2\}$
if $\mathcal{I}_1 = \emptyset$ **or** $\mathcal{I}_2 = \emptyset$ **then**
  return $w \leftarrow 1.0$ {degenerate partition}
end if
let $\mathrm{C}_{\text{mask}}^{(k)}$ denote the mask consistency of sample k
for $c \in \{1, 2\}$ **do**
  compute mean mask consistency

$$\mu_c \leftarrow \frac{1}{|\mathcal{I}_c|} \sum_{k\in\mathcal{I}_c} \mathrm{C}_{\text{mask}}^{(k)}$$

end for
select reliable cluster index

$$c_{\text{rel}} \leftarrow \arg\max_{c\in\{1,2\}} \mu_c$$

form reliable set $\mathcal{I}_{\text{rel}} \leftarrow \mathcal{I}_{c_{\text{rel}}}$
extract reliable features $\Phi_{\text{rel}} \leftarrow [\mathbf{h}_k]_{k\in\mathcal{I}_{\text{rel}}} \in \mathbb{R}^{2\times|\mathcal{I}_{\text{rel}}|}$
compute dimension-wise mean

$$\boldsymbol{\mu} \leftarrow \frac{1}{|\mathcal{I}_{\text{rel}}|} \sum_{k\in\mathcal{I}_{\text{rel}}} \mathbf{h}_k \in \mathbb{R}^2,$$

and variance

$$\boldsymbol{\sigma}^2 \leftarrow \frac{1}{|\mathcal{I}_{\text{rel}}|} \sum_{k\in\mathcal{I}_{\text{rel}}} (\mathbf{h}_k - \boldsymbol{\mu})^{\odot 2} + \varepsilon\,\mathbf{1} \in \mathbb{R}^2,$$

where $\odot$ denotes elementwise squaring.
clip current feature coordinate-wise: $\tilde{\mathbf{h}} \leftarrow \min(\mathbf{h}, \boldsymbol{\mu})$
form diagonal covariance

$$\boldsymbol{\Sigma} \leftarrow \mathrm{diag}\big(\alpha\,\boldsymbol{\sigma}^2 + \varepsilon\big) \in \mathbb{R}^{2\times 2}$$

compute Gaussian weight

$$w \leftarrow \exp\Big(-(\tilde{\mathbf{h}} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\tilde{\mathbf{h}} - \boldsymbol{\mu})\Big)$$

return w

Overall test-time adaptation loop (Alg. 3). Finally, we describe the outer loop over the unlabeled test stream $\mathcal{D}^{\text{tgt}} = \{(\mathcal{X}_i, q_i)\}_{i=1}^{N}$. This loop optimizes the reliability-weighted objective in Sec. 3.2.3, while updating only the selected parameters (language–mask projectors, mask decoder, and LoRA adapters).

All eigen-decomposition and reliability computations in Algorithms 1 and 2 are performed outside the gradient graph and operate solely on 2D consistency features, incurring negligible overhead compared to the main forward and backward passes of $\mathcal{F}_{\text{VL}}$.

Algorithm 2 Cycle-ML training step with CRL for one test sample

Require: test sample $(\mathcal{X}, q)$; vision–language model $\mathcal{F}_{\text{VL}}$ with branches $\mathcal{F}_{\text{VL}}^{\text{seg}}$ and $\mathcal{F}_{\text{VL}}^{\text{txt}}$; reliability buffer $\mathcal{B}$; loss weights $\lambda_{\text{mask}}, \lambda_{\text{text}}$; mask loss $\mathcal{L}_{mask}(\cdot,\cdot)$.

Ensure: reliability-weighted loss $\mathcal{L}_{\text{adapt}}^{(i)}$ for this sample. {forward segmentation (Sec. 3.2.1)}

1: compute

$$\mathcal{P}^{\text{ori}} \leftarrow \sigma\big(\mathcal{F}_{\text{VL}}^{\text{seg}}(\mathcal{X}, q)\big), \quad \mathcal{M}^{\text{ori}} \leftarrow \mathbb{I}\Big[\mathcal{P}^{\text{ori}} > 0.5\Big]$$

{reverse referring generation}

2: form a mask-grounded instruction q_{ref} and compute $\tilde{q} \leftarrow \mathcal{F}_{\text{VL}}^{\text{txt}}\big(\mathcal{X}, q_{\text{ref}}, \mathcal{M}^{\text{ori}}\big)$ {cycle segmentation}

3: apply post-processing $\psi(\cdot)$ to obtain a segmentation instruction $\psi(\tilde{q})$ as in Sec. 3.2.1

4: compute $\mathcal{P}^{\text{cyc}} \leftarrow \sigma\big(\mathcal{F}_{\text{VL}}^{\text{seg}}(\mathcal{X}, \psi(\tilde{q}))\big), \quad \mathcal{M}^{\text{cyc}} \leftarrow \mathbb{I}[\mathcal{P}^{\text{cyc}} > 0.5]$ {mask consistency loss}

