TRUST:TRAJECTORY-GUIDED STATE-SPACE TEMPORAL TEST-TIME ADAPTATION

Fardad Dadboud, Hamid Azad, Miodrag Bolic
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, ON, Canada
{fardad.dadboud, hamid.azad, miodrag.bolic}@uottawa.ca

Iraj Mantegh
National Research Council
Montreal, QC, Canada
iraj.mantegh@cnrc-nrc.gc.ca

ABSTRACT

Vision-language models (VLMs) enable text-conditioned object detection, but their performance degrades under temporally evolving distribution shifts. We propose **TRUST** (**TR**ajectory-g**U**ided **S**tate-space **T**emporal test-time adaptation), a backpropagation-free Bayesian framework for video object detection that treats adaptation as temporal smoothness over a global cache capturing gradual distribution shift and an instance-level state-space filtering guided by object trajectories tracking. Our method maintains a global cache state that contains prototype vision embeddings and scale statistics. The instance-level state captures object dynamics through a Kalman-style trajectory tracking that leverages an embedding smoothing over the tracks. The resulting algorithm is backpropagation-free and works without online gradients. We evaluate on the SHIFT dataset, which provides videos with continuous intra-sequence gradual shifts. The implementations are available at https://github.com/FardadDadboud/vlm.git.

1 INTRODUCTION

Visual perception systems are widely deployed in the real world, but they rarely operate under the same assumptions as those used in their training data. Instead, the test distribution gradually shifts over time due to weather, illumination, and evolving scene content. In long-lived applications, this gradual temporal shift can bring severe performance decay, especially in safety-critical perception systems (see tables 1 and 3's vanilla rows).

Test-time adaptation (TTA) mitigates this degradation by adapting the models during inference in an unsupervised scheme over unlabeled test streams. Classic methods such as TENT demonstrate that lightweight gradient-based updates, using an entropy minimization objective, can compensate for classification accuracy degradation under the domain shift Wang et al. (2020). However, gradient-based TTA can become vulnerable in realistic online regimes—notably under small batches, mixed shifts, and evolving label frequencies—and may suffer from noisy updates, error accumulation, and catastrophic forgetting over long streams Niu et al. (2023); Wang et al. (2022). Moreover, much of the TTA literature and benchmarks centers on image classification, synthetic corruption datasets, and discrete domain shift such as CIFAR-10-C / ImageNet-C, leaving gradual temporal domain shift and dense prediction tasks like object detection comparatively less explored Zhao et al. (2023); Yoo et al. (2024); Ruan & Tang (2024).

These limitations have motivated a solution migration from per-batch optimization heuristics toward explicitly temporal formulations that treat adaptation as sequential inference over a non-stationary stream. Recent work has begun to formalize temporal TTA as a Bayesian filtering problem. State-space test time Adaptation (STAD) models gradually evolving shifts via a probabilistic state-space model, learning time-varying dynamics in hidden features, and inferring time-evolving prototypes

without labels for classification task Schirmer et al. (2024). In parallel, VLMs have become attractive for deployment due to their text-conditioned interfaces, which provide flexible usage without retraining Minderer et al. (2023); Li et al. (2022); Zhou et al. (2022); Cheng et al. (2024); Liu et al. (2024). Specifically, Grounding DINO extends transformer detection with grounded pre-training and a language decoder, enabling object detection from text queries Liu et al. (2024). Yet, VLM detectors still struggle to work robustly under real-world domain shifts (detection precision degradation under weather shifts, e.g., cloudy and foggy weather, tables 1 and 3's vanilla rows). As an instance, Bayesian Class Adaptation plus (BCA+) addresses this by reframing VLM adaptation as Bayesian inference with a training-free cache-based method that jointly adapts likelihood (feature/scale similarity) and prior (historical class distribution), fused leveraging an entropy-based uncertainty guidance Zhou et al. (2025).

Despite this progress, a key mismatch remains for object detection in videos. Adaptation signals derived from frame-wise label refinement or caches can be fooled by propagative false positives/negatives under occlusion, blur, and low-visibility frames, leading to shaky updates and error accumulation over time Zhou et al. (2025). Video, however, offers an additional signal coming from temporal dependencies of the consequent frames. Objects typically move smoothly, as well as the domain shifts take place smoothly, and trajectorie tracking can stabilize both predictions and the adaptation signal. In tracking-by-detection, classical trajectory tracking and data association, from Kalman filtering and Hungarian assignment to deep learning-based trackers, all provide efficient, real-time temporal consistency, and remain strong baselines in modern MOT systems such as ByteTrack Kalman (1960); Kuhn (1955); Wojke et al. (2017); Zhang et al. (2022c). This motivates an underexploited opportunity for temporal TTA in object detection using trajectory filtering not merely to track objects, but to decide what to trust and how to update adaptation memories over time.

In this work, we propose **TRUST**, a backpropagation-free temporal TTA framework for VLM-based video object detection addressing gradual domain shift. Trust, instead of the network's gradient-based backpropagation, utilizes two adaptation components while keeping the network's weight freezed. The first one is a global temporal cache capturing a slowly evolving domain shift via maintaining temporally updated visual features prototypes, scale statistics, and priors, and correcting the detection outputs with a Bayesian inference framework. Alongside the global cache, an instance-level state-space model that tracks object trajectories via a Kalman filter and the tracked object's feature embedding is the second module. We treat visual query embeddings and bounding box geometry as noisy observations that are tried to be refined by the uncertainty-aware fusion of the two temporal adaptation modules.

We evaluate the TRUST on SHIFT dataset, which is explicitly designed for continuous, intra-sequence shift—including gradual changes in rain or fog intensity and time of day—making it particularly well-aligned with temporal TTA for video object detection Sun et al. (2022). In contrast, standard detection robustness suites (e.g., COCO-C/Cityscapes-C) primarily induce per-image corruptions without providing temporally smooth drift suitable for trajectory-consistent filtering Michaelis et al. (2019). We evaluate Trust against a temporal TTA formulations have been developed primarily for classification streams Schirmer et al. (2024) that we adopted for the object detection task, and our implementation of the BCA+ method.

Contributions.

- We introduce **TRUST**, a backpropagation-free Bayesian TTA framework for VLM-based object detection in videos, designed for streaming deployment under continuous distribution drift.
- We adopt STAD Schirmer et al. (2024) work for object detection in global (vMF and Gaussian varieties in Table 1) and instance (TRUST) levels.
- We benchmark the TRUST on SHIFT-Continuous and its precedence over strong training-free (BCA+) and temporal TTA (STAD varieties) baselines, with ablations isolating the impact of each TRUST's module (Table 4).

2 Related Work

TTA adapts a model during inference using only unlabeled test data to mitigate performance drops under distribution shift. A canonical line of work performs lightweight gradient-based updates using entropy minimization or closely related self-supervised objectives, e.g., TTT updates the model using auxiliary self-supervision at test time, TENT adapts normalization statistics via entropy minimization, and follow-ups such as MEMO, EATA, and SAR improve robustness and stability under practical constraints Sun et al. (2020); Wang et al. (2020); Zhang et al. (2022a); Niu et al. (2022; 2023). However, backpropagation-based TTA can be computationally costly for real-time deployment and can become brittle in online regimes due to noisy update signals, sensitivity to protocol/hyperparameters, and cumulative drift (e.g., error accumulation and catastrophic forgetting) in non-stationary streams Wang et al. (2022); Yuan et al. (2023); Gong et al. (2022); Zhao et al. (2023). These concerns motivate training-free or parameter-efficient alternatives that preserve pre-trained knowledge while improving robustness under shift.

While many continual TTA setups model a sequence of discrete domains (often evaluated as staged corruption types or pre-segmented shifts), real deployments often exhibit gradual, time-correlated drift and temporal correlation between samples Wang et al. (2022); Yuan et al. (2023); Gong et al. (2022); Zhao et al. (2023). SHIFT was introduced to explicitly evaluate continuous intra-sequence domain shifts (e.g., rain/fog intensity, time-of-day) for driving perception, making it a natural testbed for temporal adaptation Sun et al. (2022). Schirmer et al. (2024) propose STAD, reframing temporal TTA as Bayesian filtering over time-varying class prototypes via a probabilistic state-space model, enabling robust adaptation under label shift and small batches without requiring labels. For VLMs, BayesTTA formalizes continual-temporal TTA by tracking evolving class-conditional feature distributions and using calibrated Bayesian inference to reduce pseudo-label degradation under continuous drift Cui et al. (2025). These works highlight that temporal information is a signal that shows explicitly that modeling evolution over time improves stability and robustness in continual and temporal TTA.

Extending TTA from classification to detection is challenging due to structured outputs (sets of boxes) and stronger sensitivity to localization errors. Recent detection-focused TTA and Test-Time Adaptive Object Detection (TTAOD) methods explore self-training and stabilization for pseudo-labels (e.g., STFAR), fully test-time per-image updates, and efficiency-aware adaptation mechanisms Chen et al. (2023); Ruan & Tang (2024); Wang et al. (2025). Yoo et al. (2024) further study what, how, and when detectors should update under continually changing domains, proposing lightweight adaptor modules and update-triggering policies to maintain efficiency while avoiding unnecessary updates. In parallel, open-vocabulary vision-language detectors (e.g., Grounding DINO) offer an appealing deployment interface; text queries provide flexible supervision without retraining Liu et al. (2024). However, VLM-based detectors are also vulnerable to domain shift. Gao et al. (2025) propose a foundation-model-powered TTAOD method built on Grounding DINO, using prompt-based mean-teacher updates and an instance dynamic memory to improve pseudo-label quality. While effective, such approaches still rely on online optimization (e.g., prompt tuning) and careful stabilization of pseudo-label learning.

A complementary direction is backpropagation-free adaptation, which keeps the model frozen and performs fast, non-parametric corrections using auxiliary statistics, retrieval, or memory. Cache or template-based TTA has proven effective and efficient in adjacent settings, e.g., Tip-Adapter builds a key-value cache on CLIP features, T3A adjusts class templates at test time, and LAME performs parameter-free online adaptation via structured label propagation Zhang et al. (2022b); Iwasawa & Matsuo (2021); Boudiaf et al. (2022). BCA+ brings this spirit to VLMs and detection under a Bayesian lens. It maintains a dynamic cache that adapts both (i) the likelihood (feature and geometry similarity) and (ii) the prior (historical class frequencies) and fuses cache and base predictions with uncertainty guidance Zhou et al. (2025). This family is particularly attractive for streaming deployment because compute and memory are bounded, updates are interpretable, and the risk of catastrophic forgetting from noisy online gradients is reduced.

Video streams provide additional structure that is largely orthogonal to global domain drift. Object motion is often smooth, and enforcing trajectory consistency can suppress transient false positives and negatives. Classical tracking-by-detection methods such as SORT use a simple Kalman filter motion model and fast association to achieve robust real-time tracking, and DeepSORT further incorporates appearance cues, and ByteTrack remains a strong modern baseline via association of every detection

Bewley et al. (2016); Wojke et al. (2017); Zhang et al. (2022c). Although tracking is typically used for identity maintenance, recent work in related settings shows that temporal consistency signals can also support adaptation under shift (e.g., adapting MOT components using detection consistency cues) Segu et al. (2023).

