XF-CLS: Cross-Foundation Complementary Learning Systems for Continual Test-Time Adaptation in Open-Vocabulary Semantic Segmentation

Guan-Yu Chen[1] **Wei-Hua Li**[2] **Huei-Fang Yang**[3*] **Chu-Song Chen**[1,2*]
[1]Data Science Degree Program, National Taiwan University and Academia Sinica, Taiwan
[2]Department of Computer Science and Information Engineering, National Taiwan University
[3]Department of Computer Science and Engineering, National Sun Yat-sen University
*Corresponding Authors

Abstract

Open-vocabulary semantic segmentation (OVSS) labels each pixel with classes named in natural language, but deployed models face changing conditions such as fog, night, rain, and snow. Continual test-time adaptation (CTTA) can adapt them online without labels, yet over long streams without resets, unsupervised updates accumulate: predictions can degrade or collapse, and updated visual features can lose alignment with the frozen text embeddings that support open-vocabulary recognition. We propose Cross-Foundation Complementary Learning Systems (XF-CLS), a source-free OVSS-CTTA framework that separates rapid adaptation from slow consolidation. *Hippocampal Adaptation* (fast path) updates only visual normalization parameters on each incoming image, while *Neocortical Consolidation* (slow path) keeps a slowly updated, source-anchored memory and pulls the fast parameters back toward it to counter drift, with a finite-horizon analysis guiding the memory's update rate. A third path uses a frozen DINOv2 model as a non-drifting structural observer to guide recovery and refine dense predictions, without training another teacher or supervising updates with pseudo-labels. On long-horizon Cityscapes-to-ACDC and OnDA streams, XF-CLS reaches 32.92 and 30.75 mIoU, outperforming all compared methods, and retains 99.6% and 98.8% of source mIoU on VOC classes outside the adaptation vocabulary. Matched controls identify source-anchored recovery as the main stabilizer and frozen refinement as complementary output correction. These gains come with additional computation from online updates and structural inference. Our code is available at `https://github.com/Gary20011207/XF-CLS`.

1 Introduction

Open-vocabulary semantic segmentation (OVSS) labels every pixel with categories named in natural language, transferring image–text alignment from vision-language foundation models such as CLIP [36] to dense, pixel-level prediction. This flexibility suits open-world systems such as autonomous driving and robotics, yet deployment rarely stays in one condition: a vehicle may pass from daylight into fog, night, rain, or snow within a single trip, and OVSS predictions can degrade even when the vocabulary is unchanged.

Test-time adaptation (TTA) counters such shifts by updating the model on unlabeled test images during inference. Early OVSS-TTA evaluations primarily follow episodic protocols, adapting to each batch before resetting to the source weights [31]. In persistent deployment, however, the model faces one long stream without resets, so every update is carried forward and the stability of accumulated updates becomes essential.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

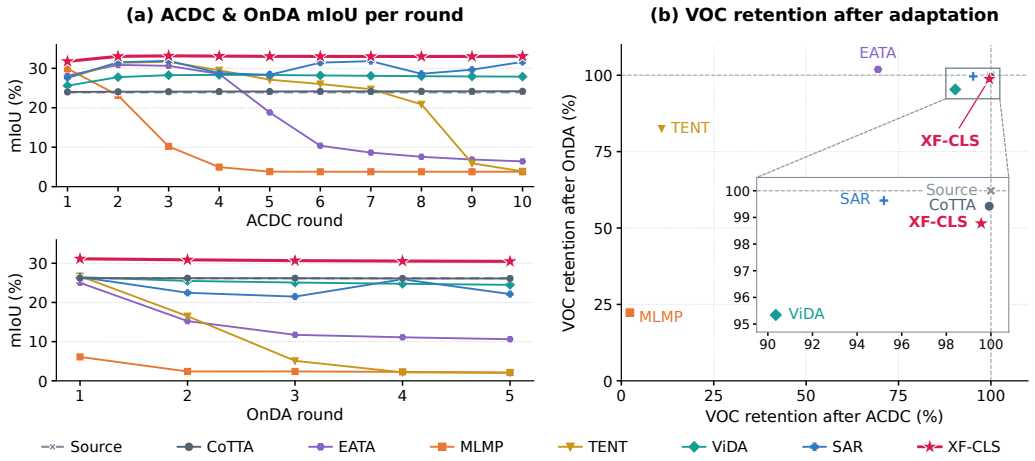

Figure 1: **(a) Continual adaptation.** Per-round mIoU on ACDC and OnDA shows that several baselines degrade or become unstable over the streams, while XF-CLS sustains adaptation gains. **(b) Open-vocabulary retention.** After each stream, we evaluate the 14 VOC classes outside the Cityscapes adaptation vocabulary without further updates. Retention is their mIoU relative to the source model (100%); values above 100% indicate improvement over source. Both XF-CLS and CoTTA retain near-source performance, but only XF-CLS also raises the driving-stream mIoU. Both panels show means over three stream orders; some baseline settings differ from the single-run rows of Tables 1–2 (Appendix I.2).

We therefore study **continual test-time adaptation for open-vocabulary semantic segmentation** (OVSS-CTTA): the model predicts and updates online on each incoming image, without source data, online target labels, or externally supplied boundary resets. Two risks make this setting hard. First, repeated unsupervised updates can amplify early mistakes into cumulative drift and even collapse [26]; Figure 1(a) shows several baselines degrading. Second, adaptation can undermine open-vocabulary recognition as updated visual features drift from the frozen text embeddings; Figure 1(b) measures post-stream retention on VOC classes outside the Cityscapes adaptation vocabulary. A useful method must keep improving on the stream while retaining what it could already recognize.

We propose **Cross-Foundation Complementary Learning Systems (XF-CLS)**, a source-free framework for stable OVSS-CTTA. Drawing on the fast/slow distinction in Complementary Learning Systems [28, 2, 1] as a design analogy, XF-CLS separates rapid adaptation from slow consolidation. *Hippocampal Adaptation* (fast path) adjusts a small set of visual normalization parameters to each incoming image, while *Neocortical Consolidation* (slow path) keeps a slowly updated memory anchored to the source model and pulls the fast parameters back toward it to counter drift.

A third, cross-foundation path uses a frozen DINOv2 model [32] as a non-drifting structural observer: its features guide recovery and refine dense predictions, without training another teacher or supervising the updates with pseudo-labels. As the observer is frozen, XF-CLS updates only a small set of parameters, although online updates and structural inference add compute (Section 4.5).

On long-horizon Cityscapes-to-ACDC [37] and OnDA streams [34], XF-CLS reaches 32.92 and 30.75 mIoU under the evaluated protocols. Our contributions are summarized as follows:

- **Setting.** We formulate and study **OVSS-CTTA**, extending open-vocabulary semantic segmentation adaptation from episodic reset-based inference to long-horizon non-stationary streams without source data, online target labels, or externally supplied boundary resets.
- **Method.** We propose XF-CLS, which couples fast *Hippocampal Adaptation*, source-anchored *Neocortical Consolidation*, and a frozen cross-foundation structural path in a parameter-efficient framework; a finite-horizon analysis guides the slow path's update rate.
- **Evidence.** XF-CLS outperforms the evaluated OVSS-CTTA ports on both streams while retaining near-source performance on VOC classes outside the adaptation vocabulary. Matched controls isolate recovery from output refinement, with synonym-prompt and recurring VOC stability diagnostics.

2 Related Work

Continual learning (CL) studies how a model can keep acquiring knowledge from non-stationary data while mitigating forgetting. It was traditionally devoted to image classification [18, 46] and was later extended to image segmentation [4, 7] and visual search [43]. Because labels are costly to obtain, weak supervision [17, 5, 22] has also been studied. Continual visual search, for example, keeps updated embeddings backward-compatible with an indexed gallery, much as OVSS must preserve its frozen text interface.

With the rapid progress of vision-language models, recent advances in CL have increasingly leveraged foundation models and moved toward TTA. The above works pursue the same continual improvement in supervised or weakly supervised settings, whereas we adapt using only unlabeled target streams, without access to source data.

Test-Time and Continual Test-Time Adaptation. Test-time adaptation (TTA) addresses this setting by updating a pre-trained model on unlabeled target data at inference. Since TENT's entropy minimization [44], methods improve stability via reliable sample filtering, sharpness-aware optimization, teacher–student consistency, stochastic restoration, and memory buffering [29, 30, 47, 51, 12]. Continual TTA (CTTA) extends TTA to non-stationary streams, where unreliable unsupervised updates can accumulate errors and induce catastrophic forgetting, as in CL [26]; CoTTA [47], RMT [12], RoTTA [51], and ViDA [24] counter these effects with restoration, mean-teacher stabilization, or visual domain adapters. These methods, however, operate on fixed, closed-set label spaces, whereas an adapted OVSS model must also stay compatible with frozen text embeddings.

Vision-Language Foundation Models and OVSS. These embeddings come from vision-language foundation models such as CLIP [36]. OVSS transfers CLIP-style image–text alignment to dense pixel-to-text prediction through mask adaptation, modified attention, dense feature extraction, or training-free refinement [23, 53, 11, 45, 3, 15]; among them, the training-free NACLIP [15] serves as our adaptable OVSS model. CLIP has also motivated TTA for zero-shot recognition: TPT [38] adapts prompts by entropy minimization over augmented views, while WATT [33], CLIPArTT [42], and related methods adapt CLIP through prompt, weight, or pseudo-label updates. Yet these TTA methods target image-level recognition, while the OVSS methods above remain static at test time.

From Episodic OVSS-TTA to OVSS-CTTA. OVSS-TTA brings these two lines together. MLMP [31] shows that dense vision-language segmentation benefits from episodic TTA with multi-level visual features and multi-prompt entropy minimization; its Appendix J also reports an online comparison in which carrying updates reduces V20-C mIoU from 77.58 to 71.77. Episodic resets remove cumulative drift by construction; we instead study OVSS-CTTA, which carries parameters forward and asks whether useful adaptation survives, and our MLMP port shows that a strong episodic objective can collapse when naively carried continually. Seg-TTO, SemLA, and ConOVS follow different protocols: per-input visual and text optimization, retrieval and fusion of a prebuilt LoRA library, and sequential training and routing of expert decoders, respectively [10, 35, 19]. The concurrent emergence of DAF and CARE [13, 25], together with recent preprints on persistent-state diagnosis and recovery [48, 49], indicates that continual OVSS adaptation is a timely and open problem; we focus on keeping a persistent OVSS state stable over long, non-stationary streams.

Cross-Foundation Priors and Complementary Learning Systems. XF-CLS builds on two strands of prior work. First, pairing CLIP with another foundation model improves static, single-image OVSS: ProxyCLIP and CLIP-DINOiser refine dense CLIP predictions with spatial features from models such as DINO [21, 50]. Second, Complementary Learning Systems (CLS) theory motivates fast and slow learning to limit catastrophic interference [28], an idea explored in continual learning [1] and TTA [20, 39]. Source anchoring and refinement have TTA precedents: CoTTA uses stochastic source restoration [47], ROID ensembles source and adapted weights [27], and AdaContrast and TeSLA refine or stabilize test-time learning [6, 40]. On this basis, XF-CLS couples source-anchored slow recovery with a frozen cross-foundation structural path to stabilize persistent OVSS adaptation. Matched controls and a finite-horizon EMA analysis support these design choices.

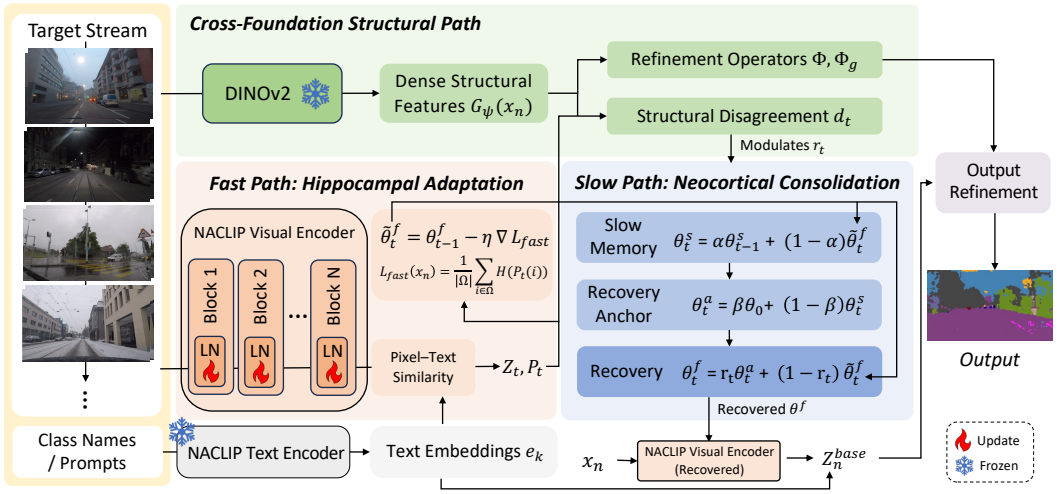

Figure 2: Overview of **XF-CLS**. **Hippocampal Adaptation** (fast path) updates only visual LayerNorm parameters by dense entropy minimization; **Neocortical Consolidation** (slow path) maintains an EMA memory and a source-anchored recovery target; recovery pulls the fast state toward this target. The frozen DINOv2 path compares its structure with the current OVSS prediction to modulate recovery, and refines the final output. The lower visual encoder is the same fast model after recovery. The gradient step is schematic (Adam in implementation); each image receives two inner updates before prediction and the scheduled refinement in Eq. (5).