5: compute bidirectional mask-consistency loss $\mathcal{L}_{\text{cyc}}^{\text{mask}} \leftarrow \mathcal{L}_{mask}\big(\mathcal{M}^{\text{ori}}, \mathcal{P}^{\text{cyc}}\big) + \mathcal{L}_{mask}\big(\mathcal{M}^{\text{cyc}}, \mathcal{P}^{\text{ori}}\big)$ {language consistency loss}

6: construct two inputs $(\mathcal{X}, q)$ and $(\mathcal{X}, \psi(\tilde{q}))$

7: feed them into $\mathcal{F}_{\text{VL}}^{\text{txt}}$ (with SAM2 mask decoding disabled) to obtain last-layer hidden states

$$\mathbf{H}_{\text{ori}},\ \mathbf{H}_{\text{cyc}} \in \mathbb{R}^{L \times D}$$

8: apply masked mean pooling over textual tokens to get language features

$$\mathbf{z}_{\text{ori}},\ \mathbf{z}_{\text{cyc}} \in \mathbb{R}^{D}$$

9: compute language consistency loss $\mathcal{L}_{\text{cyc}}^{\text{text}} \leftarrow 1 - \cos\big(\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}}\big)$ {reliability indicators and CRL weight}

10: compute mask-level consistency

$$\text{C}_{\text{mask}} \leftarrow \text{IoU}\big(\mathcal{M}^{\text{ori}}, \mathcal{M}^{\text{cyc}}\big)$$

11: compute language-level consistency

$$\text{C}_{\text{text}} \leftarrow \cos\big(\mathbf{z}_{\text{ori}}, \mathbf{z}_{\text{cyc}}\big)$$

12: form reliability vector

$$\mathbf{h} \leftarrow \big[\text{C}_{\text{mask}},\ \text{C}_{\text{text}}\big]^{\top}$$

13: call Algorithm 1 with $(\mathbf{h}, \mathcal{B})$ to obtain $w \in (0, 1]$ {sample-wise adaptation objective}

14: compute unweighted per-sample loss

$$\tilde{\mathcal{L}}_{\text{adapt}}^{(i)} \leftarrow \lambda_{\text{mask}}\, \mathcal{L}_{\text{cyc}}^{\text{mask}} + \lambda_{\text{text}}\, \mathcal{L}_{\text{cyc}}^{\text{text}}$$

15: apply reliability weight

$$\mathcal{L}_{\text{adapt}}^{(i)} \leftarrow w \cdot \tilde{\mathcal{L}}_{\text{adapt}}^{(i)}$$

16: **return** $\mathcal{L}_{\text{adapt}}^{(i)}$

C Additional Implementation Details for Mask–Language Consistency

This section clarifies the concrete forms of the reverse prompt q_{ref} and the post-processing function $\psi(\cdot)$ used in Section 3.2.1.

Reverse prompt q_{ref}. Given the first forward mask prediction $\mathcal{M}^{\text{ori}}$, we construct a `MEM_TOKEN`-based prompt that explicitly marks the referred region in the image and then asks the model to describe that region.

q_{ref} = "Here is an image. The highlighted region ismarked as follows:\n"
+ "[0]: "⟨1⟩ `MEM_TOKEN` "\n"
+ "Describe the object in [0] as a referringexpression. Focus on what makes it distinguishable."

Algorithm 3 Test-time adaptation with Cycle-ML and CRL

Require: pretrained vision–language model $\mathcal{F}_{\text{VL}}$; unlabeled target stream $\mathcal{D}^{\text{tgt}} = \{(\mathcal{X}_i, q_i)\}_{i=1}^{N}$; optimizer OPT;
Ensure: adapted model $\mathcal{F}_{\text{VL}}^{*}$.
1: initialize reliability buffer $\mathcal{B} \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ **to** N **do**
3: fetch sample $(\mathcal{X}_i, q_i)$ from $\mathcal{D}^{\text{tgt}}$
4: call Algorithm 2 with $(\mathcal{X}_i, q_i, \mathcal{F}_{\text{VL}}, \mathcal{B})$ to obtain $\mathcal{L}_{\text{adapt}}^{(i)}$
5: back-propagate $\mathcal{L}_{\text{adapt}}^{(i)}$ only through the tunable components specified in Sec. 3.2.3
6: optionally apply gradient clipping
7: update parameters via OPT
8: **end for**
9: **return** $\mathcal{F}_{\text{VL}}^{*}$

Thus, q_{ref} is not a generic text-only template, but a `MEM_TOKEN`-aware instruction that (i) binds the predicted region $\mathcal{M}^{\text{ori}}$ to an index `[0]` via special tokens, and (ii) asks the model to output a *referring expression* specifically for that indexed region.

Post-processing $\psi(\cdot)$. Given the raw generated text $\tilde{q}$ from the reverse step, we use a two-stage post-processing pipeline. In the notation of Section 3.2.1, this composite mapping corresponds to $\psi(\cdot)$.

1. **Content cleaning**:
   - Strip leading/trailing whitespace.
   - Remove model-specific special tokens `SEG_TOKEN`, `MEM_TOKEN`, `REF_TOKEN`.
   - Remove region indices such as `[0]`, `[1]` (and an optional following "is"):

     "[0] is a dog" $\rightarrow$ "a dog".