Motivated by these insights, we summarize that in Table 2, we incorporate a lightweight Kalman-style trajectory filter as an instance-level temporal mechanism that complements global temporal caches. Consistent tracks provide more reliable evidence for updating memories, while short-lived or inconsistent detections are down-weighted to reduce cache corruption.

3 Problem Setup

3.1 Data and Models

All of the used notation and variables through the main paper and appendix are summarized in Table 5.

Let $\mathcal{X}$ denote the space of RGB images and let $\mathcal{B} \subset \mathbb{R}^4$ denote axis-aligned boxes $\boldsymbol{b} = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$ parameterization. We consider C semantic classes with $\mathcal{C} = \{1, \dots, C\}$. An image $\boldsymbol{x} \in \mathcal{X}$ is annotated by a finite set of objects $Y = \{\boldsymbol{y}_k\}_{k=1}^{N} \in \mathcal{Y}, \boldsymbol{y}_k = (\boldsymbol{b}_k, c_k),\ \boldsymbol{b}_k \in \mathcal{B},\ c_k \in \mathcal{C},\ N = |Y| \in \mathbb{N}_0$, with $\Omega = \mathcal{B} \times \mathcal{C}$ and $\mathcal{Y} = \mathcal{F}(\Omega)$ the set of all finite subsets of Ω. We consider detectors defining $p_{\boldsymbol{\theta}}(Y \mid x), Y \in \mathcal{Y},\ \boldsymbol{x} \in \mathcal{X}$, with parameters $\boldsymbol{\theta}$. In practice, a detector returns a finite point estimate

$$\widehat{Y}(\boldsymbol{x}; \boldsymbol{\theta}) = \{\hat{\boldsymbol{y}}_k\}_{k=1}^{\widehat{N}}, \hat{\boldsymbol{y}}_k = (\hat{\boldsymbol{b}}_k, \hat{c}_k, \hat{s}_k), \tag{1}$$

where $\hat{s}_k \in [0, 1]$ is a confidence score and $\widehat{N}$ depends on non-maximum suppression (NMS) and thresholds.

Grounding DINO Liu et al. (2024) conditions detection on a text prompt q and uses a dual-encoder, single-decoder architecture. Given $(\boldsymbol{x}_t, q)$, the image backbone produces visual tokens, $\boldsymbol{V}_t = f_{\boldsymbol{\theta}_{\text{img}}}(\boldsymbol{x}_t) = \{\boldsymbol{v}_{t,i}\}_{i=1}^{L_v} \subset \mathbb{R}^d$, and the text backbone produces prompt tokens $\boldsymbol{U}_t = e_{\boldsymbol{\theta}_{\text{text}}}(q) = \{\boldsymbol{u}_{t,j}\}_{j=1}^{L_u} \subset \mathbb{R}^d$, where L_v and L_u are the numbers of visual and text tokens, respectively. A feature enhancer fuses modalities, $(\tilde{\boldsymbol{V}}_t, \tilde{\boldsymbol{U}}_t) = \text{Enhancer}_{\boldsymbol{\theta}_{\text{enh}}}(\boldsymbol{V}_t, \boldsymbol{U}_t)$. The cross-modality decoder applies L_{dec} refinement layers to an initial set of M object queries $H_t^{(0)} = \{\boldsymbol{h}_{t,m}^{(0)}\}_{m=1}^{M}$ and outputs $H_t^{(L_{\text{dec}})} = \{\boldsymbol{h}_{t,m}^{(L_{\text{dec}})}\}_{m=1}^{M}$. $H_t^{(L_{\text{dec}})} = \text{Decoder}_{\boldsymbol{\theta}_{\text{dec}}}\big(H_t^{(0)}, \tilde{\boldsymbol{V}}_t, \tilde{\boldsymbol{U}}_t\big)$, where L_{dec} is the number of decoder layers. Let $\boldsymbol{\theta}_{\text{img}}, \boldsymbol{\theta}_{\text{text}}, \boldsymbol{\theta}_{\text{enh}}, \boldsymbol{\theta}_{\text{dec}}$ denote the learned parameters of the image backbone, text backbone, feature enhancer, and decoder, respectively.

Each refined query yields a normalized box prediction via a learned head $\hat{\boldsymbol{b}}_{t,m} = g_{\text{box}}\big(\boldsymbol{h}_{t,m}^{(L_{\text{dec}})}\big) \in [0, 1]^4$, and token-level logits over text tokens $\boldsymbol{\ell}_{t,m} = g_{\text{logit}}\big(\boldsymbol{h}_{t,m}^{(L_{\text{dec}})}, \tilde{\boldsymbol{U}}_t\big) \in \mathbb{R}^{L_u}$. We convert $\ell_{t,m,j}$ to per-class probabilities $\boldsymbol{p}_{t,m}^{\text{vlm}} \in \Delta^{C-1}$ via token-span aggregation over phrase token sets $\{\mathcal{I}_c\}_{c=1}^{C}$ (Section B.2), and define the per-query label and confidence as $\hat{c}_{t,m} = \arg\max_c \boldsymbol{p}_{t,m}^{\text{vlm}}(c), \hat{s}_{t,m} = \max_c \boldsymbol{p}_{t,m}^{\text{vlm}}(c)$. Finally, thresholding and NMS yield the detection set $\widehat{Y}_t^{\text{GD}}$ in the unified form of Equation 1. For adaptation modules, we also retain the final decoder query embedding $\boldsymbol{f}_{t,m}^{\text{dec}} \triangleq \boldsymbol{h}_{t,m}^{(L_{\text{dec}})} \in \mathbb{R}^d$ (Section B.2).

3.2 Temporal TTA

Temporal distribution shift. At deployment, the model observes an infinite stream $\{\boldsymbol{x}_t\}_{t=1}^{\infty}$ with $\boldsymbol{x}_t \sim P_t$. Following the temporal TTA setting Schirmer et al. (2024), we assume: (i) $P_t \neq P_{\text{src}}$ (test domain distribution differs from source domain distribution), and (ii) the stream is non-stationary, i.e. there exist $t_1 < t_2$ with $P_{t_1} \neq P_{t_2}$.

Online adaptation protocol. Let $\boldsymbol{\theta}_0$ be the source-trained parameters. At test time, we maintain an adaptation state $\boldsymbol{z}_t \in \mathcal{S}$ that carries information across time (e.g., caches' prototypes, filter states). We write the deployed parameters at time t as $\boldsymbol{\vartheta}_t = (\boldsymbol{\theta}_0, \boldsymbol{z}_t)$, emphasizing that backpropagation-free

methods keep $\boldsymbol{\theta}_0$ fixed while only $\boldsymbol{z}_t$ evolves. Given $(\boldsymbol{x}_t, \boldsymbol{z}_t)$, the detector outputs $\widehat{Y}_t = \widehat{Y}(\boldsymbol{x}_t; \boldsymbol{\vartheta}_t)$ as in equation 1. A temporal TTA algorithm is a stateful mapping $\mathcal{A} : \mathcal{S} \times \mathcal{X} \to \mathcal{S}, \boldsymbol{z}_{t+1} = \mathcal{A}(\boldsymbol{z}_t, \boldsymbol{x}_t; \boldsymbol{\theta}_0)$, without observing ground-truth sets Y_t.

4 METHOD - TRUST

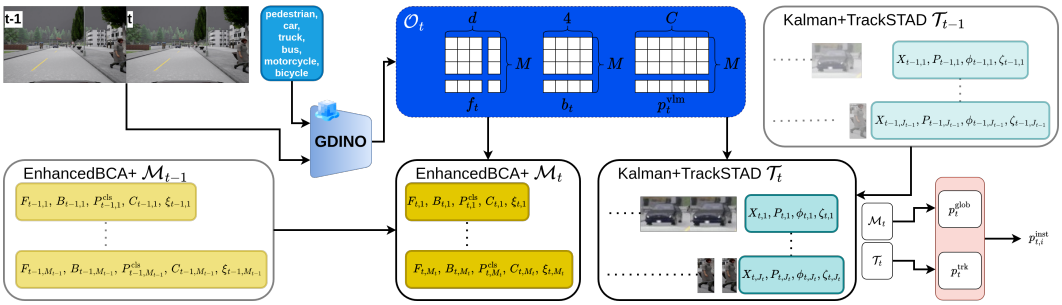

Figure 1: Block diagram of TRUST. A frozen VLM detector outputs candidate tuples $o_{t,i} = (\boldsymbol{f}_{t,i}, \boldsymbol{b}_{t,i}, \boldsymbol{p}_{t,i}^{\text{vlm}}, \hat{s}_{t,i})$. A global cache $\mathcal{M}_{t-1}$ (Enhanced BCA+) produces cache-induced beliefs fused with $\boldsymbol{p}_{t,i}^{\text{vlm}}$ to obtain $\boldsymbol{p}_{t,i}^{\text{glob}}$. A Kalman+TrackSTAD tracker maintains per-track states and yields trajectory-smoothed beliefs $\boldsymbol{p}_{t,j}^{\text{trk}}$. Matched pairs are combined via entropy-weighted fusion to produce refined instance beliefs $\boldsymbol{p}_{t,i}^{\text{inst}}$, while cache and track states are updated online for the next timestep.

4.1 MOTIVATION AND OVERVIEW

We study backpropagation-free TTA for video object detection under gradual domain shift. Two observations motivate our design. The first one is a cache method, BCA+, that adapts predictions using a memory of past detections, but (a) they lack an explicit temporal notion of instance identity and (b) their memory can drift unless it is managed with a lifecycle (aging, confirmation, deletion). Secondly, STAD-style temporal smoothing maintains global per-class state; in detection streams, the update volume is dominated by frequent classes, which can bias the temporal prior and suppress rare classes.

We propose **TRUST**, a two-layer adaptation state with a global, class-conditional cache (Enhanced BCA+) maintaining features and scales that provide priors for newly appearing objects and refine them, and an instance layer that tracks objects and runs per-track STAD (vMF or Gaussian) to smooth class belief along each trajectory.

At time t, the adaptation state decomposes as $z_t = \Big(\mathcal{M}_t, \ \mathcal{T}_t\Big), \mathcal{T}_t = \{\tau_{t,j}\}_{j=1}^{J_t}, \tau_{t,j} = (\boldsymbol{X}_{t,j}, \boldsymbol{P}_{t,j}, \boldsymbol{\phi}_{t,j}, \boldsymbol{\zeta}_{t,j})$, where $\mathcal{M}_t$ is the global memory (cache), and each track j maintains a geometric Kalman state $(\boldsymbol{X}_{t,j}, \boldsymbol{P}_{t,j})$, an appearance feature state $\boldsymbol{\phi}_{t,j}$ (EMA), and a per-track semantic state $\boldsymbol{\zeta}_{t,j}$ (TrackSTAD). They are depicted in Figure 1's Kalman+TrackSTAD blocks. Figure 2, as well, depicts the conditional dependencies of our state underlying our sequential inference and online updates.

Relation to baselines. Compared to the BCA+, TRUST adds tracker-style cache lifecycle management and an instance layer that enforces temporal coherence via trajectory-based identity. Compared to the STAD, TRUST localizes the state-space smoother to tracks, so temporal updates are driven by matched evidence for each object rather than the global class frequency in the stream, which can be dominated by the most frequent class.