3 Method

We propose **XF-CLS**, a **Cross-Foundation Complementary Learning Systems** framework for continual test-time adaptation in open-vocabulary semantic segmentation (Figure 2). Rather than designing another per-batch adaptation objective, XF-CLS is designed to keep lightweight OVSS adaptation stable when it is carried over a long non-stationary stream, targeting the two risks raised in the introduction. *Hippocampal Adaptation* (fast path) updates only visual LayerNorm parameters, *Neocortical Consolidation* (slow path) maintains a source-anchored memory used to pull the fast state back from drift, and a cross-foundation structural path uses frozen DINOv2 to modulate recovery and refine dense outputs. Matched controls identify source-anchored recovery as the main stabilizer and frozen refinement as complementary output correction; the gates and filters modulate these operations (Appendix I.4).

3.1 Problem Setup: OVSS-CTTA

Let $\mathcal{X} = \{x_1, \ldots, x_N\}$ be an unlabeled image stream and $\mathcal{C} = \{c_1, \ldots, c_K\}$ the class vocabulary supplied at test time. A frozen text encoder maps each class name to an embedding $\mathbf{e}_k$. For an image x and current visual parameters θ, the logits and probabilities are

$$Z(i,k) = s\big(\mathbf{v}_\theta(x,i), \mathbf{e}_k\big), \qquad P(i,k) = \mathrm{softmax}_k\big(Z(i,k)\big), \tag{1}$$

where $i \in \Omega$ indexes spatial locations, $\mathbf{v}_\theta$ is the visual feature, and $s(\cdot,\cdot)$ denotes image–text similarity. In OVSS-CTTA, the model must output a segmentation map and update online using only the unlabeled target stream. No source images or target labels enter online updates, and no external image-, domain-, or round-boundary reset is supplied. Method-internal recovery, including SAR's automatic reset, remains allowed. This no-reset constraint distinguishes OVSS-CTTA from episodic OVSS-TTA: updates that help short-term predictions can accumulate into long-horizon drift.

3.2 Overview of XF-CLS

XF-CLS maintains three states over the online-updated parameters: a fast state θ_t^f, a slow memory θ_t^s, and the fixed source initialization θ_0. Throughout this section, θ denotes only the parameters updated online; in our implementation these are the visual LayerNorm affine parameters of the OVSS image encoder. The text encoder and the remaining vision–language backbone are frozen.

Each image x_n receives $S = 2$ inner updates, each consisting of fast adaptation, slow consolidation, and recovery. The inner-update index t and image index n satisfy $n = \lceil t/S \rceil$:

$$\begin{aligned} \tilde{\theta}_t^f &= \mathrm{Opt}\Big(\theta_{t-1}^f, \nabla_{\theta^f} \mathcal{L}_{\mathrm{fast}}(x_n)\Big), & \theta_t^s &= \alpha \theta_{t-1}^s + (1-\alpha)\tilde{\theta}_t^f, \\ \theta_t^a &= (1-\beta)\theta_t^s + \beta \theta_0, & \theta_t^f &= (1-r_t)\tilde{\theta}_t^f + r_t \theta_t^a. \end{aligned} \tag{2}$$

Here Opt is Adam with carried optimizer state, $\beta = 0.5$, and $\theta_0^f = \theta_0^s = \theta_0$. The recovery rate r_t is defined in Section 3.4. In words, the fast state learns from the current image, the slow memory absorbs these changes only gradually, and recovery pulls the fast state toward a source-anchored blend of the slow memory. If the entropy filter skips the gradient step, $\tilde{\theta}_t^f = \theta_{t-1}^f$; consolidation and recovery still execute. A periodic source pullback can adjust the slow memory after its EMA update and before forming the anchor. After both inner updates, the recovered fast model predicts the current image, followed by frozen structural refinement (*adapt-then-predict*).

3.3 Fast Path: Hippocampal Adaptation

The fast path provides rapid online plasticity from the current target image. Rather than adapting the full vision–language model, it updates only the visual LayerNorm affine parameters, a small parameter subset that lets dense visual features adjust while the text encoder and all other backbone parameters remain frozen. At inner update t, let Z_t and P_t denote the logits and probabilities of θ_{t-1}^f on the current image x_n (Eq. (1)). The objective is uniform dense entropy:

$$\mathcal{L}_{\mathrm{fast}}(x_n) = \frac{1}{|\Omega|} \sum_{i \in \Omega} H(P_t(i)), \qquad H(P_t(i)) = -\sum_{k=1}^{K} P_t(i,k) \log P_t(i,k). \tag{3}$$

This keeps the learning signal simple: no target pseudo-labels, source replay, or cross-foundation distillation enter the loss, and agreement and disagreement signals control recovery rather than weighting pixels in the entropy objective. The entropy filter skips a gradient step when mean entropy, normalized by $\log K$, exceeds 1.3 times its running mean. It mainly suppresses startup updates (first accepted updates at images $64/66$ on ACDC/OnDA; Appendix I.4); the slow path continues consolidation and recovery throughout.

3.4 Slow Path: Neocortical Consolidation

The slow path averages fast-state updates and blends this memory with the fixed initialization to form the recovery anchor in Eq. (2). Its EMA is deliberately much slower than a standard teacher EMA, so the memory does not chase individual images; the explicit source component keeps the anchor tied to the initial model. A periodic drift check additionally blends the slow state halfway toward source when its relative drift exceeds 0.30 (Appendix C). This discrete safeguard does not bound drift continuously; it complements the continuous recovery in Eq. (2).

Agreement and structural disagreement are measured before the fast gradient step. They modulate the recovery rate:

$$r_t \;=\; r_0\,(1-a_t)\,h_t\,\mathrm{clamp}\,\big(1 + d_t - \bar{d}_t,\; g_{\min},\; g_{\max}\big), \tag{4}$$

where $a_t \in [0,1]$ averages the fraction of encoder layers agreeing with the modal class at each location. The disagreement d_t is defined in Section 3.5, and $\bar{d}_t$ is its running mean. The trend factor $h_t = \mathrm{clip}(e^{-50\Delta a_t}, 0.1, 5)$ uses the difference Δa_t between the latest and preceding ten-update mean agreements (zero before 20 updates), increasing recovery when agreement declines. The clamped term is centered at 1: typical disagreement keeps recovery close to the baseline $r_0(1-a_t)h_t$, disagreement above its running mean strengthens recovery, and lower disagreement relaxes it, within $[g_{\min}, g_{\max}]$. With the cross-foundation path disabled (*XF-CLS w/o DINO*), the gate is fixed to 1, giving $r_t = r_0(1-a_t)h_t$; source-anchored recovery remains active.

3.5 Cross-Foundation Structural Path

A slow memory derived from the same OVSS model can delay drift, but it cannot provide an independent view of the image. XF-CLS therefore uses a frozen heterogeneous foundation model G_ψ as a cross-foundation structural observer. In our implementation, the adaptable OVSS model is instantiated with NACLIP, while G_ψ is instantiated with frozen DINOv2. The frozen model is never optimized during CTTA and is not used as a pseudo-label teacher.

Recovery gating. We resize the current OVSS logits to the DINO grid and apply softmax. These probabilities form weighted class prototypes from the frozen features; each grid location is then classified by similarity to the prototypes. The disagreement d_t is the fraction of locations where this reconstruction differs from the original OVSS prediction. This circular-consistency check measures structural agreement rather than semantic correctness. Its gate is auxiliary: fixing it to one changes All Mean by less than 0.02 mIoU in our controls (Appendix I.4); output refinement provides the larger measured contribution of the frozen path.

Output refinement. The frozen path also refines the dense output of the recovered OVSS state. A full refinement operator Φ combines three components in G_ψ's feature space: top-k neighbor smoothing, prototype consensus (recomputing prototypes from the recovered state, iteratively refining them and applying an absent-class prior), and a per-pixel uncertainty/disagreement gate that controls how strongly structural cues override the OVSS prediction. A simpler conservative refinement Φ_g blends the base logits with their top-k smoothed values. An image-indexed weight blends the two refined outputs:

$$\begin{aligned} Z_n^{\text{out}} &= (1 - w_n)\, \Phi_g(Z_n^{\text{base}}, G_\psi(x_n)) + w_n\, \Phi(Z_n^{\text{base}}, G_\psi(x_n)), \\ w_n &= \max\big(w_{\min},\, e^{-(n-1)/\tau_{\text{xfr}}}\big). \end{aligned} \tag{5}$$

where Z_n^{base} is recomputed from the recovered fast state θ_{Sn}^f after both updates on image x_n. The full operator receives weight 1 at the start of the stream; as the image index grows, the output relies increasingly on the conservative branch, while the floor $w_{\min} > 0$ keeps a contribution from Φ. The schedule depends only on the image index, not on an estimate of prediction quality. Both branches retain frozen structural information. No gradient propagates through G_ψ, Φ, or Φ_g into the online update: the frozen path checks and refines, but never becomes another drifting teacher.

3.6 Horizon-Aware Consolidation

The EMA rate controls how quickly the slow memory moves away from its initialization. For the EMA recurrence alone, with a fixed fast-state displacement and $0 < \alpha < 1$, the absorbed fraction is (Appendix B)

$$q_T = 1 - \alpha^T, \qquad q_T \le \epsilon \iff \alpha \ge (1 - \epsilon)^{1/T}, \qquad \tau_{\text{sat}} \approx \frac{1}{1 - \alpha}, \tag{6}$$

for $0 < \epsilon < 1$ and $T = SN$ inner EMA updates. The time constant τ_{sat} marks a $1/e$ residual initialization weight rather than complete saturation. With $\alpha = 0.99999$, it is about 10^5 updates; over the $32{,}000/25{,}000$ updates in ACDC/OnDA, $q_T \approx 0.274/0.221$, so the initialization keeps most of its weight in the slow memory, which can therefore serve as a recovery reference. This motivates the slow rate over the evaluated horizons; the analysis concerns the EMA recurrence alone and does not bound prediction error or establish stability of the coupled system.

3.7 Final Instantiation

Our main experiments instantiate XF-CLS with NACLIP [15] ViT-L/14 as the adaptable OVSS backbone and frozen DINOv2 [32] ViT-L/14 as the cross-foundation structural model. Only the visual LayerNorm affine parameters are updated online, yielding $102{,}400$ updated parameters; optimization uses Adam. The slow path uses $\alpha = 0.99999$ ($\tau_{\text{sat}} \approx 10^5$ steps), anchor blend $\beta = 0.5$, drift-budget pullback at relative drift 0.30, and gate bounds $(g_{\min}, g_{\max}) = (0.5, 1.5)$. The base recovery rate is $r_0 = 0.001$ on ACDC and $r_0 = 0.002$ on OnDA; all other hyperparameters are shared across benchmarks. The cross-foundation refinement (XFR) module uses DINO-neighbor smoothing, prototype consensus, an absent-class prior, and uncertainty/disagreement routing with horizon-aware annealing; exact hyperparameters are provided in Appendix C. For each image, we execute the inner update of Eq. (2) for $S = 2$ iterations and then evaluate; the final segmentation is computed from θ_{Sn}^f via Eq. (5) (*adapt-then-predict*).

4 Experiments

We evaluate whether XF-CLS sustains adaptation gains over long, recurring streams (Section 4.2) and preserves recognition beyond its adaptation vocabulary (Section 4.3), and then analyze the roles and costs of its three paths (Sections 4.4–4.5).

4.1 Experimental Setup

All adaptive methods share the NACLIP backbone and the same adapt-then-predict protocol, with no source images, no online target labels, and no externally supplied boundary resets. We port the comparison methods to this OVSS protocol; their update rules and tuning budgets differ (Appendix I). Online updates are label-free, whereas recovery rates and refinement strengths were selected offline using labeled benchmark performance, so model selection is target-informed [52, 9].

Benchmarks. Both driving streams use the Cityscapes-19 vocabulary [8]. **Cityscapes-to-ACDC** cycles through fog, night, rain, and snow [37] for ten rounds (16,000 images). **OnDA** [34] uses Cityscapes validation images rendered with physics-based rain [16, 41] at 25, 50, 75, 100, and 200 mm/h, repeated for five rounds (12,500 images). Both are controlled streams in which every shift recurs, stressing long-horizon stability. We additionally use VOC2012 [14] to test recognition beyond the driving vocabulary and stability on a non-driving stream.