   - Remove common boilerplate prefixes while keeping the core description, e.g.

     "The object in [0] is a dog" $\rightarrow$ "a dog",

     using regex patterns such as

     ```
     ^The object (in question )?(in )?(referred to )?(is )?,
     ^It is,^This is,...
     ```

   The output of this step is a cleaned phrase q_{clean} that aims to preserve all descriptive content while discarding special tokens and meta-phrases.
2. **Segmentation-style rephrasing**: Given q_{clean}, we convert it into a grammatically fluent segmentation instruction q_{seg}:
   - If q_{clean} contains a copular verb ("is/are/was/were"), we split it into subject and predicate and rewrite the predicate into a relative clause (e.g., "dog is running" $\rightarrow$ "the dog that is running").
   - If q_{clean} is a noun phrase without verb, we ensure it starts with an article (insert "the" if missing).
   - Finally, we wrap it into the same task template as the forward question:

     q_{seg} ="Please segment q_{phrase} in this image."

Putting these together, the effective $\psi(\cdot)$ used in the cycle is

$$\psi(\tilde{q}) = \texttt{format_segmentation_question}\big(\texttt{clean_and_format_query}(\tilde{q})\big), \tag{36}$$

and this $q_{\text{seg}} = \psi(\tilde{q})$ is then fed into the segmentation branch to obtain the cycle prediction $\mathcal{M}^{\text{cyc}}$ used for mask consistency and reliability estimation.

D Results with a Larger 7B Vision–Language Backbone

To assess how Cycle-ML scales with model capacity, we further instantiate UniPixel with a larger 7B version of the underlying vision–language model and repeat the ReasonSeg-C experiments under the same test-time adaptation protocol as in Section 4. All architectural components (mask decoder, prompt encoder, object memory bank) and adaptation settings (updated parameter subsets, optimizer, learning rate, LoRA configuration, and reliability modeling) are kept identical, except for replacing the 3B backbone with its 7B counterpart.

Table 5 summarizes the results on the ReasonSeg validation and test splits, reporting gIoU on the clean "Original" setting as well as the average over 15 synthetic corruptions. Consistent with our 3B experiments, Cycle-ML brings clear and robust gains over the source-only model across both splits and metrics. These trends mirror our findings with the 3B backbone and confirm that Cycle-ML remains effective when scaling up model size.

Table 5: ReasonSeg performance (gIoU) of UniPixel with a 7B vision–language backbone.

Split	**Original**		**Average (15 Corruptions)**	
	No Adapt	Cycle-ML	No Adapt	Cycle-ML
Val	60.43	**61.73**	57.14	**59.52**
Test	60.54	**61.15**	56.72	**58.93**

E Effect of Language Consistency Weight

To further examine the role of the text-level cycle consistency term, we fix the mask consistency weight to $\lambda_{\text{mask}} = 1.0$ and vary the text loss weight $\lambda_{\text{text}} \in \{0, 0.01, 0.05, 0.10, 0.50\}$ during test-time adaptation. We report gIoU on the ReasonSeg validation set under two representative corruptions, Gaussian Noise and Contrast, as summarized in Table 6.

Table 6: Effect of the text-level consistency weight λ_{text} on ReasonSeg-val (gIoU) under two corruptions. λ_{mask} is fixed to 1.0.

	λ_{text}				
Corruption	0	0.01	0.05	0.10	0.50
Gaussian Noise	59.03	59.34	59.42	**59.53**	58.17
Contrast	56.63	57.84	**58.27**	56.98	55.20

From Table 6, we see that adding a small amount of text-level consistency on top of mask consistency is consistently beneficial. Compared to the mask-only variant ($\lambda_{\text{text}} = 0$), introducing a non-zero text weight ($\lambda_{\text{text}} = 0.01$) already improves robustness under both corruptions, and further increasing it to 0.05 brings the best overall performance. For Gaussian Noise, performance increases monotonically from 59.03 to 59.53 as λ_{text} grows from 0 to 0.10, before dropping to 58.17 at $\lambda_{\text{text}} = 0.50$. Under Contrast, the trend is even clearer: gIoU rises from 56.63 at $\lambda_{\text{text}} = 0$ to a peak of 58.27 at $\lambda_{\text{text}} = 0.05$, and then degrades to 56.98 and 55.20 when the text loss is overemphasized ($\lambda_{\text{text}} = 0.10$ and 0.50). These results indicate that a moderate text loss weight provides complementary semantic regularization to the mask-level cycle, improving robustness across different corruptions, while overly large λ_{text} can hurt performance, likely because noisy or unstable language predictions under distribution shift start to dominate the adaptation objective. Overall, in our setup, a relatively small text loss weight around $\lambda_{\text{text}} = 0.05$ offers a good balance between leveraging language-level consistency and preserving the stabilizing effect of mask supervision.