4.2 DETECTOR OUTPUTS

We use the frozen detector defined in Section 3.1 and extract per-candidate features/probabilities as in Section B.2. Given a fixed prompt q (Section B.2), the frozen detector returns $\mathcal{O}_t = \{o_{t,i}\}_{i=1}^{N_t}$ with

$o_{t,i} = (\boldsymbol{f}_{t,i},\, \boldsymbol{b}_{t,i},\, \boldsymbol{p}^{\mathrm{vlm}}_{t,i},\, \hat{s}_{t,i})$, where $\boldsymbol{p}^{\mathrm{vlm}}_{t,i} \in \Delta^{C-1}$ and $\hat{s}_{t,i} = \max_c \boldsymbol{p}^{\mathrm{vlm}}_{t,i,c}$. We write $\Delta^{C-1} = \{p \in \mathbb{R}^C : \sum_{c=1}^{C} p_c = 1\}$ for the probability simplex over the C target categories (blue block in Figure 1).

4.3 Global Layer: Enhanced BCA+ Cache

This module instantiates the global latent memory $\mathcal{M}_t$ (Figure 2) and corresponds to the "Enhanced-BCA+" blocks in Algorithm 1 and Figure 1. The cache stores M_t entries, $\mathcal{M}_t = \{e_{t,m}\}_{m=1}^{M_t}$, $\boldsymbol{e}_{t,m} = \big(\boldsymbol{F}_{t,m},\, \boldsymbol{B}_{t,m},\, \boldsymbol{P}^{\mathrm{cls}}_{t,m},\, C_{t,m},\, \xi_{t,m}\big)$. Here $\boldsymbol{F}_{t,m} \in \mathbb{S}^{D-1}$ is a unit feature prototype, $\boldsymbol{B}_{t,m} \in [0,1]^2$ is a normalized box-scale vector (width/height), $\boldsymbol{P}^{\mathrm{cls}}_{t,m} \in \Delta^{C-1}$ is an entry class distribution, and $C_{t,m} \in \mathbb{N}$ is an update count. The meta-state $\xi_{t,m} = (\texttt{hits}, \texttt{age}, \texttt{tsu}, \texttt{state})$ implements tracker-style lifecycle management.

Let $\bar{\boldsymbol{f}}_{t,i} = \boldsymbol{f}_{t,i}/\|\boldsymbol{f}_{t,i}\|_2$ and define the normalized scale map $\psi(b) = \Big(\frac{w_{img}(b)}{W_{img}}, \frac{h_{img}(b)}{H_{img}}\Big) \in [0,1]^2, w_{img}(b) = x_{\max} - x_{\min}, \quad h_{img}(b) = y_{\max} - y_{\min}$, where (W_{img}, H_{img}) are the image dimensions used by the detector input. We compute a hybrid similarity between candidate i and entry m is $\mathrm{sim}_t(i,m) = (1-\omega_s)\,\langle \bar{\boldsymbol{f}}_{t,i}, \boldsymbol{F}_{t-1,m}\rangle + \omega_s\, S_B\big(\boldsymbol{b}_{t,i}, \boldsymbol{B}_{t-1,m}\big)$, in which $S_B\big(b, B\big) = 1 - \frac{\|\psi(b)-B\|_2}{\sqrt{2}}$. Then form a posterior over the entries via a softmax with logit temperature τ_{logit}, $\alpha_{t,i,m} = \frac{\exp\big(\tau_{\mathrm{logit}}\cdot \mathrm{sim}_t(i,m)\big)}{\sum_{m'} \exp\big(\tau_{\mathrm{logit}}\cdot \mathrm{sim}_t(i,m')\big)}$.

The cache induces $\boldsymbol{p}^{\mathrm{cache}}_{t,i} = \sum_{m=1}^{M_{t-1}} \alpha_{t,i,m}\, \boldsymbol{P}^{\mathrm{cls}}_{t-1,m}$. We fuse $\boldsymbol{p}^{\mathrm{cache}}_{t,i}$ with $\boldsymbol{p}^{\mathrm{vlm}}_{t,i}$ using entropy-based confidence weights. The entropy is defined $H(\boldsymbol{p}) = -\sum_{c=1}^{C} p_c \log(p_c + \varepsilon)$. Subsequently, $w_{\mathrm{init}}(t,i) = \exp\big(-H(\boldsymbol{p}^{\mathrm{vlm}}_{t,i})\big), w_{\mathrm{cache}}(t,i) = \exp\big(-H(\boldsymbol{p}^{\mathrm{cache}}_{t,i})\big)$ that construct the global probability $\boldsymbol{p}^{\mathrm{glob}}_{t,i} = \mathrm{Normalize}\Big(w_{\mathrm{init}}\, \boldsymbol{p}^{\mathrm{vlm}}_{t,i} + w_{\mathrm{cache}}\, \boldsymbol{p}^{\mathrm{cache}}_{t,i}\Big)$. This global step is backprop-free and only uses similarities and weighted averaging.

4.4 Instance Layer: Trajectory-Guided Tracking

This module instantiates per-track latent states (Figure 2) and corresponds to the "Kalman+TrackSTAD" blocks in Figure 1. TRUST is trajectory-guided because it refines predictions using track identity induced by Kalman tracking and association.

Each track maintains a 8D state $\boldsymbol{X}_{t,j} = (x_{t,j}, y_{t,j}, z^{(a)}_{t,j}, r_{t,j}, \dot{x}_{t,j}, \dot{y}_{t,j}, \dot{z}^{(a)}_{t,j}, \dot{r}_{t,j})^\top$, where (x,y) is box center, $z^{(a)}$ is area, and r is aspect ratio. We use the standard linear predict and update recursions $(\boldsymbol{F}_{\mathrm{KF}}, \boldsymbol{H}_{\mathrm{KF}}, \boldsymbol{Q}_{\mathrm{KF}}, \boldsymbol{R}_{\mathrm{KF}})$, producing predicted boxes $\hat{\boldsymbol{b}}^{-}_{t,j}$.

Let $\boldsymbol{\phi}_{t-1,j}$ be the track appearance feature (EMA of detection features, L2-normalized). Our implementation supports a ByteTrack-style two-stage association with a combined cost $\mathrm{cost}(i,j) = \lambda_{\mathrm{iou}}\big(1 - \mathrm{IoU}(\boldsymbol{b}_{t,i}, \hat{\boldsymbol{b}}^{-}_{t,j})\big) + \lambda_{\mathrm{feat}}\big(1 - \langle \bar{\boldsymbol{f}}_{t,i}, \boldsymbol{\phi}_{t-1,j}\rangle\big)$, where λ_{iou} and λ_{feat} weight the IoU and appearance terms, $\mathrm{IoU}(\cdot)$ denotes the intersection-over-union between the detection box $\boldsymbol{b}_{t,i}$ and predicted track box $\hat{\boldsymbol{b}}^{-}_{t,j}$, and $\langle\cdot,\cdot\rangle$ denotes cosine similarity. It leverages gating (minimum IoU and/or maximum feature distance), then solve one-to-one matching by Hungarian. This yields matched pairs $\mathcal{P}_t$, unmatched detections $\mathcal{U}_t$, and unmatched tracks $\mathcal{V}_t$. We denote the resulting assignment by $a_{t,i} \in \{0, 1, \ldots, J_t\}$, where $a_{t,i} = j$ if detection i is matched to track j and $a_{t,i} = 0$ otherwise.

4.5 TrackSTAD: Per-Track State-Space Smoothing

To avoid the class-frequency bias of global STAD in detection streams, TRUST runs STAD per track. Each track j maintains a semantic state $\boldsymbol{\zeta}_{t,j}$ and outputs a track belief $\boldsymbol{p}^{\mathrm{trk}}_{t,j} \in \Delta^{C-1}$, which is updated only when the track is matched to a detection. Crucially, TrackSTAD is updated using the detector's raw probabilities $\boldsymbol{p}^{\mathrm{vlm}}_{t,i}$ (pre-fusion) to prevent self-reinforcement. We instantiate TrackSTAD with either (i) a vMF mixture smoother (windowed soft-EM with a transition prior) or (ii) a Gaussian/Kalman smoother; both expose the same interface (predict on misses, update on matches) and differ only in the emission model. We defer full update equations and implementation details to the appendix (Section B.4).

For a matched pair $i \leftrightarrow j$, we refine global probabilities via entropy-weighted fusion, $p_{t,i}^{\text{inst}} = \text{Normalize}\Big(\exp(-H(\boldsymbol{p}_{t,i}^{\text{glob}}))\,\boldsymbol{p}_{t,i}^{\text{glob}} + \exp(-H(\boldsymbol{p}_{t,j}^{\text{trk}}))\,\boldsymbol{p}_{t,j}^{\text{trk}}\Big)$, and set $\boldsymbol{p}_{t,i}^{\text{inst}} = \boldsymbol{p}_{t,i}^{\text{glob}}$ for unmatched detections.

4.6 Track Initialization from Global Cache

When a new track is spawned from an unmatched detection, we optionally initialize its TrackSTAD state from the best-matching cache entry. Concretely, we compute the cache posterior $\alpha_{t,i,\cdot}$, take $m^\star = \arg\max_m \alpha_{t,i,m}$, and if $\alpha_{t,i,m^\star} \geq \tau_{\text{init}}$ we transfer cache state to the track: $\boldsymbol{\rho}_{t,j,c} \leftarrow \text{Normalize}(\boldsymbol{F}_{t,m^\star}) \quad (\forall c), \boldsymbol{\pi}_{t,j} \leftarrow \boldsymbol{P}_{t,m^\star}^{\text{cls}}$, otherwise we initialize purely from the detection (raw) probabilities.

4.7 Memory Update with Tracker-Style Lifecycle Management

Given a detection $(\boldsymbol{f}, \boldsymbol{b}, \boldsymbol{p}^{\text{vlm}}, s)$, we compute the posterior over entries, let $m^\star = \arg\max_m \alpha_m$, and compare $\alpha_{m^\star}$ to a single matching threshold τ_2. Only high-confidence detections ($s \geq \tau_1$) participate in updates. If matched, we update by count-based running by some rules, $F_{m^\star} \leftarrow \text{Normalize}\Big(\frac{C_{m^\star} F_{m^\star} + \bar{f}}{C_{m^\star}+1}\Big), B_{m^\star} \leftarrow \frac{C_{m^\star} B_{m^\star} + \psi(b)}{C_{m^\star}+1}, P_{m^\star}^{\text{cls}} \leftarrow \text{Normalize}\Big(\frac{C_{m^\star} P_{m^\star}^{\text{cls}} + p^{\text{vlm}}}{C_{m^\star}+1}\Big), C_{m^\star} \leftarrow C_{m^\star} + 1$. Otherwise, if capacity allows, we create a new tentative entry.

5 Experiments

5.1 Benchmarks

SHIFT. We evaluate temporal TTA on the SHIFT dataset Sun et al. (2022), which is designed to probe distribution drift (e.g., continuous changes in weather intensity and illumination). Following the temporal setting, we use the validation split of the continuous-shift stream, and report detection performance across SHIFT condition groups (time-of-day $\times$ weather) using mAP@0.5.

Grounding DINO. We report zero-shot performance of Grounding DINO Liu et al. (2024) using post-processing used throughout the Section B.2.