Evaluation. All Mean averages the mIoUs of all round–condition pairs with equal weight. Tables 1–2 compare XF-CLS, averaged over three stream orders that permute images within each condition, with single runs of the baselines. Standard deviations, where reported, are taken across these orders with fixed initialization.

4.2 Continual Adaptation Performance

Cityscapes-to-ACDC. Table 1 shows that strong early predictions do not guarantee sustained performance. MLMP achieves the highest round-1 fog score but degrades over the rounds to an average of 14.76 mIoU, and EATA also degrades sharply after its early gains. In contrast, XF-CLS improves over the first rounds and holds this level through round 10, achieving the best score in every reported column except round-1 fog. It reaches 32.92 mIoU, 9.00 points above source NACLIP and 2.82 points above SAR, the strongest baseline. Even without DINO, XF-CLS reaches 30.21 mIoU; the complete system improves this further.

OnDA. The recurring rain shifts in Table 2 expose the same failure more severely. TENT drops from 26.75 mIoU in round 1 to 2.03 in round 5, MLMP averages only 3.08, and no comparison method exceeds source NACLIP's 26.18 all-stream mIoU. XF-CLS achieves the best score in every column, reaching 30.75 mIoU; its variant without DINO reaches 27.57. The gains hold at every intensity, including 200 mm/h, where XF-CLS raises round-5 mIoU from 23.87 to 27.50. Matched three-order runs against the strongest baseline on each benchmark lead to the same conclusion: XF-CLS exceeds SAR on ACDC by 2.79 ± 0.12 points and CoTTA on OnDA by 4.59 ± 0.01 (Appendix I). Section 4.4 separates the contributions of online adaptation and structural refinement.

Segmentation quality. Figure 3 shows the corresponding visual differences: XF-CLS keeps road regions, object extents, and boundaries more coherent under adverse weather, whereas MLMP's predictions collapse by round 10. Appendix H gives further ACDC and OnDA examples.

4.3 Open-Vocabulary Preservation

Recognition beyond the driving vocabulary. Because adaptation updates the shared visual encoder, it could also erode recognition of classes never seen during adaptation. We therefore evaluate the final adapted states on the 14 VOC classes absent from the Cityscapes vocabulary, without further updates or structural refinement. XF-CLS retains 99.56% of source mIoU after ACDC and 98.78% after OnDA (55.79 and 55.35 vs. 56.03; Figure 1(b) and Table 17). Together with Figure 1(a), this shows that the driving-stream gains do not come at the expense of recognition outside the adaptation vocabulary. These classes are unseen during adaptation, though not necessarily during foundation-model pretraining.

Table 1: Continual segmentation on Cityscapes-to-ACDC (mIoU, ↑). Selected rounds are shown; All Mean averages all 40 round–condition scores. XF-CLS is averaged over three stream orders; the other rows report single runs.

Round	1				4				7				10				**All**
Condition	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Mean↑**
Source (NACLIP)	24.51	23.94	23.64	23.59	24.51	23.94	23.64	23.59	24.51	23.94	23.64	23.59	24.51	23.94	23.64	23.59	23.92
CoTTA	24.52	23.96	23.74	23.64	24.59	24.08	23.75	24.01	24.89	24.20	23.82	23.84	24.65	24.14	23.81	24.05	24.12
EATA	26.25	26.92	29.27	30.30	31.65	26.90	28.73	27.77	11.38	5.64	7.47	7.37	8.26	4.22	5.30	5.82	17.33
TENT	25.29	24.93	25.89	27.08	33.99	29.47	31.77	31.16	34.51	27.46	30.16	30.00	31.83	24.76	27.73	27.39	29.68
MLMP	**33.18**	27.61	29.58	29.86	27.83	17.88	19.47	16.29	9.73	4.62	5.50	4.99	5.13	2.58	3.39	4.05	14.76
ViDA	25.08	25.03	25.87	27.64	32.88	26.66	27.67	28.18	31.82	25.88	26.90	27.38	31.37	25.43	26.49	26.97	27.97
SAR	26.35	26.04	28.35	29.85	34.50	26.58	29.43	27.37	35.15	29.92	32.06	31.39	33.12	29.31	31.78	31.70	30.10
XF-CLS w/o DINO	25.98	26.39	28.35	29.27	33.08	28.72	30.09	30.34	33.17	28.71	30.13	30.39	33.22	28.70	30.21	30.45	30.21
XF-CLS	32.74	**30.15**	**32.02**	**32.06**	**35.76**	**31.03**	**32.94**	**32.68**	**35.76**	**30.92**	**32.85**	**32.60**	**35.84**	**30.85**	**32.91**	**32.66**	**32.92**

Table 2: Continual segmentation on OnDA (mIoU, ↑). Selected rounds are shown; All Mean averages all 25 round–intensity scores. XF-CLS is averaged over three stream orders; the other rows report single runs.

Round	1						3						5						**All**
Intensity (mm/h)	**25**	**50**	**75**	**100**	**200**	**mean**	**25**	**50**	**75**	**100**	**200**	**mean**	**25**	**50**	**75**	**100**	**200**	**mean**	**Mean↑**
Source (NACLIP)	28.18	26.94	26.21	25.69	23.87	26.18	28.18	26.94	26.21	25.69	23.87	26.18	28.18	26.94	26.21	25.69	23.87	26.18	26.18
CoTTA	28.20	27.01	26.29	25.72	23.83	26.21	28.21	26.98	26.24	25.67	23.80	26.18	28.17	26.89	26.13	25.60	23.76	26.11	26.17
EATA	28.98	29.72	26.52	22.66	17.90	25.16	14.32	12.89	12.16	11.68	10.02	12.21	12.51	11.13	10.48	9.94	8.34	10.48	14.88
TENT	28.62	28.15	28.03	27.15	21.80	26.75	10.98	6.58	3.63	2.55	2.15	5.18	2.12	2.03	2.01	1.99	1.98	2.03	10.54
MLMP	21.97	2.89	2.00	1.99	2.00	6.17	2.51	2.45	2.41	2.33	2.23	2.39	2.34	2.23	2.21	2.14	2.08	2.20	3.08
ViDA	28.49	28.09	27.71	27.00	24.55	27.17	27.32	26.30	25.54	24.88	23.04	25.42	25.99	24.99	24.31	23.62	22.16	24.21	25.56
SAR	28.64	28.22	28.19	26.93	20.48	26.49	28.94	26.12	22.56	19.17	12.54	21.87	24.84	20.57	16.70	22.19	24.54	21.77	23.69
XF-CLS w/o DINO	28.61	27.95	27.46	27.27	25.05	27.27	29.43	28.47	27.68	27.27	25.36	27.64	29.49	28.59	27.83	27.40	25.41	27.74	27.57
XF-CLS	**33.63**	**32.27**	**31.29**	**30.80**	**27.69**	**31.14**	**32.76**	**31.91**	**30.85**	**30.29**	**27.59**	**30.68**	**32.41**	**31.69**	**30.71**	**30.20**	**27.50**	**30.50**	**30.75**

Prompt robustness. Table 3 probes the visual–text interface by evaluating frozen adapted states with canonical class names and predetermined synonyms. Before output refinement, we also measure the ground-truth-class text margin, $m_i = Z^{\mathrm{pre}}_{i,y_i} - \max_{k \neq y_i} Z^{\mathrm{pre}}_{i,k}$, averaged over valid sampled pixels; ground truth is used only for this offline diagnostic. TENT, MLMP, and EATA turn the margin negative under both prompt sets and fall to 3.71–5.86 mIoU, whereas XF-CLS keeps positive margins and achieves the best canonical mIoU (33.57). Synonyms still reduce its mIoU by 5.67 points, so a positive margin does not imply prompt invariance. ViDA has the largest margins and the best synonym mIoU here, so margin size alone does not determine continual segmentation performance.

Stability beyond driving scenes. In a separate experiment without output refinement, we adapt from source over ten cycles of VOC20 (14,490 images) with a fixed vocabulary. XF-CLS keeps its all-class mIoU nearly constant (61.42 → 61.30 from cycle 1 to 10), as does CoTTA, whereas SAR declines from 61.12 to 44.59 and disabling recovery collapses XF-CLS from 57.62 to 8.35 within three cycles (Table 18). After a driving→VOC→driving detour, XF-CLS regains its pre-detour driving performance (ACDC 30.81 vs. 30.69; OnDA 27.91 vs. 27.59 mIoU), whereas disabling recovery during the return lowers these to 24.48 and 3.74 (Table 19); all six paired order differences favor recovery.

4.4 Roles of Adaptation, Recovery, and Refinement

Table 4 separates online adaptation from frozen output refinement. Applying the same refinement schedule to frozen NACLIP gives 27.43/29.66 mIoU on ACDC/OnDA, and full XF-CLS adds 5.50/1.10 points. With XFR fixed at full strength in both arms, adaptation still adds 3.87/0.56 points (Appendix I). Online adaptation thus provides most of the ACDC gain, whereas structural refinement contributes more on OnDA, where adaptation still adds a consistent improvement. These differences reflect contributions within the full system, not independent additive gains.

What sustains adaptation? Disabling source-anchored recovery while keeping full refinement reduces all-stream mIoU to 17.38/6.51 and final-round mIoU to 5.46/1.98 on ACDC/OnDA, so frozen output correction alone cannot prevent drift in the evolving model. The auxiliary rules matter far less: jointly using a source-only anchor, removing the entropy filter, and fixing the structural gate

Table 3: Prompt robustness after ACDC adaptation. The adapted state is frozen for canonical and synonym evaluations. mIoU uses the final prediction; text margins are measured before output refinement. This diagnostic uses a separate adaptation configuration (Appendix I), so its scores are not directly comparable to online stream averages.

	mIoU			Text margin		
Method	Canonical↑	Synonym↑	Drop	Canonical↑	Synonym↑	Drop
Source	24.19	21.91	2.28	0.8185	0.0973	0.7211
CoTTA	24.42	21.95	2.47	0.8530	0.0958	0.7573
EATA	5.85	5.86	−0.01	−3.0212	−3.1194	0.0982
TENT	3.76	3.76	0.00	−3.7800	−3.3233	−0.4567
MLMP	3.71	3.71	0.00	−4.0877	−4.2520	0.1643
ViDA	29.00	**29.46**	−0.46	**6.6798**	**4.7826**	1.8972
SAR	33.28	27.84	5.44	3.2346	2.2766	0.9581
XF-CLS w/o DINO	31.46	26.82	4.64	2.1705	1.3882	0.7823
XF-CLS	**33.57**	27.90	5.67	2.1886	1.3982	0.7904

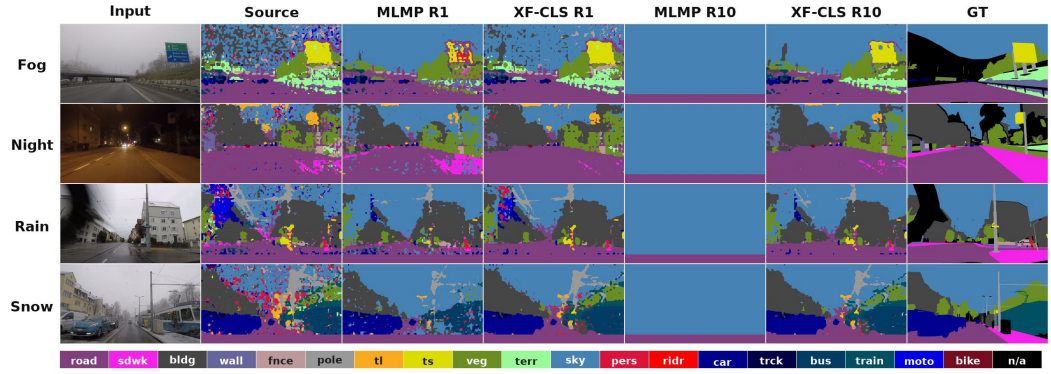

Figure 3: Qualitative ACDC predictions at rounds 1 and 10. XF-CLS keeps regions coherent in both rounds, whereas MLMP's predictions collapse by round 10.

and trend factor to one changes the mean by at most 0.11 points (Appendix I.4). The entropy filter, for instance, skips only 3.04% and 0.63% of inner steps on ACDC and OnDA, mostly during startup. These results support the division of roles in XF-CLS: Hippocampal Adaptation provides plasticity, source-anchored recovery in Neocortical Consolidation sustains it, and the frozen structural path refines the predictions.