F Additional Discussion on the Choice of TTA Baselines

In this work, the main experimental comparisons are conducted with Pseudo-labeling, TENT, CoTTA, and SVDP. These methods are selected not merely by publication recency, but because

they represent several representative and complementary test-time adaptation paradigms under a shared and controlled evaluation setting. Specifically, Pseudo-labeling represents **self-training**, TENT represents **entropy minimization**, CoTTA represents **continual test-time adaptation**, and SVDP represents a more recent **dense-prediction-oriented adaptation** strategy. All baselines are implemented on top of the same UniPixel backbone and evaluated under the same test-time protocol, so that the comparison focuses on the adaptation mechanism itself rather than differences in model architecture or auxiliary training schemes.

It is important to note that the setting studied in this paper—*test-time adaptation for reasoning/referring segmentation*—differs substantially from standard TTA settings such as closed-set image classification or vision-only semantic segmentation. Under our setting, the model must jointly handle *open-vocabulary language queries, cross-modal alignment, reasoning, and mask prediction* under distribution shift. As a result, many recent TTA methods are not directly applicable without substantial reformulation.

In particular, a large portion of recent TTA approaches are designed for *closed-set classification* Han et al. (2025); Lee et al. (2025); Hu et al. (2025); Liu et al. (2026); Yuan et al. (2023); Schirmer et al. (2024), where adaptation is performed over a fixed label space with class-wise calibration, prototype adjustment, or category-level refinement. Such assumptions do not hold in reasoning segmentation, where the prediction target is not a predefined class label but a *query-conditioned mask* grounded through multimodal reasoning. Similarly, several recent VLM- or CLIP-based TTA methods Guan et al. (2026); Karmanov et al. (2024); Peng et al. (2026); A Vargas Hakim et al. (2025); Osowiechi et al. (2024); Noori et al. (2026) rely on prompts, predefined vocabularies, prototypes, or cache mechanisms, which are difficult to transfer directly to open-ended referring and reasoning segmentation. Even for recent dense-prediction adaptation methods Yang et al. (2024); Lei & Pernkopf (2025); Yi et al. (2024), including SVDP, the primary focus remains *vision-only prediction under visual shift*, rather than the degradation of the *language–reasoning–mask grounding chain* that is central to reasoning segmentation.

For this reason, our goal is not to compare only with earlier methods, but to benchmark **representative adaptation mechanisms** in a fair and controlled setting. The chosen baselines together cover the major adaptation strategies most relevant to the TTA literature and provide informative reference points for understanding the contribution of Cycle-ML. Compared with these paradigms, the key difference of our method is that it explicitly exploits the *bidirectional cross-modal structure* of reasoning segmentation by constructing a mask–language cycle, rather than treating segmentation as a one-way prediction problem.

More broadly, these observations highlight an important distinction between reasoning segmentation and conventional TTA benchmarks: in our setting, robust adaptation requires not only improving visual prediction under corruption, but also preserving and correcting the alignment among *language, reasoning, and spatial grounding*. We therefore view the development of more directly comparable multimodal TTA baselines as an important direction for future work.

G Additional Discussion on CoTTA and SVDP on RefCOCO Benchmarks

We provide additional discussion on the RefCOCO-related comparisons for CoTTA and SVDP, since these methods were not included as extensively as other baselines across all RefCOCO benchmarks in the main paper. Their omission is **not due to selective reporting**, but mainly due to **computational and practical considerations** in our setting.

Specifically, our task involves *test-time adaptation for multimodal generative reasoning/referring segmentation*, which is considerably more expensive than standard classification or vision-only segmentation. For **CoTTA**, the adaptation procedure requires **EMA updates, stochastic restoration, and repeated test-time augmentation**, making evaluation substantially heavier when conducted over multiple RefCOCO splits. For **SVDP**, although it is a representative recent method for dense-prediction test-time adaptation, applying it in our multimodal setting still introduces additional adaptation overhead and implementation complexity beyond the main comparison pipeline. To keep the main paper focused on stable, reproducible, and controlled comparisons under a unified computational budget, we prioritized the most essential benchmark settings in the main text.

Table 7: Additional CoTTA results on RefCOCO and RefCOCO+ under representative corruptions.

Corruption	Method	RefCOCO (val/testA/testB)	RefCOCO+ (val/testA/testB)
Impulse Noise	No Adapt	73.24 / 77.02 / 72.08	69.16 / 71.49 / 65.26
	CoTTA	72.34 / 77.21 / 74.35	68.52 / 72.64 / 66.21
	Cycle-ML	**74.54 / 77.65 / 76.29**	**72.19 / 74.57 / 68.10**
Zoom Blur	No Adapt	59.13 / 58.41 / 58.40	59.04 / 59.25 / 56.66
	CoTTA	60.22 / 59.24 / 60.14	60.41 / 60.47 / 57.76
	Cycle-ML	**61.25 / 61.36 / 62.70**	**61.08 / 62.79 / 60.12**

To further address this concern, we additionally report **CoTTA** results on **RefCOCO** and **RefCOCO+** under representative corruptions in Table 7. The results are consistent with our observations on ReasonSeg: CoTTA improves over **No Adapt** in several cases, showing that it is a meaningful continual TTA baseline, while **Cycle-ML** remains stronger overall. We believe this is because generic continual adaptation or dense-prediction-oriented adaptation mainly improves visual robustness, but does not explicitly address the core challenge in our setting, namely the degradation of the *language–reasoning–mask grounding chain* under distribution shift. In contrast, Cycle-ML directly targets this issue through the proposed *mask–language cycle*.