BCA+. We compare against BCA+ Zhou et al. (2025), a backpropagation-free Bayesian cache method that adapts only a bounded state $z_t = \mathcal{M}_t$ while keeping the detector parameters $\boldsymbol{\theta}_0$ frozen. For each frame, we first extract all M GroundingDINO queries before thresholding. Given the cache $\mathcal{M}_t$, we compute a posterior over cached entries using a mixture of cosine feature similarity and normalized box-scale similarity, form a cache-induced probability $\boldsymbol{p}_{t,m}^{\text{cache}}$, and fuse it with $\boldsymbol{p}_{t,m}^{\text{vlm}}$ using entropy-based uncertainty weighting. We then update $\mathcal{M}_t$ with high-confidence post-NMS detections, yielding a streaming, bounded-memory baseline (see Section B.3).

STAD-vMF. We compare against STAD-vMF Schirmer et al. (2024), adapted to detection by treating each detection as a data point with embedding $\tilde{\boldsymbol{f}}_{t,i} \in \mathbb{S}^{d-1}$. Each class maintains a temporally evolving prototype direction $\boldsymbol{\rho}_{t,c} \in \mathbb{S}^{d-1}$, initialized from the class text embedding. Given normalized embeddings, STAD-vMF uses a vMF emission model followed by a softmax. Prototypes are updated online using only confident detections (score $\geq \tau_{\text{upd}}$) via a windowed EM step. Finally, STAD posteriors are fused with detector probabilities (entropy-weighted) to produce temporally smoothed class predictions without updating detector weights (see Section B.4.1).

STAD-Gaussian. We also include the Gaussian STAD variant Schirmer et al. (2024), where each class prototype is tracked as a Gaussian mean $\mu_{t,c} \in \mathbb{R}^d$ (diagonal covariance) updated with a per-class Kalman filter using confident detection embeddings, with optional fixed-lag smoothing. The resulting Gaussian-mixture class posteriors are fused with detector probabilities using the same fusion rule as the STAD-vMF (see Section B.4.2).

Table 1: Detection performance on SHIFT-Continuous (`continuous/val/1x/front`), reported as mAP@0.5 (%). Results are grouped by weather to reflect continuous weather drift. All methods use the same frozen Grounding DINO detector and prompt; adaptation is online and label-free. Average is the mean over all shown conditions (higher is better).

Method	Clear	Cloudy	Foggy	Overcast	Rainy	Average
Vanilla	28.27	20.97	22.52	33.71	27.99	27.03
BCA+	29.90	21.36	22.45	34.09	27.71	27.43
STAD-vMF	29.02	**21.90**	23.30	34.49	28.41	27.73
STAD-Gaussian	29.91	21.62	23.01	**37.38**	28.18	**28.28**
TRUST	**30.47**	21.75	**23.32**	34.94	**28.74**	28.19

5.2 Results

Table 1 reports mAP@0.5 on the SHIFT dataset's continuous domain shift set's daytime weather groups, while Table 3 expands the comparison to all condition groups. We include the direct-test Grounding DINO baseline (Vanilla), BCA+, the adopted version of the two STAD variants, STAD-vMF and STAD-Gaussian, and our method, TRUST. An ablation study on the daytime-clear subset is presented in Table 4 to isolate the impact of each module and its combinations.

Across daytime weather conditions, STAD remains highly effective, but the picture is more nuanced than a single best method. STAD-Gaussian achieves the best daytime average (Table 1), driven primarily by a large gain under overcast drift (37.38 vs. 33.71 for Vanilla), suggesting that prototype and state temporal smoothing in the embedding space is particularly beneficial when the stream exhibits sustained, systematic appearance drift. At the same time, TRUST is competitive with the strongest baselines and achieves the second-best daytime average (28.19), while obtaining the best performance in *clear*, *foggy*, and *rainy* conditions. Compared to Vanilla, TRUST improves mAP by +2.20 (clear), +0.80 (foggy), and +0.75 (rainy), indicating that incorporating instance-level temporal evidence can help recover missed detections under gradual drift and intermittent visibility changes. BCA+ yields modest but generally consistent gains over Vanilla on most daytime conditions, supporting the hypothesis that a global memory of stable prototypes can reduce variance and stabilize open-vocabulary predictions under mild drift.

Ablations (Table 4) further clarify where TRUST's gains originate. Trajectory tracking alone already provides a strong boost on *daytime-clear* (29.51 vs. 28.27), highlighting that temporal aggregation and persistence are valuable even without semantic adaptation. Adding per-track temporal smoothing (TrackSTAD) on top of tracking produces the largest additional gains (up to 30.44), suggesting that the main benefit of our instance layer comes from maintaining a temporally coherent belief per track, rather than solely relying on a global cache. Enhanced BCA+ alone provides a smaller improvement (28.72), but when combined with tracking and per-track smoothing it remains beneficial, with TRUST reaching 30.48 on *daytime-clear*. Overall, these results indicate that global memory and instance-level reasoning are complementary, but their effectiveness depends on the drift regime.

6 Conclusion

We introduced TRUST, a modular, backpropagation-free framework for temporal TTA of VLM-based object detection. TRUST combines a global cache (Enhanced BCA+) with an instance layer that maintains track-level semantic beliefs over time via trajectory tracking and per-track vision embedding temporal smoothing. On SHIFT dataset's continuous domain shift set, strong temporal baselines based on STAD consistently improve over the frozen GroundingDINO detector under several drifting conditions (Table 1, Table 3). Within daytime weather drifts, TRUST is competitive with the best-performing baselines. It attains the best performance in clear, foggy, and rainy conditions and achieves the second-best daytime average, narrowly trailing STAD-Gaussian, which is boosted by a large advantage under overcast drift.

At the same time, the observed gains are still relatively small in absolute terms, and the full setting introduces a large hyperparameter surface that can affect robustness across domains. Our ablations show that the greatest improvements come from the instance pathways (trajectory tracking), while per-instance vision embedding smoothing (TrackSTAD) and the global cache (Enhanced BCA+) provide smaller but complementary benefits. These findings suggest several concrete directions to strengthen TRUST: (i) reduce sensitivity to hyperparameters, (ii) report efficiency metrics (runtime and memory) to quantify the deployment trade-offs of backpropagation-free adaptation, and (iii) broaden evaluation beyond a single VLM-based detector by testing conventional detectors and adding comparisons to gradient-based TTA methods. Finally, while our experiments focus on SHIFT and self-driving videos, the framework is not specific to driving scenes; extending TRUST to other long-horizon drifting streams (including tiny-object and aerial settings) is a natural next step.

REFERENCES

Atif Belal, Heitor R Medeiros, Marco Pedersoli, and Eric Granger. Vlod-tta: Test-time adaptation of vision-language object detectors. *arXiv preprint arXiv:2510.00458*, 2025.

Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*, pp. 3464–3468. Ieee, 2016.

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8344–8353, 2022.

Yijin Chen, Xun Xu, Yongyi Su, and Kui Jia. Stfar: Improving object detection robustness at test-time by self-training with feature alignment regularization. *arXiv preprint arXiv:2303.17937*, 2023.

Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16901–16911, 2024.

Shuang Cui, Jinglin Xu, Yi Li, Xiongxin Tang, Jiangmeng Li, Jiahuan Zhou, Fanjiang Xu, Fuchun Sun, and Hui Xiong. Bayestta: Continual-temporal test-time adaptation for vision-language models via gaussian discriminant analysis. *arXiv preprint arXiv:2507.08607*, 2025.

Fardad Dadboud, Hamid Azad, Varun Mehta, Miodrag Bolic, and Iraj Mantegh. Drift: Autonomous drone dataset with integrated real and synthetic data, flexible views, and transformed domains. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

Yingjie Gao, Yanan Zhang, Zhi Cai, and Di Huang. Test-time adaptive object detection with foundation model. *arXiv preprint arXiv:2510.25175*, 2025.

Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. *Advances in Neural Information Processing Systems*, 34:2427–2440, 2021.

RE Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, D*, 82:35–44, 1960.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10965–10975, 2022.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pp. 38–55. Springer, 2024.

Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019.

Matthias Minderer, Alexey Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pp. 16888–16905. PMLR, 2022.

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

Xiaoqian Ruan and Wei Tang. Fully test-time adaptation for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1038–1047, 2024.

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

Mattia Segu, Bernt Schiele, and Fisher Yu. Darth: Holistic test-time adaptation for multiple object tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9717–9727, 2023.

Tao Sun, Mattia Segu, Janis Postels, Yuxuan Wang, Luc Van Gool, Bernt Schiele, Federico Tombari, and Fisher Yu. Shift: a synthetic driving dataset for continuous multi-task domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21371–21382, 2022.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

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

Kunyu Wang, Xueyang Fu, Xin Lu, Chengjie Ge, Chengzhi Cao, Wei Zhai, and Zheng-Jun Zha. Efficient test-time adaptive object detection via sensitivity-guided pruning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 10577–10586, 2025.

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

Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649. IEEE, 2017.

Jayeon Yoo, Dongkwan Lee, Inseop Chung, Donghyun Kim, and Nojun Kwak. What how and when should object detectors update in continually changing test domains? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23354–23363, 2024.

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

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems*, 35:38629–38642, 2022a.

Renrui Zhang, Wei Zhang, Rongyao Fang, Peng Gao, Kunchang Li, Jifeng Dai, Yu Qiao, and Hongsheng Li. Tip-adapter: Training-free adaption of clip for few-shot classification. In *European conference on computer vision*, pp. 493–510. Springer, 2022b.

Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pp. 1–21. Springer, 2022c.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. On pitfalls of test-time adaptation. *arXiv preprint arXiv:2306.03536*, 2023.

Lihua Zhou, Mao Ye, Shuaifeng Li, Nianxin Li, Jinlin Wu, Xiatian Zhu, Lei Deng, Hongbin Liu, Jiebo Luo, and Zhen Lei. Bayesian test-time adaptation for object recognition and detection with vision-language models. *arXiv preprint arXiv:2510.02750*, 2025.

Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *European conference on computer vision*, pp. 350–368. Springer, 2022.

A Extended Related Work and Additional Context

Table 2: Positioning of test-time adaptation (TTA) methods related to our setting. Temporal: explicit sequential inference beyond frame-wise processing. Detection: targets object detection (not only classification). BP-free: no gradient or backpropagation updates at test time. Update target: internal state updated online (e.g., cache, prototypes, track states). VLM: supports VLM-based detection. TRUST uniquely couples a global cache with per-track state-space inference for trajectory-consistent adaptation.

Method	Temporal	Detection	BP-free	Update target	VLM
TTT Sun et al. (2020)	✗	✗	✗	weights	✗
TENT Wang et al. (2020)	✗	✗	✗	BN/affine	✗
CoTTA Wang et al. (2022)	✗	✗	✗	weights/EMA	✗
NOTE Gong et al. (2022)	✗	✗	✗	weights	✗
STAD Schirmer et al. (2024)	✓	✗	✓	prototypes/filter	✗
Yoo et al. Yoo et al. (2024)	✗	✓	✗	adaptor modules	✗
BCA+ Zhou et al. (2025)	✗	✓	✓	cache	✓
STFAR Chen et al. (2023)	✗	✓	✗	detector	✗
TTAOD-FM Gao et al. (2025)	✗	✓	✗	prompts	✓
TRUST (Ours)	✓	✓	✓	Global cache + per-track SSM	✓

A.1 Broader landscape of TTA objectives and stability mechanisms

Early and widely used TTA objectives optimize prediction confidence without labels, most prominently entropy minimization Wang et al. (2020). To reduce the batch dependence and improve single-sample applicability, augmentation-based objectives minimize marginal entropy across views Zhang et al. (2022a). Subsequent work identifies failure modes—including collapse under hard shifts, sensitivity to hyperparameters, and harmful updates under label shift—and proposes reliability-aware sample selection and anti-forgetting regularization Niu et al. (2022; 2023); Zhao et al. (2023). These analyses support a design principle relevant to our setting: in long, non-i.i.d. streams, adaptation must be conservative and should preferentially update only when evidence is reliable.