4.5 Efficiency and Design Sensitivity

Computation and memory. XF-CLS updates only 102,400 parameters, but online updates and the frozen observer add computation (Table 5). Reusing each image's frozen DINO features reduces observer passes from three to one and latency from 691.95 to 396.20 ms, with identical outputs and state trajectories. The slow memory and source reference together occupy less than 1 MiB; the backward pass through the adaptable backbone dominates peak GPU memory (Appendix I.5).

Table 4: Roles of adaptation, refinement, and recovery. Values are all-stream mIoU (mean ± sample SD over three stream orders). Frozen NACLIP and full XF-CLS use the same refinement schedule. Removing output refinement retains the structural recovery signal; removing recovery retains full refinement.

Configuration	ACDC↑	OnDA↑
Frozen NACLIP + XFR	27.4273 ± 0.0004	29.6555 ± 0.0006
XF-CLS w/o output refinement	30.2153 ± 0.0054	27.5872 ± 0.0008
XF-CLS	32.9246 ± 0.0019	30.7527 ± 0.0025
XF-CLS w/o recovery	17.3778 ± 0.3966	6.5064 ± 0.0212

Table 5: Computation and memory on an H200 GPU. Latency excludes data loading and transfer; memory is peak allocated. The profile includes adaptation startup (20 warm-up and 100 timed images per repeat, three repeats). DINO passes count physical forwards per image.

Variant	Latency (ms)	FPS	Peak (GiB)	DINO passes
Source (NACLIP)	68.99	14.50	2.12	0
Frozen NACLIP + XFR	219.13	4.56	3.43	1
XF-CLS w/o DINO	246.22	4.06	18.04	0
XF-CLS	691.95	1.45	19.17	3
XF-CLS (feature reuse)	396.20	2.52	19.24	1

Design sensitivity. On a fixed ACDC stream without output refinement, adapting all visual-LN affine parameters gives 30.23 mIoU, compared with 25.71, 28.84, and 25.28 when updates are restricted to LN scales, the later half of the visual LNs, or the final LN (Appendix I.4). The consolidation rate matters: $\alpha = 0.999$ gives 20.73 ACDC mIoU, whereas 0.9999 and 0.99999 both reach 32.50 (Table 14). One recovery rate shared by both benchmarks ($r_0 = 0.002$) still gives about 30.75 mIoU on each (Appendix I). Within XFR, removing graph smoothing costs $1.33/1.51$ points on ACDC/OnDA relative to the full-setting mean, more than removing prototype consensus ($0.31/0.40$) or the disagreement gate ($0.23/0.21$); these studies use the configurations in Appendix F. Sweeps over the XFR schedule and blend strength show no strong sensitivity to any single XFR hyperparameter (Appendix D).

Robustness. Appendix E further varies the prompt, backbone, condition order, and stream length on ACDC. Across three alternative prompt families (street-scene, adverse-weather, and semantic-region templates), XF-CLS averages 31.20 mIoU with a worst case of 29.96, at most 2.79 points below the reference prompt and well above the source model's 25.57 average over the same templates (Table 8). With the smaller NACLIP ViT-B/16 backbone and unchanged hyperparameters, it improves source mIoU from 27.77 to 32.91, and the variant without DINO reaches 31.44 (Table 9). Under reversed and alternating condition orders, it reaches 32.43 mIoU in both cases, whereas TENT collapses to 4.54 and 5.22 (Table 10). In a reduced 20-round stress test with 50 images per condition, the variant without DINO shows no late collapse, ending at 29.41 mIoU compared with 3.84 for TENT (Table 11). Because the variant without DINO is also stable across these variations, stability comes mainly from source-anchored consolidation, while the frozen structural path adds accuracy.

5 Conclusion

We presented **XF-CLS**, a source-free framework for stable OVSS-CTTA. It couples *Hippocampal Adaptation* for online visual updates with source-anchored *Neocortical Consolidation* and a frozen cross-foundation structural path. On ACDC and OnDA, XF-CLS sustains segmentation gains while retaining near-source performance on VOC classes outside the adaptation vocabulary, and it remains stable on a recurring VOC stream outside the driving domain. Matched controls identify source-anchored recovery as the main stabilizer, while frozen refinement further improves dense predictions. These results suggest that, once updates are carried across a long stream, keeping them stable matters as much as the per-image adaptation objective. They also show that short evaluations can hide this failure: MLMP starts with the best round-1 fog score on ACDC but falls to 5.13 by round 10.

Limitations and future work. Recovery and refinement settings were selected offline using labeled benchmark performance. Our experiments use NACLIP backbones and controlled, finite streams with a fixed vocabulary; recorded deployment sequences with gradual or unseen shifts remain to be tested. The retention test covers VOC classes outside the adaptation vocabulary, which may still appear in foundation-model pretraining, and our horizon analysis covers the EMA recurrence rather than the full adaptive system. Online updates and structural inference add substantial computation (Table 5), and synonym prompts still reduce mIoU (Table 3). Future work includes label-free model selection, lighter structural observers, stronger prompt robustness, and evaluation across OVSS architectures and changing vocabularies.

Acknowledgements

This work was supported in part by the National Science and Technology Council (NSTC) under Grants NSTC 115-2634-F-002-012, 115-2221-E-002-183-MY3, 115-2221-E-110-068-MY3 and 115-2640-E-110-005, and National Taiwan University (NTU) under Grant 115L900902. We thank the National Center for High Performance Computing (NCHC) of NARLabs in Taiwan for providing computational and storage resources.

References

[1] Elahe Arani, Fahad Sarfraz, and Bahram Zonooz. Learning fast, learning slow: A general continual learning method based on complementary learning system. In *ICLR*, 2022.

[2] Sam Blakeman and Denis Mareschal. A complementary learning systems approach to temporal difference learning. *Neural Networks*, 122:218–230, 2020.

[3] Walid Bousselham, Felix Petersen, Vittorio Ferrari, and Hilde Kuehne. Grounding everything: Emerging localization properties in vision-language transformers. In *CVPR*, 2024.

[4] Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *CVPR*, 2020.

[5] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental learning in semantic segmentation from image labels. In *CVPR*, 2022.

[6] Dian Chen, Dequan Wang, Trevor Darrell, and Sayna Ebrahimi. Contrastive test-time adaptation. In *CVPR*, 2022.

[7] Tzu-Ting Chuang, Ting-Yun Wei, Yu-Hsing Hsieh, Chu-Song Chen, and Huei-Fang Yang. Continual cell instance segmentation of microscopy images. In *ICASSP*, 2023.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[9] Sebastian Cygert, Damian Sójka, Tomasz Trzciński, and Bartłomiej Twardowski. Realistic evaluation of test-time adaptation: Unsupervised model selection. In *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, pages 75–86, 2026.

[10] Ulindu De Silva, Didula Samaraweera, Sasini Wanigathunga, Kavindu Kariyawasam, Kanchana Ranasinghe, Muzammal Naseer, and Ranga Rodrigo. Test-time optimization for domain adaptive open vocabulary segmentation. *Neural Networks*, 205:109689, 2027.

[11] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary universal image segmentation with MaskCLIP. In *ICML*, 2023.

[12] Mario Döbler, Robert A. Marsden, and Bin Yang. Robust mean teacher for continual and gradual test-time adaptation. In *CVPR*, 2023.

[13] Chandler Timm C. Doloriel, Yunbei Zhang, Sarthak Kumar Maharana, Muhammad Salman Siddiqui, Tor Kristian Stevik, Fadi Al Machot, Kristian Hovde Liland, and Habib Ullah. Towards continual test-time adaptation of vision-language models in open-vocabulary semantic segmentation. *arXiv preprint arXiv:2608.29923*, 2026.

[14] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010.

[15] Sina Hajimiri, Ismail Ben Ayed, and Jose Dolz. Pay attention to your neighbours: Training-free open-vocabulary semantic segmentation. In *WACV*, 2025.

[16] Shirsendu Sukanta Halder, Jean-Francois Lalonde, and Raoul de Charette. Physics-based rendering for improving robustness to rain. In *ICCV*, 2019.

[17] Yu-Hsing Hsieh, Guan-Sheng Chen, Shun-Xian Cai, Ting-Yun Wei, Huei-Fang Yang, and Chu-Song Chen. Class-incremental continual learning for instance segmentation with image-level weak supervision. In *ICCV*, 2023.

[18] Ching-Yi Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen. Compacting, picking and growing for unforgetting continual learning. In *NeurIPS*, 2019.

[19] Dongjun Hwang, Yejin Kim, Minyoung Lee, Seong Joon Oh, and Junsuk Choe. OVS meets continual learning: Towards sustainable open-vocabulary segmentation. In *NeurIPS*, 2025.

[20] Md Akil Raihan Iftee, Mir Sazzat Hossain, Rakibul Hasan Rajib, Tariq Iqbal, Md Mofijul Islam, M Ashraful Amin, Amin Ahsan Ali, and AKM Mahbubur Rahman. SloMo-Fast: Slow-momentum and fast-adaptive teachers for source-free continual test-time adaptation. *arXiv preprint arXiv:2511.18468*, 2025.

[21] Mengcheng Lan, Chaofeng Chen, Yiping Ke, Xinjiang Wang, Litong Feng, and Wayne Zhang. ProxyCLIP: Proxy attention improves CLIP for open-vocabulary segmentation. In *ECCV*, 2024.

[22] Wei-Hua Li, Huei-Fang Yang, and Chu-Song Chen. Continual learning for weakly-supervised histopathology tissue segmentation. In *Proc. IEEE Conf. Comput. Intell. Bioinform. Comput. Biol. (CIBCB)*, 2025.

[23] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted CLIP. In *CVPR*, 2023.

[24] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang. ViDA: Homeostatic visual domain adapter for continual test time adaptation. In *ICLR*, 2024.

[25] Fen Luo, Sen Li, and Zhanqiang Huo. CARE: Conflict-aware regulation and evolution for continual test-time open-vocabulary semantic segmentation. *J. King Saud Univ. Comput. Inf. Sci.*, 38(7):799, 2026.

[26] Sarthak Kumar Maharana, Shambhavi Mishra, Yunbei Zhang, Shuaicheng Niu, Taki Hasan Rafi, Jihun Hamm, Marco Pedersoli, Jose Dolz, and Yunhui Guo. Continual test-time adaptation in computer vision: Methods, benchmarks, and future directions. *Transactions on Machine Learning Research*, 2026.

[27] Robert A. Marsden, Mario Döbler, and Bin Yang. Universal test-time adaptation through weight ensembling, diversity weighting, and prior correction. In *WACV*, 2024.

[28] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

[29] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, 2022.

[30] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *ICLR*, 2023.

[31] Mehrdad Noori, David Osowiechi, Gustavo A. Vargas Hakim, Ali Bahri, Moslem Yazdanpanah, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. Test-time adaptation of vision-language models for open-vocabulary semantic segmentation. In *NeurIPS*, 2025.

[32] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[33] David Osowiechi, Mehrdad Noori, Gustavo A. Vargas Hakim, Moslem Yazdanpanah, Ali Bahri, Milad Cheraghalikhani, Sahar Dastani, Farzad Beizaee, Ismail Ben Ayed, and Christian Desrosiers. WATT: Weight average test time adaptation of CLIP. In *NeurIPS*, 2024.

[34] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *ECCV*, 2022.

[35] Reza Qorbani, Gianluca Villani, Theodoros Panagiotakopoulos, Marc Botet Colomer, Linus Härenstam-Nielsen, Mattia Segu, Pier Luigi Dovesi, Jussi Karlgren, Daniel Cremers, Federico Tombari, and Matteo Poggi. Semantic library adaptation: LoRA retrieval and fusion for open-vocabulary semantic segmentation. In *CVPR*, 2025.

[36] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[37] Christos Sakaridis, Haoran Wang, Ke Li, René Zurbrügg, Arpit Jadon, Wim Abbeloos, Daniel Olmeda Reino, Luc Van Gool, and Dengxin Dai. ACDC: The adverse conditions dataset with correspondences for robust semantic driving scene perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 48(3):2970–2988, 2026.

[38] Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *NeurIPS*, 2022.

[39] Guanglong Sun, Shuang Cui, Bo Lei, Liyuan Wang, Zihan Zhai, Hongwei Yan, Hang Su, Jun Zhu, and Yi Zhong. ComMem: Complementary memory systems for test-time adaptation of vision-language models. *arXiv preprint arXiv:2606.28719*, 2026.

[40] Devavrat Tomar, Guillaume Vray, Behzad Bozorgtabar, and Jean-Philippe Thiran. TeSLA: Test-time self-learning with automatic adversarial augmentation. In *CVPR*, 2023.

[41] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *IJCV*, 129(2):341–360, 2021.

[42] 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 *WACV*, 2025.

[43] Timmy S. T. Wan, Jun-Cheng Chen, Tzer-Yi Wu, and Chu-Song Chen. Continual learning for visual search with backward consistent feature embedding. In *CVPR*, 2022.