Overall, these observations suggest that, although representative general-purpose and dense-prediction TTA methods remain useful references, **explicit multimodal grounding adaptation** is more effective for reasoning and referring segmentation.

H Additional Analysis of CRL Stability and Reliability Estimation

This section provides additional analysis of the proposed Consistency-Reliability Learning (CRL) module from four perspectives: buffer size sensitivity, cold-start behavior, the role of Eq. 11, and robustness under severe shifts.

H.1 Buffer Size Sensitivity

The sliding buffer size K controls the usual responsiveness–stability trade-off: smaller K reacts faster to stream changes but yields noisier clustering, while larger K provides more stable reliability estimates but adapts more slowly. In practice, CRL is relatively insensitive to K because reliability is estimated in a low-dimensional 2D consistency space $h = [C_{\text{mask}}, C_{\text{text}}]^\top$, rather than from high-dimensional features or logits, and because Eq. 12 uses Gaussian soft weighting rather than brittle hard trust.

We provide a sensitivity study in Fig. 5, which evaluates $K \in \{64, 128, 256, 512, 1024, 2048\}$ on ReasonSeg-test under four corruption types. Under mild shifts such as Snow and Fog, performance remains stable across a broad range of K. Under more severe corruptions such as Shot Noise and Zoom Blur, variation is larger, but Cycle-ML remains robust overall and consistently outperforms **No Adapt**. Based on this trade-off, we use $K = 512$ as the default setting.

H.2 Cold-Start and Streaming Behavior

At the beginning of inference, the buffer is only partially populated, so spectral clustering may be unstable if applied too early. To avoid unreliable clustering in this cold-start phase, CRL uses a simple safeguard: when the buffer is too small for meaningful partitioning, or when clustering degenerates, we set the reliability weight to $w = 1.0$. Thus, CRL does not force unstable reliability decisions in the first few steps; reliability estimation becomes active only after the buffer contains enough samples.

We use a sliding rather than cumulative buffer so that outdated statistics are gradually replaced by recent ones, which is better suited to test-time adaptation. Our current evaluation protocol assumes a

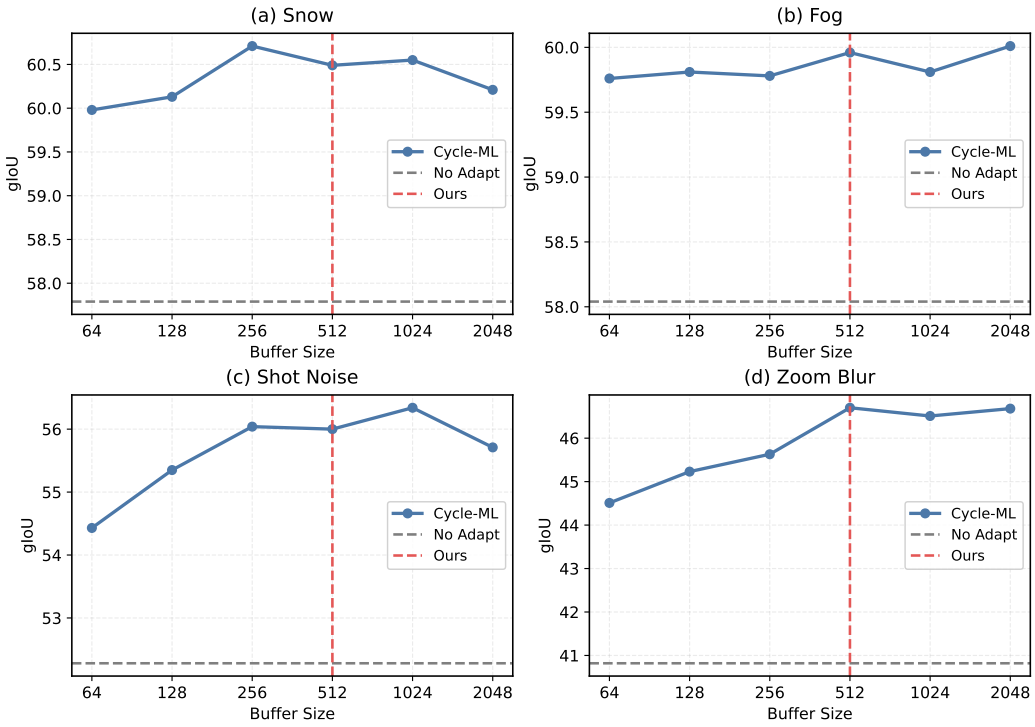


Figure 5: Sensitivity analysis of buffer size K on ReasonSeg-test (3407 samples) under four corruption types. Snow/Fog represent relatively mild shifts, while Shot Noise/Zoom Blur are more severe. Overall, Cycle-ML remains robust across a wide range of K, and consistently outperforms No Adapt. Based on the stability–performance trade-off and the low computational cost of CRL in the 2D consistency space, we use $K = 512$ as the default setting.

stationary stream, where the corruption type is fixed within each run; under this setting, a moderate-to-large buffer is beneficial because it provides stable local statistics. If the domain shift were to change abruptly during inference, a smaller-to-moderate K would generally be preferable for faster adaptation.