A.2 Continual/online adaptation under non-i.i.d. streams

CTTA methods explicitly address non-stationary streams where i.i.d. assumptions break down. CoTTA combines augmentation-averaged predictions with stochastic restoration to slow error accumulation and mitigate forgetting Wang et al. (2022). NOTE further studies non-i.i.d. TTA and

emphasizes robustness to temporally correlated corruption patterns Gong et al. (2022). These methods primarily target classification/segmentation; nevertheless, their failure analyses (drift, confirmation bias, long-horizon degradation) directly transfer to detection streams, motivating memory and filtering mechanisms for stabilizing pseudo-supervision.

A.3 Temporal TTA as probabilistic inference

Temporal TTA can be framed as online inference in a dynamical model. STAD models the evolution of classifier prototypes via a probabilistic state-space model and performs Bayesian filtering to adapt decision boundaries under gradual shift Schirmer et al. (2024). BayesTTA complements this view for VLMs by modeling evolving feature distributions with a Bayesian discriminant analysis perspective and calibrated pseudo-labeling Cui et al. (2025). These works suggest that (i) explicitly representing uncertainty over time and (ii) leveraging temporal correlation can improve stability versus purely frame-wise objectives.

A.4 TTA for object detection

Detection-specific TTA must contend with noisy pseudo-boxes, proposal coupling, and localization-classification interactions. Recent work studies efficient continual TTA for object detection via lightweight adaptor modules and update scheduling Yoo et al. (2024), while fully TTA explores single-image update rules and stabilization tailored to detection noise Ruan & Tang (2024). These methods largely assume closed-vocabulary detectors and typically require parameter updates, which can be expensive and may drift in long videos.

A.5 VLM-based detection and adaptation

Open-vocabulary detectors built on VLM pretraining (e.g., Grounding DINO) enable category-flexible detection through text and improve transfer beyond closed-set paradigms Liu et al. (2024). For VLMs, adaptation strategies have included prompt tuning, parameter-efficient updates, and memory-based calibration. BCA+ proposes a Bayesian training-free cache that adapts likelihood and prior and extends to detection by combining feature and spatial cues Zhou et al. (2025). VLM-powered test-time adaptive detection explores prompt-based mean-teacher updates and an instance dynamic memory to preserve pseudo-label quality Gao et al. (2025). Very recent efforts begin to define TTA specifically for vision-language object detectors and to benchmark common shifts, highlighting prompt selection and proposal-consistency signals Belal et al. (2025). In contrast, our emphasis is on backpropagation-free temporal adaptation where memory updates are governed by probabilistic filtering and temporal consistency rather than gradient steps.

A.6 Tracking-by-detection, Kalman filtering, and why it matters for TTA

Classical tracking-by-detection uses motion models (often Kalman filters) to predict object state, smooth noisy detections, and facilitate data association efficiently Bewley et al. (2016); Wojke et al. (2017). From an adaptation viewpoint, filtering can serve as a principled reliability gate: temporally consistent tracks provide stronger evidence than isolated high-confidence boxes, enabling curated updates to memory/caches without modifying network parameters. This perspective aligns naturally with temporal TTA goals Schirmer et al. (2024) and motivates integrating trajectory-level filtering with VLM-based detection streams.

A.7 Benchmarks and application context

SHIFT provides time-ordered synthetic driving streams with continuous domain factors, making it well-suited for studying gradual shift and temporal adaptation in a controlled setting Sun et al. (2022). Our longer-term target is drone perception, where domain shifts (viewpoint, weather, season, synthetic-to-real) and small-object challenges are prominent; DrIFT provides such factors but with limited temporal smoothness, motivating future evaluation on more video-like drone benchmarks as they become available Dadboud et al. (2025).

B EXPERIMENTAL DETAILS

B.1 SHIFT CONTINUOUS VALIDATION PROTOCOL

We use the SHIFT continuous stream to match the temporal TTA (TempTTA) assumption of intra-sequence gradual drift. Concretely, we evaluate on `continuous/val/1x/front` (continuous shift, validation split, 1x stream, front camera) following the standard SHIFT devkit organization Sun et al. (2022). We report mAP@0.5 on grouped conditions (time-of-day $\times$ weather), consistent with Table 1.

B.2 GROUNDING DINO INFERENCE AND FEATURE EXTRACTION

We use the `transformers` GroundingDINO backend to obtain (i) predicted boxes and text-conditioned class probabilities and (ii) decoder-query embeddings used by BCA+ and STAD. Given C category phrases $\{q_c\}_{c=1}^{C}$, we form a single prompt string $q = q_1 \ . \ q_2 \ . \ \cdots \ . \ q_C \ .$ A forward pass returns (a) decoder-level logits over text tokens, and (b) query-level box predictions for M decoder queries.

Token-span aggregation for per-class probabilities. Let $\ell_{t,m,j}$ denote the raw logit for decoder query m and text token j at time t (returned as `outputs.logits`). Using the tokenizer offset mapping, we identify for each phrase q_c the set of token indices $\mathcal{I}_c$ whose character spans overlap with q_c. We aggregate multi-token phrases by a max operator: $z_{t,m,c} = \max_{j \in \mathcal{I}_c} \ell_{t,m,j}, p_{t,m}^{\mathrm{vlm}} = \mathrm{softmax}\big(z_{t,m,\cdot}\big) \in \Delta^{C-1}$, and define the per-query label and score as $\hat{c}_{t,m} = \arg\max_c p_{t,m}^{\mathrm{vlm}}(c)$ and $\hat{s}_{t,m} = \max_c p_{t,m}^{\mathrm{vlm}}(c)$.

Bounding Boxes. GroundingDINO predicts boxes in normalized center-size form $(c_x, c_y, w_{img}, h_{img}) \in [0,1]^4$. We convert them to pixel-space $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$ using the image width W_{img} and height H_{img}. In `detect`, we then apply a confidence threshold on $\hat{s}_{t,m}$ and NMS to obtain the final set $\widehat{Y}_t^{\mathrm{GD}}$; in `detect_with_features`, we additionally return all M query outputs (before thresholding) for cache/SSM inference.

Decoder query embeddings. Let $f_{t,m}^{\mathrm{dec}} \in \mathbb{R}^d$ denote the decoder query embedding for query m (extracted from the last decoder hidden state; when available we capture the corresponding `vision_hidden_state` via a hook). We ℓ_2-normalize it as $\bar{f}_{t,m}^{\mathrm{dec}} = f_{t,m}^{\mathrm{dec}} / \| f_{t,m}^{\mathrm{dec}} \|_2$.

Hybrid adapter features and text prototypes. Because decoder/query embeddings are not class-specific, we provide adapters with a hybrid representation that concatenates normalized decoder features with normalized semantic probabilities: $\bar{p}_{t,m} = \frac{p_{t,m}^{\mathrm{vlm}}}{\|p_{t,m}^{\mathrm{vlm}}\|_2}, \tilde{f}_{t,m} = \Big[(1 - \alpha_{\mathrm{mix}})\, \bar{f}_{t,m}^{\mathrm{dec}} \ ; \ \alpha_{\mathrm{mix}}\, \bar{p}_{t,m}\Big] \in \mathbb{R}^{d+C}$, with mixing weight $\alpha_{\mathrm{mix}} \in [0,1]$. For prototype-based temporal models, we also extract per-class text embeddings by averaging token embeddings over the same span: $e_c^{\mathrm{text}} = \frac{1}{|\mathcal{I}_c|} \sum_{j \in \mathcal{I}_c} u_{t,j} \in \mathbb{R}^d, \bar{e}_c^{\mathrm{text}} = \frac{e_c^{\mathrm{text}}}{\|e_c^{\mathrm{text}}\|_2}$, and form a matching hybrid text prototype $\tilde{e}_c = \Big[(1 - \alpha_{\mathrm{mix}})\, \bar{e}_c^{\mathrm{text}} \ ; \ \alpha_{\mathrm{mix}}\, \bar{e}_c^{\mathrm{1hot}}\Big] \in \mathbb{R}^{d+C}$, where $e_c^{\mathrm{1hot}} \in \mathbb{R}^C$ is the one-hot vector for class c. We return both raw and hybrid features/prototypes for reproducibility and ablations.

B.3 BCA+ IMPLEMENTATION DETAILS

We implement BCA+ as a backpropagation-free online state $z_t = \mathcal{M}_t$ that stores a bounded cache of recent high-confidence detections and corrects class probabilities at test time while keeping θ_0 fixed. At each time t, before applying any confidence threshold, the GroundingDINO detector yields M query-level proposals $\big\{(\tilde{f}_{t,m}, \hat{b}_{t,m}, p_{t,m})\big\}_{m=1}^{M}$, where $\hat{b}_{t,m} \in \mathcal{B}$ and $p_{t,m} \in \Delta^{C-1}$. We emphasize that we run cache inference on all M proposals and only threshold afterward.

Hybrid cache embedding. Because decoder/query embeddings in GroundingDINO are not class-specific, we use a hybrid embedding that concatenates the normalized decoder feature with the

normalized text-conditioned class distribution: $\tilde{f}_{t,m} = \Big[(1-\alpha)\,\bar{f}_{t,m}\;;\;\alpha\,\bar{p}_{t,m}\Big] \in \mathbb{R}^{d+C}$, $\bar{f}_{t,m} = \frac{f_{t,m}}{\|f_{t,m}\|_2}$, $\bar{p}_{t,m} = \frac{p_{t,m}}{\|p_{t,m}\|_2}$, with $\alpha \in [0,1]$.

Cache state. The cache at time t contains M_t entries $\mathcal{M}_t = \{(f^{(m)},\, s^{(m)},\, p^{\mathrm{cls}^{(m)}},\, c^{(m)})\}_{m=1}^{M_t}$, where $f^{(m)} \in \mathbb{R}^{d+C}$ is a normalized feature, $s^{(m)} \in [0,1]^2$ stores normalized box scale, $p^{\mathrm{cls}^{(m)}} \in \Delta^{C-1}$ is the stored pseudo-label distribution, and $c^{(m)} \in \mathbb{N}$ is an update count. For a box $\hat{b}_{t,m} = (x_{\min}, y_{\min}, x_{\max}, y_{\max})$, we define normalized scale $\psi_{t,m} = \Big(\frac{w_{img_{t,m}}}{W_{img}}, \frac{h_{img_{t,m}}}{H_{img}}\Big), w_{img_{t,m}} = x_{\max} - x_{\min},\;\; h_{img_{t,m}} = y_{\max} - y_{\min}$, with image size (W_{img}, H_{img}).