[44] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

[45] Feng Wang, Jieru Mei, and Alan Yuille. SCLIP: Rethinking self-attention for dense vision-language inference. In *ECCV*, 2024.

[46] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8): 5362–5383, 2024.

[47] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, 2022.

[48] Wenxi Wang. Stateful streaming open-vocabulary segmentation test-time adaptation: Persistent-state diagnosis and freeze-to-confirm recovery. *Research Square preprint*, 2026. doi:10.21203/rs.3.rs-10866033/v1.

[49] Wenxi Wang. Persistent-state management for streaming test-time adaptation in open-vocabulary segmentation. *Research Square preprint*, 2026. doi:10.21203/rs.3.rs-11043818/v1.

[50] Monika Wysoczańska, Oriane Siméoni, Michaël Ramamonjisoa, Andrei Bursuc, Tomasz Trzciński, and Patrick Pérez. CLIP-DINOiser: Teaching CLIP a few DINO tricks for open-vocabulary semantic segmentation. In *ECCV*, 2024.

[51] Longhui Yuan, Binhui Xie, and Shuang Li. Robust test-time adaptation in dynamic scenarios. In *CVPR*, 2023.

[52] Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. In *ICML*, 2023.

[53] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from CLIP. In *ECCV*, 2022.

XF-CLS: Cross-Foundation Complementary Learning Systems for Continual Test-Time Adaptation in Open-Vocabulary Semantic Segmentation

Appendix

This appendix supplements the main-text description of Hippocampal Adaptation, Neocortical Consolidation, and the frozen cross-foundation structural path. Appendix A discusses broader impacts. Appendix B derives the EMA time constant and its finite-horizon interpretation. Appendix C specifies the online protocol and implementation. Appendices D–F report sensitivity, robustness, and component analyses. Appendix G reports runtime, and Appendix H presents qualitative comparisons. Appendix I provides matched controls, vocabulary-retention evaluations, and recovery analyses.

A Broader impacts

This work improves open-vocabulary semantic segmentation over the evaluated continual streams. More stable adaptation could benefit applications such as transportation and environmental monitoring, although benchmark accuracy does not establish deployment safety. Potential risks include:

- **Surveillance and privacy risk:** More stable online adaptation could be used in large-scale vision surveillance systems. Although our experiments are restricted to public academic benchmarks (Cityscapes/ACDC/OnDA/VOC2012) and do not use personal identifiers, real deployments should still follow local privacy regulations, limit camera usage to appropriate settings, and apply de-identification and minimal data-retention policies.
- **Bias amplification and unfairness:** The method adapts on unlabeled target streams; if the target stream contains systematic biases, long-horizon adaptation may reinforce them. Deployment should therefore monitor performance disparities across scenes or groups, add fairness checks where relevant, and disable online updates or roll back to a trusted checkpoint when harmful drift is detected.
- **Misuse and overconfidence:** Automatic adaptation can produce misleading confidence signals, especially under extreme degradation. We include bounded reliability gating and source-anchored recovery, but safety-critical applications should still use external monitoring, human review, or conservative redundant systems.

We mitigate these risks through the design choices described in the main paper: source-anchored slow memory to limit accumulated drift in the evaluated streams, a frozen structural observer that is never trained as a teacher, and runtime gradient skipping and source-anchored interpolation together with a drift budget (see Section 3.4 and the experimental setup in Section 4). The datasets and pretrained models are cited; their use remains subject to their respective licenses and access conditions.

The evaluated stability gains do not replace application-specific validation before deployment.

B Derivation and Scope of the EMA Time Constant

We derive Eq. (6) explicitly. Consider the slow-EMA recursion in Eq. (2),

$$\theta_t^s = \alpha\theta_{t-1}^s + (1-\alpha)\tilde{\theta}_t^f, \qquad \alpha \in (0,1),$$

ignoring the separate periodic drift-budget pullback. Here the time index counts inner EMA updates and the displacement in the denominator is assumed nonzero. Suppose at $t = 0$ the fast parameters jump to a drifted value $\tilde{\theta}_{\text{drift}}^f$ and remain there: $\tilde{\theta}_t^f \equiv \tilde{\theta}_{\text{drift}}^f$ for $t \geq 1$. Unrolling the recursion yields

$$\theta_T^s = \alpha^T\theta_0^s + (1-\alpha)\sum_{i=1}^{T}\alpha^{T-i}\tilde{\theta}_{\text{drift}}^f = \alpha^T\theta_0^s + (1-\alpha^T)\tilde{\theta}_{\text{drift}}^f,$$

using $\sum_{i=1}^{T}\alpha^{T-i} = (1-\alpha^T)/(1-\alpha)$. Subtracting θ_0^s from both sides,

$$\theta_T^s - \theta_0^s = (1-\alpha^T)\left(\tilde{\theta}_{\text{drift}}^f - \theta_0^s\right),$$

so the absorption ratio is

$$q_T \;=\; \frac{\|\theta_T^s - \theta_0^s\|}{\|\tilde{\theta}^f_{\text{drift}} - \theta_0^s\|} \;=\; 1 - \alpha^T.$$

Requiring $q_T \le \epsilon$ gives $\alpha^T \ge 1 - \epsilon$, hence $\alpha \ge (1-\epsilon)^{1/T}$. For $0 < \epsilon < 1$ and positive integer T, the equivalence holds exactly. Define $\tau_{\text{sat}} = \min\{t \in \mathbb{N} : \alpha^t \le e^{-1}\} = \lceil -1/\log\alpha \rceil$. As $\alpha \to 1$, $\log\alpha = -(1-\alpha) + O((1-\alpha)^2)$, giving $\tau_{\text{sat}} \approx 1/(1-\alpha)$. This is a $1/e$ residual time constant, not complete saturation. This yields the standard EMA time-constant interpretation used in Section 3.6.

For varying fast states with $\|\tilde{\theta}_i^f - \theta_0^s\| \le D$ for all $i \le T$, the triangle inequality gives $\|\theta_T^s - \theta_0^s\| \le (1-\alpha)\sum_{i=1}^{T} \alpha^{T-i} D = (1-\alpha^T)D$. A fixed displacement of norm D attains equality. Thus the worst-case statement requires a common displacement bound; it does not bound prediction error or prove stability of the coupled adaptive system.

C Experimental Protocol and Implementation Details

Online protocol. All experiments follow the source-free OVSS-CTTA protocol described in Section 3. The model processes an unlabeled target stream sequentially, adapts to each image before predicting it, and carries its adapted parameters forward without source images, online target labels, or externally supplied image-, batch-, domain-, or round-boundary resets. Method-internal recovery, including SAR resets, remains allowed. This no-reset constraint distinguishes OVSS-CTTA from episodic OVSS-TTA, where the model is restored to its source initialization after each target batch.

Adapted and frozen parameters. Unless otherwise stated, the adaptable OVSS model is NACLIP ViT-L/14 and the frozen structural model is DINOv2 ViT-L/14. XF-CLS updates only the visual LayerNorm affine parameters of the OVSS image encoder, corresponding to $102{,}400$ parameters; baseline subsets are specified separately. The text encoder, token embeddings, remaining vision–language backbone, and DINOv2 are frozen throughout CTTA.

Optimization and recovery. Fast and slow parameters are initialized from the source model. For each image, XF-CLS repeats the complete gradient, slow-EMA, and recovery sequence $S = 2$ times using Adam, then predicts from the updated fast state (*adapt-then-predict*). If the entropy filter skips a gradient step, slow-state maintenance and recovery still proceed. The inner-update index t and image index n therefore refer to different clocks. The fast objective is the uniform dense entropy of Eq. (3). Neocortical Consolidation uses slow parameter averaging with $\alpha = 0.99999$, source-anchor blend $\beta = 0.5$, drift-budget threshold 0.30, drift pullback 0.5, and gate bounds $(g_{\min}, g_{\max}) = (0.5, 1.5)$. The base recovery rate is $r_0 = 0.001$ on ACDC and $r_0 = 0.002$ on OnDA, selected offline using labeled benchmark mIoU; all other hyperparameters are shared across benchmarks. The entropy filter compares the normalized image entropy with 1.3 times its running mean (EMA rate 0.01, initialized at zero). The drift-budget check runs at every inner iteration of images whose index is divisible by 100. In Eq. (4), $\bar{d}_t = 0.99\bar{d}_{t-1} + 0.01 d_t$ is updated before the gate is computed, and Δa_t is the mean of the last ten minus the mean of the preceding ten inner-step agreement values; it is zero until twenty values are available.

Cross-foundation refinement. The XFR module uses top-11 DINO-neighbor smoothing, two iterations of prototype consensus, an absent-class prior, and uncertainty/disagreement routing with graph/prototype blend strengths $0.45/0.35$ and horizon-aware annealing ($\tau_{\text{xfr}} = 5600$, $w_{\min} = 0.12$). The output schedule uses the image index, $w_n = \max(w_{\min}, \exp(-(n-1)/\tau_{\text{xfr}}))$. Both Φ and the conservative branch Φ_g use frozen DINOv2 features. Annealing changes their relative weight; it does not remove structural refinement. No gradient is propagated through DINOv2, Φ, or Φ_g into the online update.

Benchmarks. For Cityscapes-to-ACDC, we use fog, night, rain, and snow, repeated for 10 continual rounds, totaling $\sim$16,000 images and $\sim$32,000 inner update attempts under $S = 2$. This stream tests repeated heterogeneous domain shifts. For OnDA, we use rain intensities of 25, 50, 75, 100, and 200 mm/h, repeated for 5 continual rounds, totaling $\sim 12{,}500$ images and $\sim 25{,}000$ inner update attempts. This stream tests recurring rain-intensity shifts under progressively stronger visual degradation.

Table 6: XFR decay sensitivity. Single-seed runs comparing different annealing horizons τ_{xfr} for the cross-foundation refinement weight w_n. Δ is measured against the $\tau_{\text{xfr}} = 5600$ reference on the same benchmark.

XFR decay (images)	**ACDC↑**	Δ	**OnDA↑**	Δ
1400	32.36	−0.14	30.03	−0.28
2800	32.43	−0.07	30.13	−0.18
5600	32.50	+0.00	30.31	+0.00

Table 7: XFR hyperparameter sensitivity around the reference blend. Sensitivity rows are single-seed full-stream runs except the reference and strong-blend rows, which report the mean ± standard deviation over three runs.

Configuration	**top-k**	**Graph**	**Proto**	**ACDC↑**	Δ	**OnDA↑**	Δ
Reference blend	11	0.35	0.25	32.503 ± 0.008	+0.00	30.302 ± 0.007	+0.00
Smaller graph neighborhood	7	0.35	0.25	32.34	−0.16	30.14	−0.16
Larger graph neighborhood	15	0.35	0.25	32.62	+0.12	30.37	+0.07
Weak XFR blend	11	0.25	0.15	32.02	−0.48	29.75	−0.55
Strong XFR blend	11	0.45	0.35	$\mathbf{32.921 \pm 0.007}$	**+0.42**	$\mathbf{30.751 \pm 0.007}$	**+0.45**

Hardware. The RTX 4090 (24 GB) throughput measurements and H200 profiles use different protocols and are reported separately (Appendices G and I.5). The matched controls in Table 4 and VOC evaluations ran on single H200 GPUs. The long full-stream runs used RTX 3090 (24 GB) and RTX 4090 nodes; their wall-clock totals reflect shared-cluster load.

Reference configurations. The sensitivity tables distinguish a reference XFR blend ($0.35/0.25$ for graph/prototype refinement, $\tau_{\text{xfr}} = 5600$ images) from the main-paper blend ($0.45/0.35$). The recovery-rate sweep instead fixes $\tau_{\text{xfr}} = 2800$. These reference experiments and the matched main-paper controls are separate run cohorts; their means and error bars should not be interchanged. The reduced 20-round ACDC test uses 50 images per condition, totaling 4,000 images. It tests more cycles, but fewer images than the main 16,000-image ACDC stream.

D XFR Schedule and Hyperparameter Sensitivity

This section analyzes how XF-CLS depends on the cross-foundation refinement (XFR) schedule and its key blending hyperparameters. We separately vary (i) the annealing horizon τ_{xfr} in Eq. (5), which controls the transition from full refinement toward the conservative graph branch; (ii) the graph neighborhood size top-k used by both Φ_g and Φ; and (iii) the graph and prototype blend strengths inside Φ.

Annealing horizon. Table 6 varies the image-based decay scale while holding the remaining settings fixed. Increasing τ_{xfr} from 1400 to 5600 improves ACDC/OnDA mIoU by $0.14/0.28$. These modest differences favor a slower transition toward the conservative branch within the tested range.