H.3 Role and Limitation of Eq. 11

Eq. 11 is not used to rank individual samples solely by mask consistency. In our pipeline, buffered cycles are first clustered in the joint 2D consistency space $h = [C_{\text{mask}}, C_{\text{text}}]^\top$, and Eq. 11 is used only afterward to assign which cluster is treated as relatively reliable. Therefore, Eq. 11 should be interpreted as a cluster-labeling heuristic conditioned on joint cross-modal clustering, rather than a standalone confidence measure.

The rationale is statistical: the less reliable cluster typically contains a larger fraction of inconsistent cycles and therefore tends to have a lower mean mask consistency than the comparatively reliable cluster. This does not imply that every sample in the selected cluster is a true positive, nor does it provide a formal guarantee under arbitrary severe shifts. Instead, Eq. 11 identifies the comparatively better subset within the current buffer.

H.4 Reliability Estimation Under Severe Shifts

Under severe shifts, reverse-generated text may become partially nonsensical or hallucinatory, and the model may also produce confident but self-consistent segmentation errors. In this regime, CRL should be viewed as a *relative* reliability estimator rather than an oracle: if most buffered cycles are poor, the selected cluster may still be only less unreliable than the rest. Therefore, blindly enforcing cycle consistency on all samples can indeed be harmful.

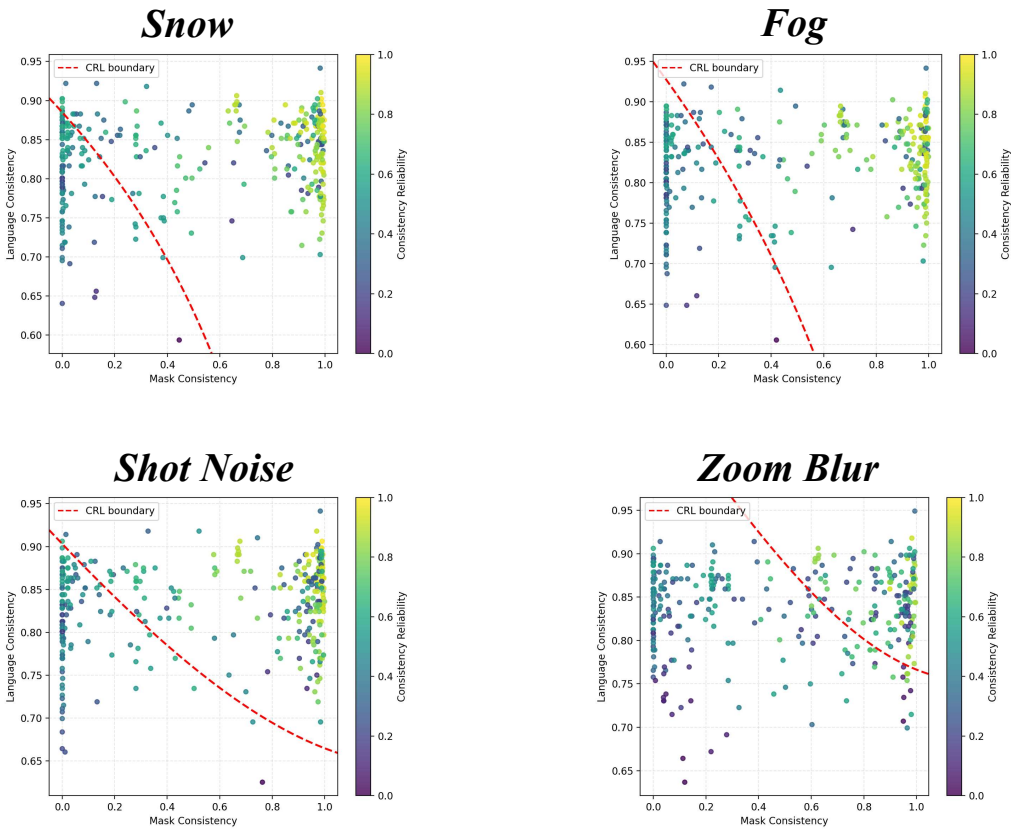


Figure 6: **Visualization of CRL under different corruption severities**. We show the clustering results of buffered cycles under two relatively mild corruptions (*Snow*, *Fog*) and two more severe corruptions (*Shot Noise*, *Zoom Blur*). In each case, CRL identifies a cluster of comparatively reliable cycles based on the joint consistency of the mask and language signals. Under milder shifts, the reliable cluster is larger and more compact, indicating that more buffered cycles remain trustworthy. Under more severe shifts, the reliable cluster becomes smaller and more selective, suggesting that CRL adaptively localizes the relatively reliable subset according to shift severity and reduces the influence of noisy cycles.