Posterior over cache entries. For each proposal m, we compute feature similarity and scale similarity to each cache entry: $S_F(m,m') = \langle \bar{\tilde{f}}_{t,m}, f^{(m')} \rangle, S_B(m,m') = 1 - \frac{\|\psi_{t,m} - \psi^{(m')}\|_2}{\sqrt{2}}$, and combine them with a mixing weight $\omega_s \in [0,1]$ to form a likelihood score $\ell_{m'} = (1-\omega_s)\, S_F(m,m') + \omega_s\, S_B(m,m')$. The posterior over cache indices is then $P(U = m' \mid x_t, m) = \mathrm{softmax}\big(\tau_{\mathrm{logit}}\, \ell_{m'}\big)$, where $\tau_{\mathrm{logit}} > 0$ is a logit-temperature.

Cache-based correction. We form the cache-induced class probabilities by posterior averaging of cached pseudo-label distributions: $p_{t,m}^{\mathrm{cache}} = \sum_{m'=1}^{M_t} P(U = m' \mid x_t, m)\, p^{\mathrm{cls}^{(m')}}$.

Uncertainty-weighted fusion. We fuse the base and cache-induced probabilities using entropy-based uncertainty weighting: $E(p) = -\sum_{c=1}^{C} p(c) \log(p(c)+\epsilon), w_{\mathrm{init}} = \exp\big(-E(p_{t,m})\big), w_{\mathrm{cache}} = \exp\big(-E(p_{t,m}^{\mathrm{cache}})\big)$, $p_{t,m}^{\mathrm{fuse}} = \frac{w_{\mathrm{init}}\, p_{t,m} + w_{\mathrm{cache}}\, p_{t,m}^{\mathrm{cache}}}{w_{\mathrm{init}} + w_{\mathrm{cache}}}$. We set the final per-proposal score and label as $\hat{s}_{t,m} = \max_c p_{t,m}^{\mathrm{fuse}}(c)$ and $\hat{c}_{t,m} = \arg\max_c p_{t,m}^{\mathrm{fuse}}(c)$. Afterward, we apply a confidence threshold and NMS (IoU threshold from the detector config) to obtain the final detection set $\widehat{Y}_t$ in the unified form of equation 1.

Cache initialization. When the cache is empty, we initialize $\mathcal{M}_t$ from the first frame's high-confidence post-NMS detections by greedy clustering within each predicted class. We assign a detection to an existing cluster if its combined similarity exceeds a stricter initialization threshold τ_2^{init} and the cluster size is below a maximum; otherwise we start a new cluster. Each cluster produces one cache entry using the (normalized) centroid feature and the mean class-probability vector, with count set to the cluster size.

Cache update. Let τ_1 be the high-confidence threshold used to decide which post-NMS detections enter the cache update. For each selected detection, we reuse the pre-computed posterior $P(U \mid x_t, m)$ (computed with a frozen snapshot of the cache at the start of the frame), find $m^\star = \arg\max_{m'} P(U = m' \mid x_t, m), s^\star = P(U = m^\star \mid x_t, m)$. If $s^\star \geq \tau_2$, we update entry $m^\star$ by count-based averaging: $f^{(m^\star)} \leftarrow \mathrm{norm}\Big(\frac{c^{(m^\star)} f^{(m^\star)} + \bar{\tilde{f}}_{t,m}}{c^{(m^\star)}+1}\Big), s^{(m^\star)} \leftarrow \frac{c^{(m^\star)} s^{(m^\star)} + s_{t,m}}{c^{(m^\star)}+1}, p^{\mathrm{cls}^{(m^\star)}} \leftarrow \frac{c^{(m^\star)} p^{\mathrm{cls}^{(m^\star)}} + p_{t,m}^{\mathrm{vlm}}}{c^{(m^\star)}+1}$, and increment $c^{(m^\star)}$. If $s^\star < \tau_2$, we create a new cache entry, subject to a maximum cache size; when full, we fall back to updating $m^\star$. This yields a bounded-memory, streaming adaptation baseline with no gradient updates.

B.4 STAD IMPLEMENTATION DETAILS

Detection-to-STAD mapping. At test time t, the detector produces a set of $\widehat{N}_t$ candidate detections (after thresholding and NMS), each with (i) a projected embedding $f_{t,i} \in \mathbb{R}^d$ in the same space as the class text embeddings, (ii) detector class probabilities $p_{t,i}^{\mathrm{vlm}} \in \Delta^{C-1}$, and (iii) a detection confidence score $s_{t,i}$ as the raw VLM score before the fusion. We normalize embeddings to the unit sphere: $\tilde{f}_{t,i} = f_{t,i}/\|f_{t,i}\|$. STAD operates on the set $\{\tilde{f}_{t,i}\}_{i=1}^{M_t}$, interpreting each detection embedding as one "data point" in the mixture model at time t. Only detections with $s_{t,i} \geq \tau_{\mathrm{upd}}$ are used to update the temporal state.

Fusion protocol and anti-drift rule. Let $p_{t,i}^{\text{ssm}}$ be the STAD posterior over classes for detection i at time t. We form fused probabilities $p_{t,i}^{\text{fuse}}$ either by a fixed convex combination $p_{t,i}^{\text{fuse}} = \lambda\, p_{t,i}^{\text{ssm}} + (1-\lambda)\, p_{t,i}^{\text{vlm}}$, or by entropy-weighted fusion: $w_{t,i}^{\text{vlm}} = \exp(-H(p_{t,i}^{\text{vlm}}))$, $w_{t,i}^{\text{ssm}} = \exp(-H(p_{t,i}^{\text{ssm}}))$, $p_{t,i}^{\text{fuse}} = \frac{w_{t,i}^{\text{vlm}} p_{t,i}^{\text{vlm}} + w_{t,i}^{\text{ssm}} p_{t,i}^{\text{ssm}}}{w_{t,i}^{\text{vlm}} + w_{t,i}^{\text{ssm}}}$. Crucially, to avoid self-reinforcement, we update STAD using the detector's raw probabilities $p_{t,i}^{\text{vlm}}$ (pre-fusion), not $p_{t,i}^{\text{fuse}}$.

B.4.1 STAD-vMF

Each class c maintains a vMF variational posterior parameterized by a mean direction $\rho_{t,c} \in \mathbb{S}^{d-1}$ and concentration $\gamma_{t,c}$, and (optionally) a mixing coefficient $\pi_{t,c}$. Initialization uses the normalized text embedding $\rho_{0,c} = \text{norm}(e_c^{\text{text}})$ and a fixed $\gamma_{0,c} = \gamma_{\text{init}}$, with uniform $\pi_{0,c} = 1/C$.

Prediction. We compute the expected prototype $\mathbb{E}[w_{t,c}] = A_d(\gamma_{t,c})\, \rho_{t,c}$, and form logits for each detection embedding $\tilde{f}_{t,i}$: $\ell_{t,i,c}^{\text{ssm}} = \frac{\kappa^{\text{ems}}}{T} \langle \tilde{f}_{t,i}, \mathbb{E}[w_{t,c}] \rangle + \mathbf{1}[\texttt{use\_pi}] \log(\pi_{t,c})$. The STAD posterior is $p_{t,i}^{\text{ssm}} = \text{softmax}(\ell_{t,i,\cdot}^{\text{ssm}})$.

Windowed EM update. We maintain a sliding window of the last W_{win} frames of confident embeddings and their soft assignments (stored as probabilities). Let $\mathcal{D}_t = \{(\tilde{f}_n,\, p_n^{\text{vlm}},\, \omega_n)\}_{n=1}^{N}$ be the pooled windowed set, where $\omega_n \in (0, 1]$ is a temporal weight that increases for more recent frames (linear recency weighting). We perform I EM iterations:

E-step (soft responsibilities). Compute SSM posteriors from current state, then optionally combine with the detector's raw probabilities through a geometric mixture (VLM prior weight η): $\lambda_{n,c} \propto \left(p_{n,c}^{\text{vlm}}\right)^{\eta} \left(p_{n,c}^{\text{ssm}}\right)^{1-\eta}$, $\lambda_{n,\cdot} = \text{softmax}\Big(\eta \log p_{n,\cdot}^{\text{vlm}} + (1-\eta) \log p_{n,\cdot}^{\text{ssm}}\Big)$. We apply temporal weights: $\tilde{\lambda}_{n,c} = \omega_n\, \lambda_{n,c}$.

M-step Let $R_c = \sum_n \tilde{\lambda}_{n,c}$ be the effective count. We update a class only if $R_c \geq N_{\min}$ (`min_updates_per_class`). To reduce collapse under heavy class imbalance, we cap $R_c \leq 0.5N$ before updating and before computing π. The weighted sufficient statistic is $s_c = \sum_n \tilde{\lambda}_{n,c}\, \tilde{f}_n$. We incorporate a transition prior (prototype inertia) using global κ^{trans}: $\beta_{t,c} = \kappa^{\text{ems}}\, s_c + \kappa^{\text{trans}}\, \mathbb{E}[w_{t-1,c}]$, $\rho_{t,c} \leftarrow \text{norm}(\beta_{t,c})$. We update $\gamma_{t,c}$ via the mean resultant length estimate with clamping for stability: $\bar{r}_c = \frac{\|\beta_{t,c}\|}{\kappa^{\text{ems}} R_c + \kappa^{\text{trans}}}$, $\bar{r}_c \leftarrow \min(\bar{r}_c,\, 0.95)$, $\gamma_{t,c} \leftarrow \text{EMA}\big(\gamma_{t-1,c},\, A_d^{-1}(\bar{r}_c)\big)$, and clamp $\gamma_{t,c} \in [\gamma_{\min}, \gamma_{\max}]$. Mixing coefficients are updated with a Dirichlet prior α_0: $\pi_{t,c} \leftarrow \frac{R_c + \alpha_0}{\sum_{c'}(R_{c'} + \alpha_0)}$, optionally EMA-smoothed. Optionally, global κ^{ems} and κ^{trans} can be updated with an EMA, but we keep this disabled by default due to instability.

B.4.2 STAD-Gaussian

The Gaussian variant tracks each class prototype as a Gaussian mean $\mu_{t,c} \in \mathbb{R}^d$ with covariance $P_{t,c}$ (diagonal for efficiency), initialized from text embeddings. For a detection embedding $\tilde{f}_{t,i}$, we compute mixture logits using an isotropic emission covariance $R = \sigma^2 I$ with $\sigma^2 = \texttt{r\_base}$: $\log p(\tilde{f}_{t,i} \mid c) = -\frac{1}{2\sigma^2} \|\tilde{f}_{t,i} - \mu_{t,c}\|_2^2 + \mathbf{1}[\texttt{use\_pi}] \log(\pi_{t,c})$, $p_{t,i}^{\text{ssm}} = \text{softmax}(\log p(\tilde{f}_{t,i} \mid \cdot))$. For updates, we compute responsibilities (from raw detector probabilities or p^{ssm} if absent), then per class c with $R_c = \sum_i \lambda_{i,c} \geq N_{\min}$ we form a class observation (responsibility-weighted mean) $y_c = \sum_i \lambda_{i,c} \tilde{f}_{t,i} / R_c$ and perform a diagonal Kalman update with process noise $Q = \texttt{q\_scale} \cdot I$: $P_{t,c}^{-} = P_{t-1,c} + Q$, $R_c^{\text{obs}} = \sigma^2 / R_c$, $K_{t,c} = P_{t,c}^{-} \oslash (P_{t,c}^{-} + R_c^{\text{obs}})$, $\mu_{t,c} = \mu_{t-1,c} + K_{t,c} \odot (y_c - \mu_{t-1,c})$, $P_{t,c} = (1 - K_{t,c}) \odot P_{t,c}^{-}$. We renormalize $\mu_{t,c}$ to the unit sphere for consistency with the embedding geometry and update $\pi_{t,c}$ with a Dirichlet prior and EMA smoothing. Optionally, we apply fixed-lag Rauch–Tung–Striebel (RTS) smoothing over the last W_{win} states.