Graph neighborhood and blend strength. Table 7 sweeps the graph neighborhood size and the (graph, prototype) blend strengths around the reference blend defined in Appendix C. Two trends emerge. First, the graph neighborhood size has a moderate but consistent effect: a larger top-k gives a small improvement on both benchmarks within the tested range. Second, the blend strengths matter more than top-k: a weaker blend (Graph 0.25, Proto 0.15) loses up to 0.55 mIoU on OnDA, while a stronger blend (Graph 0.45, Proto 0.35) gives the largest improvement on both benchmarks. The main paper uses this strong blend; the three-run sensitivity estimates here come from a separate cohort from the matched controls in Table 4.

Within these tested ranges, schedule and neighborhood changes have modest effects, while the stronger blend gives the largest gain. These are target-informed sensitivity comparisons, not an unlabeled selection procedure.

Table 8: Prompt-template preservation on ACDC after continual adaptation. Alternative prompt means are averaged over street-scene, adverse-weather, and semantic-region templates. XF-CLS retains high mIoU under these prompt variations, while TENT has low accuracy throughout. The signed change is worst alternative minus reference mIoU.

Method	**Reference↑**	**Alt. Avg.↑**	**Worst Alt.↑**	**Worst Δ↑**
Source	23.930	25.573	24.085	+0.155
TENT	3.775	3.775	3.775	+0.000
XF-CLS w/o DINO	30.637	29.628	28.355	−2.283
Full XF-CLS	**32.745**	**31.203**	**29.958**	−2.787

Table 9: Backbone-scale evaluation on ACDC with NACLIP ViT-B/16. Same 10-round CTTA protocol as the main result; XF-CLS hyperparameters are unchanged.

Method	**mIoU↑**	**Gain vs. Source↑**
Source	27.768	+0.000
XF-CLS w/o DINO	31.440	+3.672
Full XF-CLS	**32.907**	**+5.139**

E Prompt, Backbone, and Stream Robustness

This section evaluates four robustness dimensions of XF-CLS that complement the main-text results: (i) sensitivity to the OVSS prompt template, (ii) transfer to a smaller NACLIP backbone, (iii) sensitivity to the order in which adverse-condition domains are presented, and (iv) behavior under more repeated condition cycles.

Prompt-template preservation. An OVSS-CTTA method should not destroy the language-conditioned interface even when the deployment-time prompt differs from the one used at adaptation. Table 8 compares each method on ACDC under a reference prompt and three alternative prompt families (street-scene, adverse-weather, and semantic-region templates). TENT yields only 3.78 mIoU under every tested prompt family. This indicates poor segmentation across these prompts; equal aggregate scores alone do not establish complete loss of text dependence. XF-CLS w/o DINO and the full XF-CLS preserve high reference mIoU and remain within a few mIoU points of that score across alternative prompts. Full XF-CLS has the highest reference and alternative-prompt average mIoU. Its worst alternative-prompt drop is 2.787 points, compared with 2.283 without DINO, so high retained accuracy should not be interpreted as prompt invariance.

Backbone scale. Table 9 replaces the ViT-L/14 OVSS backbone with the smaller NACLIP ViT-B/16, keeping the same 10-round CTTA protocol and all XF-CLS hyperparameters. The relative gain pattern carries over: full XF-CLS improves over the Source baseline by $+5.14$ mIoU and over the variant without DINO by $+1.47$ mIoU. The gains also hold at this smaller NACLIP scale. This is a same-family comparison and does not establish transfer to a second OVSS model family.

Stream-order sensitivity. The main-text ACDC stream uses fog → night → rain → snow. Table 10 repeats the 10-round CTTA evaluation under two perturbed orders: a reverse order (snow → rain → night → fog) and a hard alternating order (night → snow → fog → rain) that changes which conditions are adjacent. TENT collapses under both perturbations, dropping below 5.5 mIoU in overall accuracy. The full variant yields 32.43 for both orders, whereas the variant without DINO yields approximately $30.12/30.13$; these reference runs are separate from the matched main-paper controls. The source-anchored design remains stable under these two tested condition orders.

Long-horizon stress test. To probe more repeated shifts than the 10-round main-text evaluation, Table 11 runs ACDC for 20 rounds with 50 images per condition. This setup is intended as a diagnostic stress test rather than a full-stream measurement. TENT accumulates entropy-driven errors and collapses after roughly the mid-horizon point, dropping its worst-round score to 3.71. XF-CLS w/o DINO does not exhibit late-stream collapse: its AUC over the 20 rounds is more than 8 points higher than TENT, and its final-round score is comparable to its early-round score. For XF-CLS on

Table 10: ACDC stream-order sensitivity. Reverse uses snow → rain → night → fog; hard alternating uses night → snow → fog → rain. The final-entry column reports the last domain score in the 10-round stream.

Order	Method	Overall↑	Final entry↑	Behaviour
Reverse	TENT	4.537	3.080	Collapse with entropy updates
Reverse	XF-CLS w/o DINO	30.12	33.33	Stable without XFR
Reverse	**Full XF-CLS**	**32.43**	**35.43**	Stable with XFR
Hard alternating	TENT	5.220	3.370	Collapse under hard order
Hard alternating	XF-CLS w/o DINO	30.13	30.01	Stable without XFR
Hard alternating	**Full XF-CLS**	**32.43**	**32.22**	Stable with XFR

Table 11: Long-horizon stress test on ACDC. Each method uses 20 rounds with 50 images per condition; these results are diagnostic rather than full-stream accuracy. TENT collapses after the mid-horizon, while XF-CLS w/o DINO remains stable through R20.

Method	AUC↑	Final↑	Worst↑	Behaviour
TENT	19.070	3.840	3.708	Collapses after R13
XF-CLS w/o DINO	**27.869**	**29.413**	**23.520**	No late collapse through R20

this reduced stream, 20 rounds contain $8{,}000$ inner EMA updates; the fixed-displacement absorption is about 0.077. This describes the EMA timescale, rather than proving semantic stability.

These robustness checks support the source-anchored mechanism for the tested prompt templates, NACLIP sizes, condition orders, and finite streams. XF-CLS without DINO remains stable in these tests. Full XF-CLS improves the prompt, backbone, and order comparisons; the reduced 20-round table evaluates only the variant without DINO and TENT.

F Component Diagnostics

This section varies XFR components, recovery strength, and the slow-EMA rate under the reference settings. Appendix I.4 complements these sweeps with matched interventions at the main operating point.

XFR component ablation. Table 12 removes one XFR sub-component at a time from the reference blend. The DINO graph smoothing path is the largest single contributor: removing it costs -1.33 mIoU on ACDC and -1.51 mIoU on OnDA, larger than either the prototype-consensus or disagreement-gate removal alone. However, the graph-only XFR configuration (-0.48 ACDC, -0.55 OnDA) still falls short of the full module, indicating that prototype consensus and disagreement-aware routing add complementary value beyond local structural smoothing. The disagreement-gate removal has the smallest effect among these three interventions. These results support the full refinement module at this reference setting, with graph smoothing contributing the largest gain.

Recovery-rate sensitivity. Table 13 sweeps the base recovery rate r_0 in Eq. (4) while keeping the rest of the consolidation and refinement system fixed. Disabling recovery entirely causes severe long-horizon drift: ACDC falls to 18.15 and OnDA to 6.46 mIoU. Weak recovery is insufficient to

Table 12: XFR component ablations under the reference blend. The Full row shows the fixed-order run; component scores are single runs, with their deltas computed against the three-run full-setting mean.

Configuration	ACDC↑	Δ	OnDA↑	Δ
XF-CLS (reference blend)	32.50	+0.00	30.31	+0.00
w/o DINO graph	31.18	−1.33	28.79	−1.51
w/o cross-foundation disagreement gate	32.27	−0.23	30.09	−0.21
w/o prototype consensus	32.19	−0.31	29.90	−0.40
Graph-only XFR	32.03	−0.48	29.75	−0.55

Table 13: Recovery-rate sensitivity under a common reference configuration. Values shown are paired ($r_0^{\mathrm{ACDC}}/r_0^{\mathrm{OnDA}}$).

Setting	ACDC↑	OnDA↑	Interpretation
No recovery $(0/0)$	18.15	6.46	Long-horizon drift / collapse
Weak recovery $(0.0005/0.001)$	15.68	25.92	Insufficient pullback on at least one benchmark
Final recovery $(0.001/0.002)$	32.43	30.13	Stable operating point
Strong recovery $(0.002/0.005)$	29.99	29.29	Over-conservative but non-collapsing

Table 14: Horizon-aware consolidation sensitivity. Single-run comparison of EMA rates at a fixed finite horizon; the time-constant interpretation is given in Appendix B.

Slow EMA rate	ACDC↑	OnDA↑	Observed behavior
0.999	20.73	26.63	Lower accuracy at the faster EMA rate
0.9999	32.50	30.24	Strong at this finite horizon
0.99999	32.50	30.30	Final stable long-horizon setting

pull the fast state back on at least one benchmark—it stabilizes OnDA but lets ACDC drift to 15.68 mIoU. The final operating point $(0.001/0.002)$ gives the best stable performance on both streams. Stronger recovery gives lower aggregate accuracy than the selected rates, consistent with a trade-off between adaptation and attraction to the source-anchored recovery state. This sweep shows that recovery strength affects the best operating point. Benchmark-specific tuning improves that point, but the shared-rate control in Appendix I also remains stable.

Slow-EMA sensitivity. Table 14 examines the horizon-aware analysis of Section 3.6 and Appendix B empirically. At $\alpha = 0.999$ the slow memory has $\tau_{\mathrm{sat}} \approx 10^3$, much smaller than the $32{,}000$-inner-update ACDC horizon, corresponding to faster tracking of a displaced fast state in the EMA analysis; ACDC accuracy is 20.73 mIoU. At $\alpha = 0.9999$ ($\tau_{\mathrm{sat}} \approx 10^4$, still below the horizon), performance already reaches $32.50/30.24$. At $\alpha = 0.99999$ ($\tau_{\mathrm{sat}} \approx 10^5$) it gives $32.50/30.30$. These runs show sensitivity to the EMA rate, but do not establish a necessary threshold at $\tau_{\mathrm{sat}} > T$ or isolate the EMA from the coupled recovery dynamics. We keep $\alpha = 0.99999$ as the more conservative finite-horizon choice.

Across these reference experiments, removing refinement components lowers accuracy, disabling recovery leads to severe degradation, and faster EMA tracking can reduce performance. These findings support the system design within the tested settings; the EMA derivation remains a recurrence analysis rather than a guarantee for the full system.

G Efficiency and Runtime

This section reports the inference cost and online adaptation footprint of XF-CLS relative to representative CTTA baselines. We measure two complementary quantities. First, controlled per-image throughput (FPS) on a single workstation under a fixed probe protocol. Second, wall-clock cost of long full-stream runs on a shared cluster, where machine availability and load vary between jobs.

Hardware. Throughput numbers in Table 15 are measured on a single NVIDIA RTX 4090 (24 GB) using a fixed probe of 50 images per condition or intensity under the corresponding online protocol. Wall-clock numbers in Table 16 reflect server-side execution distributed across NVIDIA RTX 3090 (24 GB) and RTX 4090 (24 GB) nodes; XF-CLS holds two ViT-L/14 models concurrently in GPU memory (NACLIP and frozen DINOv2), well within the 24 GB envelope of either card. The wall-clock totals mix hardware and system load and should be read as operational references rather than controlled comparisons.

Throughput and online adaptation footprint. XF-CLS updates only $102{,}400$ parameters online, identical to the LayerNorm-adaptation footprint of TENT, CoTTA, EATA, SAR, and our variant without DINO, and far smaller than the $3{,}342{,}336$ parameters updated by ViDA. The throughput drop relative to the static baseline has two distinct components. The first is online adaptation itself:

Table 15: Efficiency and adaptation footprint. FPS is measured on a single NVIDIA RTX 4090 (24 GB) using a fixed probe of 50 images per condition or intensity under the corresponding online protocol. Updated parameters count only parameters modified online; frozen NACLIP/DINOv2 parameters are not counted as updated.

Method	ACDC FPS↑	OnDA FPS↑	Updated params↓	Note
Source	4.958	5.171	0	Forward-only NACLIP
TENT	3.183	3.255	102,400	LayerNorm entropy adaptation
MLMP	2.430	0.452	102,400	OnDA run uses heavier multi-step protocol
CoTTA	1.500	1.517	102,400	Restoration-based CTTA
EATA	1.954	1.975	102,400	Reliability-filtered entropy adaptation
SAR	1.966	2.001	102,400	Sharpness-aware reliable entropy
ViDA	1.088	1.095	3,342,336	Adapter-heavy OVSS baseline
XF-CLS w/o DINO	1.459	1.465	102,400	XF-CLS without the DINOv2 observer or refinement
Full XF-CLS	0.808	0.816	102,400	Frozen DINOv2/XFR adds runtime, not updated parameters

Table 16: Server wall-clock reference for long XF-CLS runs. These logs were collected from server jobs distributed across NVIDIA RTX 3090 and RTX 4090 nodes under mixed system load and are intended as operational runtime references rather than controlled benchmarks. ACDC long20 contains 32,000 images; OnDA long10 contains 25,000 images.