Our design mitigates this issue in three ways. First, reliability is estimated jointly from mask and language consistency, reducing the chance of assigning high confidence based on only one modality. Second, after the comparatively reliable cluster is selected, Eq. 12 further applies Gaussian soft weighting, so ambiguous samples receive smaller updates. Third, the language consistency term is intentionally down-weighted in the overall objective, preventing noisy reverse-generated text from dominating adaptation. As shown in Tab. 3 and Sec. E, a small text-consistency weight is beneficial, while over-emphasizing it degrades performance.

This behavior is supported by both quantitative and qualitative evidence. Tab. 4 shows that blindly applying cycle learning is harmful ($57.09 \rightarrow 51.98$ gIoU), while hard reliability assignment improves performance to 58.97, and Gaussian soft weighting further improves it to 59.77. Fig. 6 further shows that under mild shifts the reliable cluster is larger and more compact, whereas under severe shifts it becomes smaller and more selective. This suggests that CRL adaptively localizes the comparatively reliable subset of the buffer and reduces the influence of noisy cycles, rather than uniformly trusting all self-consistent predictions.

I Failure Mode When Most Buffered Cycles Are Unreliable

A particularly challenging failure case arises when the sliding buffer is dominated by poor cycles under a sufficiently severe domain shift. In this regime, CRL can at best identify a *relatively* more reliable subset of samples within the current buffer, rather than guarantee that the selected cluster is truly correct. Therefore, CRL should not be interpreted as an oracle correctness estimator under arbitrary severe shifts.

This limitation is not unique to Cycle-ML, but reflects a broader challenge in unsupervised test-time adaptation. Methods based on entropy minimization, pseudo-labeling, or self-training also rely on model-induced regularization or pseudo-supervision on shifted target samples, and can similarly reinforce incorrect predictions when the current model is already biased. In this sense, the relevant question is not whether Cycle-ML completely removes this risk, but whether it handles this common TTA failure mode more explicitly and conservatively than standard objectives.

This consideration motivates the design of CRL. Rather than adapting on all cycle-induced signals equally, CRL estimates relative reliability from the joint mask–language consistency space and applies Gaussian soft weighting so that weak or ambiguous cycles contribute less to the update. The purpose of CRL is therefore not to certify correctness, but to reduce the error-amplifying effect of naive cycle consistency in difficult regimes.

Our current results provide supportive, though not fully decisive, evidence for this interpretation. First, Tab. 4 shows that directly applying cycle consistency without reliability modeling is harmful and performs below **No Adapt**, whereas CRL-based hard assignment and especially Gaussian soft weighting recover and improve performance. This indicates that reliability-aware weighting suppresses noisy cycles more effectively than naive cycle optimization. Second, Fig. 6 shows that under more severe corruptions, the selected reliable cluster becomes smaller and more selective, rather than remaining artificially large, which is consistent with a conservative reliability mechanism. Third, Fig. 5 shows that even under stronger corruptions, moderate-to-large buffers remain beneficial overall, although the variance becomes larger, suggesting that the problem becomes harder rather than the method collapsing uniformly. Finally, the additional cross-domain result on SegEarth-R1 further suggests that the gains of Cycle-ML are not limited to a single synthetic corruption protocol.

At the same time, these results do not yet constitute a decisive validation for the extreme case in which most buffered cycles are unreliable. A more direct analysis would require explicitly relating adaptation behavior to the global reliability of the current buffer, for example through the average consistency level, the resulting CRL weights, and the point at which adaptation should become increasingly conservative. We leave this more failure-oriented analysis to future work.

Overall, the evidence suggests that Cycle-ML does not eliminate the fundamental risk of self-reinforcing errors under severe shifts, but it does address this issue more explicitly than naive cycle consistency by introducing relative reliability estimation and conservative weighting.

J Synthetic Corruptions versus Real-World Distribution Shifts

Synthetic corruptions do not fully capture the complexity of real-world distribution shifts, such as sensor-specific noise, illumination changes, or other modality-dependent artifacts. In this work, we adopt the 15 standard corruption types because they provide a controlled and reproducible benchmark for test-time adaptation, enabling fair comparison with prior methods.

To further examine robustness beyond synthetic corruptions, we also evaluate our method under a real cross-domain shift. Specifically, we investigate multimodal reasoning-segmentation datasets from other domains, including SegEarth-R1 for remote sensing Li et al. (2025) and MedSeg-R for medical imaging Huang et al. (2025). Since MedSeg-R is not publicly available, we report cross-domain transfer results on SegEarth-R1 by directly applying UniPixel without further training.

As shown in Tab. 8, our method improves over the non-adaptive baseline by 3.68 gIoU on the validation split and 3.96 gIoU on the test split. These results suggest that Cycle-ML also improves robustness under a real cross-domain shift, complementing the synthetic corruption benchmark used in the main paper.

Table 8: Cross-domain results (gIoU) on SegEarth-R1.