Hyperparameters. Key hyperparameters are: update threshold τ_{upd}, window size W_{win}, EM iterations I, temperature T, vMF globals $\kappa^{\text{ems}}, \kappa^{\text{trans}}$, vMF bounds $(\gamma_{\min}, \gamma_{\max})$ and vMF VLM-prior weight η, Dirichlet prior α and EMA decays for π (and optionally γ), and for Gaussian q_scale, $\sigma^2 = \texttt{r\_base}$, and optional RTS smoothing.

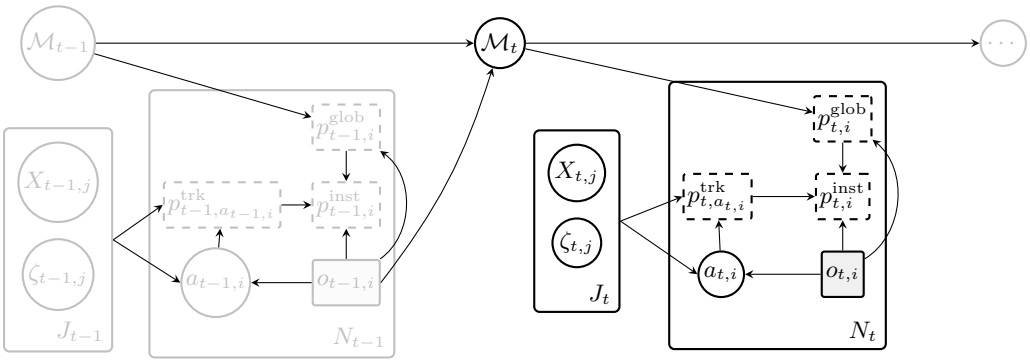

Figure 2: Graphical model of TRUST at time t. Shaded nodes are observed (frame x_t and detector proposals), while latent variables include the global cache state $\mathcal{M}_t$ and per-track states. Plates indicate repetition over detections i and tracks j; arrows encode dependencies used for sequential inference and online updates.

Algorithm 1 TRUST: Global–Instance Backprop-Free Temporal TTA

Require: Stream $\{\boldsymbol{x}_t\}_{t=1}^{T}$, frozen detector $\boldsymbol{\theta}_0$, cache $\mathcal{M}_0$, tracks $\mathcal{T}_0 = \emptyset$

for $t = 1$ to T **do**
  $\mathcal{O}_t = \{o_{t,i}\}_{i=1}^{N_t^{\text{cand}}} \leftarrow \textsc{Detect}(\boldsymbol{x}_t; \boldsymbol{\theta}_0)$
  for all $i \in \{1, \ldots, N_t^{\text{cand}}\}$ **do**
    $\boldsymbol{p}_{t,i}^{\text{glob}} \leftarrow \textsc{EnhancedBCA+}(\mathcal{M}_{t-1}, \boldsymbol{f}_{t,i}, \boldsymbol{b}_{t,i}, \boldsymbol{p}_{t,i}^{\text{vlm}})$
  end for
  $\widehat{Y}_t^{\text{glob}} \leftarrow \text{NMS}\big(\{(i, \boldsymbol{f}_{t,i}, \boldsymbol{b}_{t,i}, \boldsymbol{p}_{t,i}^{\text{glob}})\}\big)$
  $\textsc{Kalman + TrackSTAD\_Predict}(\mathcal{T}_{t-1})$
  $(\mathcal{P}_t, \mathcal{U}_t, \mathcal{V}_t) \leftarrow \textsc{Associate}(\widehat{Y}_t^{\text{glob}}, \mathcal{T}_{t-1})$
  for all $(i, j) \in \mathcal{P}_t$ **do**
    $\mathcal{T}_t \leftarrow \textsc{KalmanUpdate}(\tau_{t-1,j}; \boldsymbol{b}_{t,i})$
    $\mathcal{T}_t \leftarrow \textsc{UpdateAppearance}(\tau_{t-1,j}; \boldsymbol{f}_{t,i})$
    $\mathcal{T}_t \leftarrow \textsc{TrackSTADUpdate}(\tau_{t-1,j}; \boldsymbol{p}_{t,i}^{\text{vlm}})$
    $p_{t,i}^{\text{inst}} \leftarrow \textsc{EntropyFuse}\big(\boldsymbol{p}_{t,i}^{\text{glob}}, \boldsymbol{p}_{t,j}^{\text{trk}}\big)$
  end for
  $\textsc{SpawnNewTracks}(\mathcal{U}_t; \mathcal{M}_{t-1})$
  $\textsc{TrackLifecycle}(\mathcal{T}_t, \mathcal{V}_t)$ {age/lost/remove}
  $\mathcal{M}_t \leftarrow \textsc{UpdateCache}(\mathcal{M}_{t-1}, \mathcal{O}_t)$
  $\textsc{CacheStep}(\mathcal{M}_t)$ {age + stale cleanup}
end for

B.5 Baseline plumbing and shared detector wrapper

All baselines share the same Grounding DINO forward pass and differ only in the adaptation state update and probability fusion. Our code routes through a single `detect_with_features` interface that returns boxes, per-class probabilities, and query/text embeddings, which are consumed by (i) the BCA+ cache adapter and (ii) the STAD-vMF temporal adapter.

C Method - TRUST

D Compute Resources

We conducted large-scale grid-search experiments on Digital Research Alliance of Canada (DRAC) clusters and a local workstation. The clusters were Narval (32 CPU cores and 4 NVIDIA A100

Table 3: Detection performance on SHIFT-Continuous (`continuous/val/1x/front`), reported as mAP@0.5 (%). Results are grouped by time-of-day and weather to reflect continuous illumination and weather drift. All methods use the same frozen Grounding DINO detector and prompt; adaptation is online and label-free. Average is the mean over all shown conditions (higher is better).

Condition	Vanilla	BCA+	STAD-vMF	STAD-G	TRUST
Daytime					
clear	28.27	29.90	29.02	29.91	30.48
cloudy	20.97	21.36	21.90	21.62	21.75
foggy	22.52	22.45	23.30	23.01	23.32
overcast	33.71	34.09	34.49	37.38	34.94
rainy	27.99	27.71	28.41	28.18	28.74
Night					
clear	34.68	35.13	34.42	34.69	−−
cloudy	17.94	18.46	17.03	17.01	−−
foggy	19.34	19.03	19.93	20.56	−−
rainy	22.37	23.16	23.32	24.03	−−
Dawn/Dusk					
clear	27.42	26.98	27.47	27.60	−−
cloudy	26.90	27.63	28.94	28.52	−−
foggy	23.31	23.60	22.20	22.74	−−
rainy	40.22	41.12	40.46	40.51	−−
Average	26.29	26.61	26.80	27.23	−−

40GB GPUs.[1]), Nibi (112 CPU cores and 8 NVIDIA H100 80GB GPUs.[2]), Fir (48 CPU cores and 4 NVIDIA H100 80GB GPUs.[3]), Trillium (96 CPU cores and 4 NVIDIA H100 80GB GPUs.[4]), Rorqual (64 CPU cores and 4 NVIDIA H100 80GB GPUs.[5]), and a local workstation with a NVIDIA GeForce RTX 3090 (24GB VRAM) GPU.

E Discussion of full results

Table 3 expands the comparison to all time-of-day groups. Across condition groups, STAD variants provide the most consistent improvements over Vanilla, with STAD-Gaussian achieving the best overall average. Notably, STAD-vMF improves under several challenging drifts (e.g., overcast and some dawn/dusk conditions), supporting the hypothesis that temporal state-space smoothing in an embedding/prototype space is effective for SHIFT's gradual non-stationarity. BCA+ provides smaller but generally positive gains, consistent with the view that maintaining a global memory of stable prototypes can reduce prediction variance under mild drift.

For TRUST, we report results for the daytime weather groups in Table 1 and leave non-daytime groups as future evaluation. Within the evaluated daytime regime, TRUST's improvements are concentrated in conditions where temporal persistence and per-track belief maintenance are most beneficial (clear/foggy/rainy), while STAD-Gaussian remains strongest under heavily shifted overcast sequences. Overall, these results suggest that combining global memory with instance-level temporal reasoning is promising but not uniformly dominant across drift types; closing the remaining gap requires systematically improving robustness across regimes and reducing dependence on large hyperparameter grids.

[1] `https://docs.alliancecan.ca/wiki/Narval`
[2] `https://docs.alliancecan.ca/wiki/Nibi`
[3] `https://docs.alliancecan.ca/wiki/Fir`
[4] `https://docs.alliancecan.ca/wiki/Trillium`
[5] `https://docs.alliancecan.ca/wiki/Rorqual`

Table 4: Ablation study on SHIFT-Continuous (`continuous/val/1x/front`) for the daytime-clear domain, reported as mAP@0.5 (%). We evaluate the contribution of each TRUST component—Enhanced BCA+ (global cache), per-track temporal adaptation (TrackSTAD-vMF / TrackSTAD-Gaussian), and trajectory tracking—as well as their combinations. Higher is better.

mAP	Enhanced BCA+	TrackSTAD-vMF	TrackSTAD-Gaussian	Trajectory Tracking
28.27				
29.51				✓
28.72	✓			
28.78			✓	
28.61		✓		
30.44			✓	✓
30.09		✓		✓
30.48	✓	✓		✓
30.37	✓		✓	✓

Table 5: Notation used throughout the paper.

Symbol	Meaning
Indices, counts, and dimensions	
t	Time index in the test stream.
T	Stream length in a finite evaluation (used as $\{x_t\}_{t=1}^{T}$).
i	Detection index within a set of candidate observations/detections at time t.
j	Track index (active track j at time t).
k	Generic detection element index in the unified output set $\widehat{Y} = \{\hat{y}_k\}_{k=1}^{\widehat{N}}$.
m	(i) GroundingDINO decoder query index, and (ii) cache-entry index (disambiguated by context / subscripts).
c	Class index; $c \in \{1, \dots, C\}$.
C	Number of class phrases in the prompt / number of classes.
L_v	Number of visual tokens produced by the image backbone.
L_u	Number of text tokens produced by the text backbone / tokenizer.
L_{dec}	Number of decoder refinement layers in GroundingDINO.
M	Number of decoder object queries in GroundingDINO.
M_t	Cache size at time t (number of entries in $\mathcal{M}_t$).
$\widehat{N}$	Predicted cardinality after post-processing (e.g., thresholding + NMS).
D	Dimension of the features used by cache/SSM (e.g., d or $d{+}C$ depending on the chosen representation).
d	Embedding dimension of token/query features (decoder/text embeddings).
W_{img}, H_{img}	Image width and height.
Spaces, sets, and distributions (RFS view)	
x_t	Input image/frame at time t.
P_{src}	Source (training) distribution.
P_t	Test-time distribution at time t (possibly non-stationary).
Y_t	Ground-truth set of objects at time t (random finite set).
$\widehat{Y}_t$	Predicted detection set at time t (finite set).
$\hat{y}_k$	A predicted element: $\hat{y}_k = (\hat{b}_k, \hat{c}_k, \hat{s}_k)$.
$\mathcal{B}$	Bounding-box space (e.g., $[0,1]^4$ normalized boxes or pixel-space boxes).
$\mathcal{C}$	Label space / set of classes (typically $\{1, \dots, C\}$).
Ω	Single-object state space, typically $\Omega = \mathcal{B} \times \mathcal{C}$ (as written).