Variant	ACDC long20 wall time	OnDA long10 wall time	Updated params↓	Extra frozen model
XF-CLS w/o DINO	6.48 h (0.73 s/img)	9.32 h (1.34 s/img)	102,400	None
XF-CLS (reference blend)	23.09 h (2.60 s/img)	8.81 h (1.27 s/img)	102,400	DINOv2 ViT-L/14 for XFR

repeated forward/backward passes, optimizer steps, and slow-state maintenance. With $S = 2$, the variant without DINO runs at 1.459 FPS on ACDC, compared with 4.958 FPS for Source. The second is repeated frozen cross-foundation inference plus output refinement, which further reduces throughput to 0.808 FPS in the original uncached execution. Neither factor expands the online-updated parameter set. XF-CLS therefore combines a small updated parameter set with substantial adaptation and inference cost.

Long-run wall clock. Table 16 reports wall-clock totals for full multi-round XF-CLS runs. These runs span tens of thousands of adapt-then-predict steps and are scheduled across the RTX 3090 / RTX 4090 cluster, so the per-image times reflect shared-cluster execution rather than a controlled benchmark. For ACDC long20 (32,000 images), XF-CLS w/o DINO completes in ~ 6.5 hours and full XF-CLS in ~ 23 hours. For OnDA long10 (25,000 images), the two variants finish within an hour of each other (9.32 vs. 8.81 hours). Full XF-CLS takes longer on ACDC in these logs; the reversed OnDA ordering illustrates why mixed-load wall-clock totals cannot isolate method overhead.

These measurements clarify the cost of both the inner online updates and frozen structural observation/refinement; a small trainable parameter count does not imply low inference cost. The variant without DINO reduces cost and remains stable in the evaluated streams, at lower segmentation accuracy. The matched H200 profile in Appendix I.5 separately quantifies the benefit of reusing frozen DINO features.

H Qualitative Results

Figures 4 and 5 provide qualitative segmentation comparisons on ACDC and OnDA. Each row shows the input image, Source, MLMP and XF-CLS at round 1, MLMP and XF-CLS at round 10, and ground truth. The OnDA round-10 snapshots extend beyond the five-round main evaluation in Table 2. The visual trends are consistent with the quantitative results in Section 4.2.

On ACDC, MLMP produces reasonable predictions in early rounds but progressively loses object extent and semantic consistency under repeated weather cycling, sometimes collapsing to degenerate outputs in late rounds. XF-CLS preserves coherent road, vehicle, sky, and structural regions across fog, night, rain, and snow conditions, both at the early-round and late-round time slices.

On OnDA, the failure mode is sharper. Under progressively stronger rain (up to 200 mm/h), MLMP degrades early in the stream and remains unstable for the rest of the run. XF-CLS maintains stable road and vehicle regions even at the heaviest rain intensities, consistent with the quantitative finding that frozen cross-foundation structural refinement is especially valuable on OnDA.

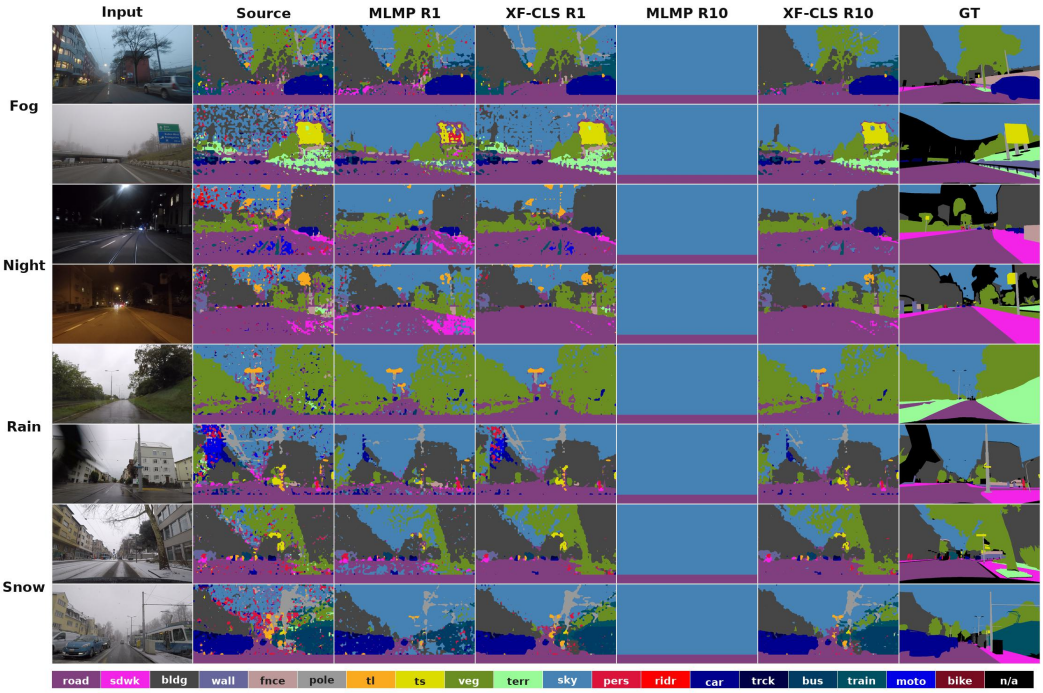

Figure 4: Qualitative comparison on Cityscapes-to-ACDC. Rows cover fog, night, rain, and snow. Columns show input, Source, MLMP at R1, XF-CLS at R1, MLMP at R10, XF-CLS at R10, and ground truth. XF-CLS maintains stable semantic regions in late rounds, while MLMP collapses in several late-round cases.

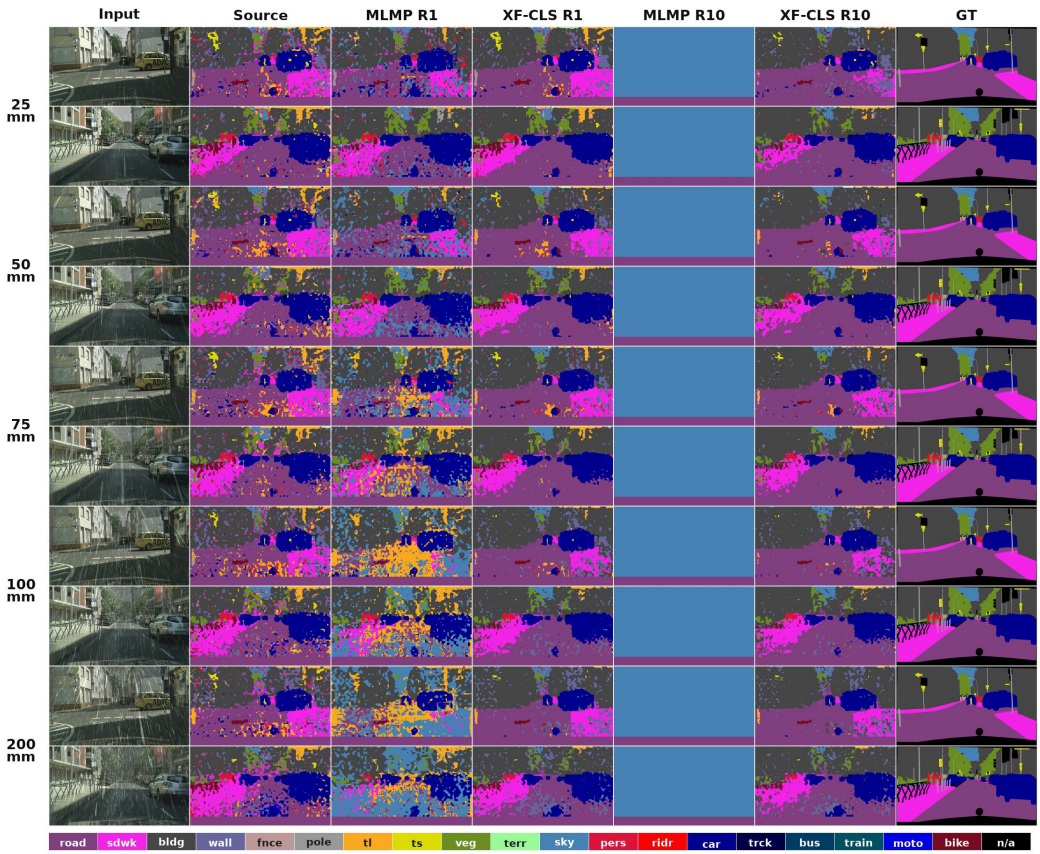

Figure 5: Qualitative comparison on OnDA. Rows cover rain intensities from 25 to 200 mm/h. Columns show input, Source, MLMP at R1, XF-CLS at R1, MLMP at R10, XF-CLS at R10, and ground truth. Under progressively stronger rain, XF-CLS produces more stable road, vehicle, and background regions than the continual MLMP baseline.

I Additional Evidence and Protocol Clarifications

This section complements the main results with matched controls, retention evaluations, and mechanism analyses. Unless stated otherwise, variability is the sample standard deviation over three pre-generated within-condition stream permutations with fixed initialization. Round–condition cells share online states and are not independent statistical trials. ACDC and OnDA are controlled repeated-shift stress tests.

I.1 Operating points, controls, and baseline ports

The online updates use neither source images nor target labels. Offline development did inspect labeled benchmark mIoU when selecting r_0 and the XFR blend. The fixed-order common-rate control ($r_0 = 0.002$ on both streams, same blend) reaches approximately 30.75 mIoU on both streams, but it is a robustness check rather than a label-free selector of the optimum. Swapping the benchmark-specific rates reduces ACDC/OnDA mIoU by about $2.17/4.61$. This remaining selection limitation is consistent with the concerns studied in TTAB and unsupervised model-selection evaluations [52, 9].

Frozen NACLIP + XFR uses the same image-based refinement schedule as full XF-CLS while holding model parameters fixed; the final fast/slow parameter hashes match Source. XF-CLS w/o output refinement retains the frozen observer for recovery, whereas XF-CLS w/o DINO removes it. Fixing the refinement weight to $w_n = 1$ in both frozen and adaptive controls also gives positive adaptation increments ($+3.87/+0.56$ on ACDC/OnDA), consistent with the scheduled-control result. These internal controls do not reproduce ProxyCLIP or CLIP-DINOiser and cannot establish superiority to those methods.

Source baseline. Source in Table 2 is raw NACLIP without adaptation or structural refinement, with 26.178 all-stream mIoU and $28.18, 26.94, 26.21, 25.69, 23.87$ across the five rain intensities. The appendix sensitivity experiments use the reference configurations specified in Appendix C; their scores and standard deviations are separate from the matched main-paper controls.

Port differences. All ports use a common NACLIP backbone, stream manifests and adapt-then-predict harness. Inner steps, updated subsets, and development budgets differ. TENT uses one visual-LN step; MLMP uses one on ACDC and ten on OnDA. Simplified CoTTA uses two steps, teacher EMA 0.999 and restoration 0.001, but omits augmentation-averaged teacher prediction. EATA uses dense entropy/diversity filtering and Fisher anchoring. SAR uses two-pass SAM and an internal loss-EMA-triggered source reset. ViDA uses rank-$4/64$ attention-output LoRA. The EATA/SAR entropy-margin ratio 0.8 was selected during port development after 0.4 yielded almost no dense updates. Baseline learning rates differ between experiments; these are controlled OVSS ports rather than exact reproductions of published defaults. The multi-order comparisons below are separate runs from the fixed-order baseline rows. XF-CLS uses Adam at learning rate 2×10^{-5}, batch size one, resized images 1120×630, 224×224 patches and stride 112. The text encoder remains fixed. In XFR, Φ_g blends logits, whereas Φ forms a normalized probability mixture and returns its log; Eq. (5) blends these output scores. The output prototypes are recomputed after recovery, while the recovery audit uses pre-update logits.

On the matched orders, SAR reaches 30.137 ± 0.118 on ACDC versus 32.9246 ± 0.0019 for XF-CLS; simplified CoTTA reaches 26.163 ± 0.012 on OnDA versus 30.7527 ± 0.0025. Paired gaps are 2.788 ± 0.118 and 4.590 ± 0.010. These are supporting controlled-port comparisons, alongside the raw source, matched static control, and recovery intervention.