Method	Val	Test
No Adapt	29.11	27.47
Cycle-ML	**32.79**	**31.43**

K Inference Latency Discussion

Although Cycle-ML introduces additional computation for adaptation, it does not directly increase the inference latency of the current prediction. In our pipeline, the current sample is first processed by the current model to produce the output, and the segment→generate→segment cycle is then used as an adaptation step that mainly benefits subsequent samples in the stream. Therefore, the cycle update is not on the critical path of the prediction returned for the current sample.

The additional computation is thus associated primarily with adaptation cost rather than the blocking latency of inference itself. Moreover, CRL adds negligible overhead, since it operates only on a 2D consistency feature and its spectral partitioning step is performed outside the gradient graph. During adaptation, we also update only lightweight components, including the language–mask projectors, mask decoder, and LoRA adapters, while keeping the rest of the vision-language backbone frozen.

In this sense, Cycle-ML mainly affects overall computational budget or throughput, rather than directly delaying the prediction for the current input. This makes it more suitable for robustness-sensitive streaming settings than for scenarios with extremely strict hard real-time constraints.

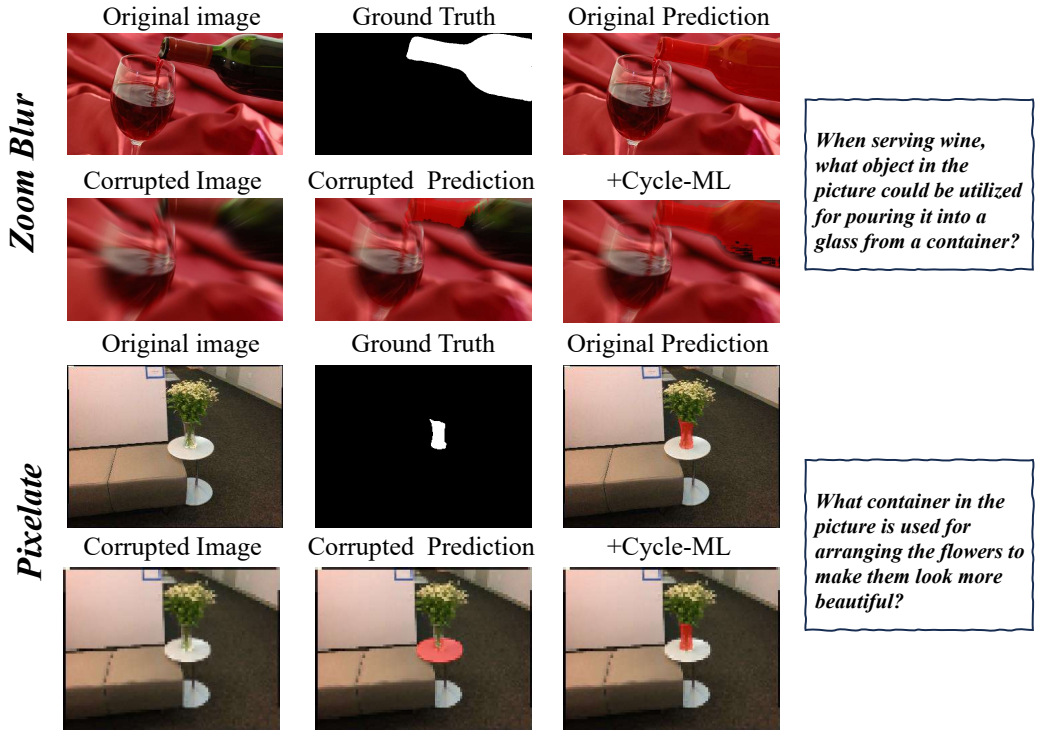


Figure 7: Qualitative results on ReasonSeg-val under Zoom Blur and Pixelate corruptions.

L Qualitative Results

To complement the quantitative results in the main paper, we provide additional qualitative examples on ReasonSeg-val under image corruptions. Figure 7 visualizes two representative cases with *Zoom*

Blur and *Pixelate* corruptions. For each case, we show the original image, ground-truth mask, and the prediction from the source model ("Original Prediction"), as well as the corrupted image and predictions before and after applying our test-time adaptation method ("Corrupted Prediction" vs. "+Cycle-ML"). The corresponding referring questions are also displayed on the right.

As shown in Figure 7, zoom blur and pixelation significantly degrade the source model's predictions: the masks on corrupted images become misaligned with the intended objects, often shrinking to background regions or drifting to visually salient but semantically irrelevant areas. After applying Cycle-ML, the predicted masks on corrupted images are noticeably closer to the ground truth, better capturing the correct wine-serving container in the first example and the flower-arranging container in the second example. These results qualitatively demonstrate that our cycle-based test-time adaptation more robustly preserves fine-grained grounding between language instructions and visual regions under distribution shifts.

M Ethics Statement

This paper studies test-time adaptive reasoning segmentation, a methodological problem in multimodal language understanding and machine learning. The proposed approach is designed to improve robustness under distribution shifts and does not introduce any apparent new ethical concerns beyond those already associated with vision–language models. However, like other adaptive multimodal systems, it may still inherit biases from pretrained models, produce erroneous segmentations, or generate unreliable textual feedback under challenging conditions. Therefore, the method should not be deployed in high-stakes or safety-critical settings without careful evaluation and appropriate human oversight.