(continued)

Symbol	Meaning
$\mathcal{F}(\Omega)$	Set of all finite subsets of Ω (finite-set operator).
Δ^{C-1}	Probability simplex over C classes.
$\mathbb{S}^{D-1}$	Unit sphere in $\mathbb{R}^D$ (for L2-normalized features).
$\mathbb{N}$, $\mathbb{N}_0$	Positive / nonnegative integers.
Unified detection output and post-processing	
$\hat{b}_k$	Predicted bounding box.
$\hat{c}_k$	Predicted class label.
$\hat{s}_k$	Predicted confidence score (scalar in $[0, 1]$).
$\mathrm{NMS}(\cdot)$	Non-maximum suppression operator.
$\mathrm{IoU}(b, b')$	Intersection-over-union between boxes b and b'.
$\langle a, b\rangle$	Cosine similarity when a, b are L2-normalized (dot product).
GroundingDINO (VLM detector) notation	
q	Natural-language prompt string (concatenated class phrases).
q_c	Class phrase for class c (used to form the overall prompt).
$V_t = \{v_{t,i}\}_{i=1}^{L_v}$	Visual tokens from the image backbone at time t.
$U_t = \{u_{t,j}\}_{j=1}^{L_u}$	Text tokens from the text backbone at time t.
$(\tilde{V}_t, \tilde{U}_t)$	Enhanced/fused tokens after the feature enhancer module.
$H_t^{(0)} = \{h_{t,m}^{(0)}\}_{m=1}^{M}$	Initial set of M object queries.
$H_t^{(L_{\mathrm{dec}})} = \{h_{t,m}^{(L_{\mathrm{dec}})}\}_{m=1}^{M}$	Final refined queries after L_{dec} decoder layers.
$f_{\theta_{\mathrm{img}}}(\cdot)$	Image backbone with parameters θ_{img}.
$e_{\theta_{\mathrm{text}}}(\cdot)$	Text backbone with parameters θ_{text}.
$\mathrm{Enhancer}_{\theta_{\mathrm{enh}}}(\cdot)$	Feature enhancer with parameters θ_{enh}.
$\mathrm{Decoder}_{\theta_{\mathrm{dec}}}(\cdot)$	Cross-modality decoder with parameters θ_{dec}.
$g_{\mathrm{box}}(\cdot)$	Box regression head producing normalized boxes.
$g_{\mathrm{logit}}(\cdot)$	Head producing token-level logits over text tokens.
$\hat{b}_{t,m} \in [0, 1]^4$	Normalized box predicted for query m at time t.
$\ell_{t,m,j}$	Raw logit for query m and text token j (time t).
$\mathcal{I}_c$	Token index set corresponding to phrase span of class c.
$z_{t,m,c}$	Aggregated phrase logit for class c from token-span aggregation.
$p_{t,m}^{\mathrm{vlm}} \in \Delta^{C-1}$	Per-query class probability vector (after aggregation + softmax).
$\hat{c}_{t,m}$, $\hat{s}_{t,m}$	Per-query predicted class and confidence.
$f_{t,m}^{\mathrm{dec}} = h_{t,m}^{(L_{\mathrm{dec}})}$	Final decoder-query embedding (used by cache/SSM).
$\bar{f}_{t,m}^{\mathrm{dec}}$	L2-normalized decoder-query embedding.
$\bar{p}_{t,m}$	L2-normalized probability vector (optional hybrid feature).
$\tilde{f}_{t,m} \in \mathbb{R}^{d+C}$	Hybrid feature (concatenated normalized decoder feature and probs).
e_c^{text}, $\bar{e}_c^{\mathrm{text}}$	(Normalized) text prototype for class c (token-span average).
$\tilde{e}_c \in \mathbb{R}^{d+C}$	Hybrid text prototype (optional).
α_{mix}	Mixing weight in hybrid feature/prototype concatenation.
Temporal TTA protocol	
θ_0	Source-trained (frozen) parameters.
z_t	Adaptation state carried across time (caches / filter states / prototypes / etc.).
$\vartheta_t = (\theta_0, z_t)$	Deployed parameterization at time t.
$\mathcal{A}(\cdot)$	Temporal adaptation mapping: $z_{t+1} = \mathcal{A}(z_t, x_t; \theta_0)$.
Global memory (Enhanced BCA+)	
$\mathcal{M}_t$	Global cache / memory at time t.
$e_{t,m}$	Cache entry m at time t.
$F_{t,m} \in \mathbb{S}^{D-1}$	Unit feature prototype stored in entry m (L2-normalized).
$B_{t,m} \in [0, 1]^2$	Normalized box-scale vector (width/height).

(continued)

Symbol	Meaning
$P_{t,m}^{\mathrm{cls}} \in \Delta^{C-1}$	Stored class distribution (entry-level).
$C_{t,m} \in \mathbb{N}$	Update count for entry m.
$\xi_{t,m}$	Meta-state for lifecycle (e.g., `hits`, `age`, `tsu`, `state`).
$\mathcal{O}_t = \{o_{t,i}\}_{i=1}^{N_t^{\mathrm{cand}}}$	Candidate observations/detections for cache inference at time t.
$o_{t,i} = (\tilde{f}_{t,i}, b_{t,i}, p_{t,i}^{\mathrm{vlm}}, \hat{s}_{t,i})$	Observation tuple: feature, box, class probs, confidence.
$b_{t,i}$	A detection box used for cache similarity and tracking.
$\psi(b)$	Box-scale map.
S_F	Feature similarity (cosine): $S_F = \langle \tilde{f}_{t,i}, F_{t-1,m} \rangle$.
S_B	Scale similarity: $S_B = 1 - \|\psi(b_{t,i}) - B_{t-1,m}\|_2 / \sqrt{2}$.
S_C	Class similarity: $S_C = \langle p_{t,i}^{\mathrm{vlm}}, P_{t-1,m}^{\mathrm{cls}} \rangle$.
ω_s, ω_c	Weights for scale and class terms in the combined similarity logit.
τ_{logit}	Temperature applied to similarity logits before softmax.
$\mathrm{sim}_{t,i,m}$	Combined similarity logit between detection i and entry m.
$\alpha_{t,i,m}$	Cache posterior over entries for detection i (softmax over m).
$p_{t,i}^{\mathrm{cache}}$	Cache-induced class distribution for detection i.
$p_{t,i}^{\mathrm{glob}}$	Fused global distribution (cache + VLM), before track fusion.
$H(p)$	Shannon entropy of a categorical distribution p.
$w_{\mathrm{init}}, w_{\mathrm{cache}}$	Entropy-based weights for fusing p^{vlm} and p^{cache}.
ε	Small constant for numerical stability in entropy computations.
Global cache hyperparameters / controls	
τ_1	Confidence threshold for considering detections in cache update (high-confidence gate).
τ_2	Posterior/match threshold for assigning detections to existing cache entries.
τ_2^{init}	Initial posterior threshold used in batch-init / early-time init.
$M_{\max}$	Maximum cache size.
$\mathrm{age}_{\max}$	Maximum age allowed for stale entries (lifecycle pruning).
δ	Generic small constant / increment used in lifecycle bookkeeping (as used).
Instance layer	
$\mathcal{T}_t$	Set of active tracks at time t.
$\tau_{t,j}$	Track j at time t (tuple of states).
$X_{t,j}$	Geometric Kalman state for track j (center/scale/aspect + velocities).
$P_{t,j}$	Kalman covariance for track j.
F_{KF}	Kalman state transition matrix (constant-velocity).
Q_{KF}	Kalman process noise covariance.
H_{KF}	Kalman measurement matrix.
R_{KF}	Kalman measurement noise covariance.
$\hat{b}_{t,j}^{-}$	Predicted (prior) box from track j before measurement update.
$\phi_{t-1,j}$	Track appearance feature.
$\bar{f}_{t,i}$	L2-normalized detection feature for association.
$\mathrm{cost}(i,j)$	Association cost between detection i and track j.
$\lambda_{\mathrm{iou}}, \lambda_{\mathrm{feat}}$	Weights for IoU and feature terms in the association cost.
$\mathcal{P}_t$	Set of matched detection-track pairs at time t.
$\mathcal{U}_t$	Unmatched detections at time t.
$\mathcal{V}_t$	Unmatched tracks at time t.
$a_{t,i}$	Assigned track index for detection i (or $\varnothing$ if unmatched).
$\tau_{\mathrm{hi}}, \tau_{\mathrm{lo}}$	High/low score thresholds for ByteTrack-style two-stage association.
TrackSTAD	
$\zeta_{t,j}$	Per-track semantic/SSM state for track j (TrackSTAD).

(continued)

Symbol	Meaning
$p_{t,j}^{\mathrm{trk}}$	Track-level class belief distribution for track j.
$p_{t,i}^{\mathrm{inst}}$	Instance-level fused distribution (track belief + global fused distribution).
$m^{\star}$	Best-matching cache-entry index for a new track (argmax of posterior).
τ_{init}	Threshold for transferring cache state into a newly spawned track.
ρ_c	Per-class prototype direction used by TrackSTAD-vMF, for class c.
π	Per-class mixing weights in TrackSTAD.
STAD-vMF notation	
$\rho_{t,c} \in \mathbb{S}^{D-1}$	vMF mean direction for class c at time t.
$\gamma_{t,c}$	vMF concentration (or equivalent) for class c at time t.
$\pi_{t,c}$	Class mixing weights (Dirichlet-distributed) at time t.
κ^{ems}	E-step concentration parameter used in the responsibility computation.
κ^{trans}	Transition concentration parameter used in the dynamic prior.
$A_D(\cdot), A_D^{-1}(\cdot)$	vMF mean resultant length mapping and its inverse in D dimensions.
$\ell_{t,i,c}^{\mathrm{ssm}}$	SSM-derived logits for detection i, class c at time t.
$p_{t,i}^{\mathrm{ssm}}$	SSM-derived probability distribution for detection i.
λ	Fusion weight between VLM probabilities and SSM probabilities.
$w_{t,i}^{\mathrm{vlm}}, w_{t,i}^{\mathrm{ssm}}$	Confidence weights for VLM and SSM in fusion.
η	Weighting factor for the VLM prior in the E-step observation mixing.
$\mathcal{D}_t$	Sliding-window dataset of recent features/probabilities/weights.
W_{win}	Sliding-window length.
N_{min}	Minimum effective sample size required to update the vMF parameters.
I	EM iteration count.
α_0	Dirichlet prior parameter for π.
STAD-Gaussian notation	
$\mu_{t,c}$	Gaussian mean for class c at time t.
$P_{t,c}$	Gaussian covariance for class c at time t.
R_{G}	Observation covariance (per class) for the Gaussian model.
Q_{G}	Process noise covariance
σ^2	Base observation variance parameter (set via `r_base`).
$\odot, \oslash$	Elementwise multiply / elementwise divide.
I_D	Identity matrix in D dimensions.