I.2 Retention beyond the adaptation vocabulary

We evaluate the official VOC2012 segmentation validation split [14] (1,449 images) with a fixed 20-class vocabulary and fixed evaluation order. Fourteen classes are absent from the prior Cityscapes-19 adaptation vocabulary; this does not mean they were unseen during CLIP pretraining. Evaluation uses no further adaptation, no output XFR and no DINO. Background and void labels (0 and 255) are ignored; the denominator is foreground-only VOC20, not VOC21 including background. Predictive states are unchanged during evaluation. Standard deviations reflect the preceding driving-stream orders, not repeated VOC evaluation. For simplified CoTTA, the evaluated state is its inference teacher; for SAR it is the inference student. For the expanded comparison, ViDA uses its predictive

teacher LoRA tensors and runtime scales, with the same registered VOC text readout as every other state.

Table 17: Expanded post-stream VOC20 retention of saved predictive states. All methods use the same fixed foreground VOC20 readout without further adaptation, XFR or DINO. Values are mean $\pm$ sample SD over three preceding driving-stream orders. Retention divides absent-14 mIoU by the source value 56.0343. These three-order checkpoint runs are distinct from the single-run baseline rows of Tables 1–2; see the protocol description.

Stream	State	All-20 mIoU	Absent-14 mIoU	Retention (%)
–	Source	61.074	56.034	100.00
ACDC	TENT	4.861 ± 0.208	6.108 ± 0.200	10.90
ACDC	MLMP	1.007 ± 0.022	1.302 ± 0.026	2.32
ACDC	EATA	39.200 ± 0.972	38.914 ± 0.915	69.45
ACDC	SAR	57.805 ± 1.925	53.346 ± 1.890	95.20
ACDC	Simpl. CoTTA	61.030 ± 0.141	55.991 ± 0.140	99.92
ACDC	ViDA	54.765 ± 1.491	50.630 ± 1.602	90.36
ACDC	XF-CLS	60.581 ± 0.040	55.789 ± 0.035	99.56
OnDA	TENT	48.053 ± 0.196	46.116 ± 0.145	82.30
OnDA	MLMP	13.320 ± 0.068	12.495 ± 0.231	22.30
OnDA	EATA	61.612 ± 0.278	57.093 ± 0.236	101.89
OnDA	SAR	60.351 ± 0.201	55.829 ± 0.102	99.63
OnDA	Simpl. CoTTA	60.806 ± 0.102	55.709 ± 0.181	99.42
OnDA	ViDA	56.282 ± 1.149	53.424 ± 1.339	95.34
OnDA	XF-CLS	60.126 ± 0.009	55.348 ± 0.009	98.78

The results support near-complete aggregate retention for XF-CLS, not improved VOC accuracy or uniform per-class preservation. Several saved states retain as much as or more than XF-CLS (simplified CoTTA after both streams; EATA and SAR after OnDA), but in the same runs each stays within 0.3 mIoU of the source driving-stream All Mean or falls below it, whereas XF-CLS combines 98.8–99.6% retention with the highest driving-stream mIoU (Figure 1). This is an adaptation–retention trade-off, not universal dominance. Within each benchmark, the same three adaptation orders are paired.

Checkpoint evaluation protocol. We evaluate the saved TENT, MLMP, EATA, SAR, simplified CoTTA and ViDA states on both driving streams, each over three orders. These frozen retention evaluations introduce no new adaptation or hyperparameter selection. These checkpoint runs use learning rate 2×10^{-5}; TENT uses one step; MLMP uses one/ten on ACDC/OnDA; CoTTA, EATA and ViDA use two; SAR uses one SAM update with two gradient passes. Some settings differ from the fixed-order baseline rows in Tables 1–2; their error bars therefore describe this checkpoint cohort only. CoTTA teacher LN and ViDA teacher LoRA/scales are restored, rather than their students. A common frozen text readout isolates visual-state retention; it is not a claim to replay each port's original text-processing implementation. All six ports are included regardless of their rank. In particular, EATA after OnDA reaches 57.093 ± 0.236 absent-14 mIoU (101.89% of source), exceeding XF-CLS on this retention metric despite its lower driving-stream performance in these runs. Retention ratios can exceed 100% and are not clipped. Thus driving-stream degradation and vocabulary retention need not coincide. The text-margin comparison in Table 3 is complementary: a logit margin and absent-vocabulary retention measure different properties.

I.3 Recurring VOC stability and recovery after a semantic detour

Starting from source, we also adapt for ten VOC20 cycles (14,490 images), with distinct pre-generated image orders for cycles 2–10, no external reset, $r_0 = 0.002$, and output XFR disabled. This is a separate experiment from evaluating a driving-adapted checkpoint without updates.

Setting only recovery to zero gives all-20 mIoU $57.623 \pm 0.323 \rightarrow 33.187 \pm 0.961 \rightarrow 8.353 \pm 0.764$ in cycles 1–3. The canonical method retains 99.81% of cycle-1 all-20 mIoU at cycle 10; simplified CoTTA is also stable. This supports a non-driving finite-stream stability claim with a fixed expanded vocabulary, not arbitrary within-stream vocabulary switching or cross-family generality.

Table 18: VOC20 continual adaptation with output XFR disabled. Values are mean ± sample SD over three schedules. These are separate source-start streams.

Method	All-20 mIoU		Absent-14 mIoU	
	Cycle 1	Cycle 10	Cycle 1	Cycle 10
XF-CLS	61.420 ± 0.042	61.301 ± 0.012	56.258 ± 0.056	56.050 ± 0.008
SAR	61.119 ± 0.041	44.590 ± 2.753	55.897 ± 0.052	39.411 ± 2.544
Simplified CoTTA	61.082 ± 0.037	61.002 ± 0.055	56.042 ± 0.038	55.952 ± 0.061

In an uninterrupted driving → ten-cycle VOC20 → driving return, output XFR stays disabled. ACDC changes from 30.6946 before the detour to 25.2365 after it, then 30.8147 at return round 4. OnDA changes from 27.5911 to 26.6784, then 27.9143 at return round 2. These are re-acquisition results, not lossless retention through the detour. The recovered states retain $98.78\%/98.74\%$ of source absent-14 VOC mIoU.

Table 19: Paired return-stage intervention from identical post-VOC10 online states; output XFR is disabled in both arms. Mean ± order sample SD.

Return endpoint	Canonical	Recovery disabled	Paired difference
ACDC, round 4	30.8147 ± 0.0314	24.4771 ± 0.1412	6.3376 ± 0.1699
OnDA, round 2	27.9143 ± 0.0162	3.7431 ± 0.1069	24.1712 ± 0.0996

Both return arms start from the same post-VOC10 online state and differ only in the return-stage recovery term. All six paired order differences favor canonical recovery. Without recovery, ACDC initially reaches $30.5896/30.6870$ at rounds 1/2, but declines to $26.9193/24.4771$ at rounds 3/4; OnDA falls from 22.2133 at round 1 to 3.7431 at round 2. Recovery therefore sustains re-acquisition and prevents delayed collapse; it is not required for the initial adaptation gain.

I.4 Which mechanisms carry the observed gains?

On the fixed sequence with output XFR off, canonical ACDC/OnDA means are $30.2278/27.5928$. Setting recovery to zero reduces them to $17.9715/6.26$ and final-round mIoU to $6.1625/1.98$. Replacing the mixed anchor with source only changes aggregate mIoU by $-0.20/-0.14$; fixing the structural gate to one by $-0.01/-0.00$; disabling the entropy filter by $+0.11/+0.01$; fixing the agreement-trend multiplier to one by $+0.15/+0.03$. These are separate single-factor interventions under a fixed configuration.

A joint simplification (source-only anchor, no entropy filter, gate and trend both one) gives $30.329 \pm 0.012/27.477 \pm 0.002$ without output XFR and $32.982 \pm 0.010/30.666 \pm 0.004$ with XFR. Differences from matched canonical means are at most 0.114; maximum order-averaged final-round reduction is 0.239, without observed condition-specific collapse. This shows the four rules can be jointly simplified here, not that each is independently unnecessary. In ten VOC cycles, this simplified core changes all-20 mIoU from 61.233 ± 0.039 to 60.884 ± 0.017, and absent-14 from 56.009 ± 0.053 to 55.555 ± 0.020. The sofa class drops about 5–6 points; aggregate retention is not uniform class-wise preservation. The main results use the full XF-CLS configuration.

Visual-LayerNorm scope. On the fixed ACDC sequence with output XFR off, all visual-LN affine parameters (102,400) give 30.2278 mIoU; scale only (51,200) gives 25.7060; late-half affine (49,152) gives 28.8400; final visual LN only (2,048) gives 25.2783. All restricted choices are lower in each of the 40 correlated round–condition cells. This supports the full affine scope among the tested conservative alternatives, not universal optimality over LoRA or adapters.

Startup and entropy filtering. Fast and slow states both start from source; DINO features are available on the first image. Fast gradients may be skipped while slow maintenance and recovery continue. Instrumented fixed trajectories accept $31{,}026/32{,}000$ inner steps on ACDC and $24{,}842/25{,}000$ on OnDA (skip rates $3.04\%/0.63\%$). The first accepted updates are at images $64/66$; $495/79$ images have at least one skip. Mean entropy for skipped/accepted steps is $0.3693/0.2519$ on ACDC and $0.4117/0.3967$ on OnDA. Spearman correlation of skips with offline post-prediction error is

$0.047/-0.009$. ACDC skip rates are $5.51/1.60/1.91/3.15\%$ for fog/night/rain/snow; all 158 OnDA skips occur in the first 25-mm segment. This mostly reflects startup position rather than weather severity. Ground truth enters only the completed-trajectory diagnostic.

I.5 Matched runtime and memory measurements

On the same H200 allocation, we use 20 warm-up and 100 timed images per repeat, three repeats, and CUDA synchronization. Each repeat starts from source and times images 21–120, including entropy-filter startup. Timing excludes data loading and host-to-device transfer; it is not steady-state or whole-stream throughput. This is a separate profile from the RTX 3090/4090 results in Appendix G. The resulting profile is reported in Table 5 (Section 4.5).

The source reference and slow state store $204{,}800$ FP32 values (under 1 MiB). The observer adds about 1.1 GiB to the uncached profile; the adaptable ViT-L/14 backward pass dominates peak allocation. Peak reserved memory is 19.99 GiB. Exact per-image DINO-token caching reuses the three logical requests in one physical forward, giving $1.75\times$ throughput with identical measured outputs and state trajectories. It is an implementation optimization, not a new adaptation method. Uncertainty-triggered DINO invocation was not evaluated. These figures describe parameter efficiency with substantial compute cost; they do not establish real-time deployment or measure the total cost of all experiments and tuning.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction state the setting, the method, and the main results. Sections 3–4 and the appendix support these claims, and the Conclusion states their scope and limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: The *Limitations and future work* paragraph of the Conclusion discusses target-informed hyperparameter selection, the scope of our evidence (NACLIP backbones; controlled, finite streams with a fixed vocabulary), the scope of the retention test and the horizon analysis, the added computation, and the remaining synonym sensitivity.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes].

Justification: Section 3 states the finite-horizon analysis of the slow-memory EMA and its assumptions; Appendix B gives the complete derivation and states what the analysis does and does not cover.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: Section 3.7, Section 4, and Appendix C specify the backbones, updated parameters, optimizer and learning rate, inner steps, EMA and recovery rates, XFR settings, stream construction, and evaluation protocol; Appendix I describes how each baseline was ported. Reference code and example run scripts are provided in the supplemental material.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: The supplemental material provides reference code for the XF-CLS update and refinement operations, with example run configurations for ACDC and OnDA. All datasets are publicly available, and Section 3.7 and Appendix C specify the full protocol.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Section 3.7 and Appendix C give all hyperparameters, the optimizer, and the stream and evaluation protocols. Section 4 states that recovery rates and refinement strengths were selected offline using labeled benchmark performance.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

Justification: XF-CLS results are averaged over three stream orders with fixed initialization. Table 4, the matched comparisons in Section 4.2, and the appendix tables report sample standard deviations across these orders (Section 4). Baseline rows in Tables 1–2 are single runs, and the reported deviations do not capture variation due to initialization.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: Table 5 reports per-image latency and peak memory on an H200 GPU. Appendix C and Appendix G give the GPU types and full-stream run times.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: The work uses public datasets under their licenses, does not involve human subjects, and follows the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes].

    Justification: We discuss the broader impacts in Appendix A.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: The work does not release new models or scraped datasets with a high risk of misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes].

Justification: We cite the creators of all datasets (Cityscapes, ACDC, OnDA, PASCAL VOC) and models (CLIP, NACLIP, DINOv2) where they are introduced, and use them only for non-commercial research under their published licenses and terms of use; no data are redistributed.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes].

Justification: The reference code in the supplemental material and at `https://github.com/Gary20011207/XF-CLS` documents its required host interfaces, dependencies and limitations. No new datasets or pretrained models are introduced.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A].

Justification: The work does not involve crowdsourcing or human-subjects experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: No human-subjects research was conducted, so IRB approval is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: Large language models were not used as part of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.