ADMISSION CONTROL FOR TRAINING-FREE ZERO-SHOT TEMPORAL ACTION LOCALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot temporal action localization is typically benchmarked using detectors fine-tuned on in-domain segments, but these gains do not transfer across domains. We operate out-of-domain throughout: a frozen image–text encoder is applied directly to test videos, and no weight is ever updated. On that encoder we build a chain of four components. A cross-video memory collects per-class evidence across the test stream, gated by confidence, and blends it into the frame relevance curve. A class-conditioned admission rule, built from a per-class text contrast direction, decides which frames the memory accepts. A video-level reliability term reads the same contrast at video granularity and rescales a video's scores against the rest of the stream. A decode threshold then determines segment boundaries at a quantile of each video's own relevance curve. The resulting method outperforms the state of the art on THUMOS14 and ActivityNet, at both seen/unseen class splits.

1 INTRODUCTION

Temporal action localization predicts the temporal extent and the category of every action instance in an untrimmed video. The supervised formulation is mature, and it assumes a closed vocabulary together with dense boundary annotations. Zero-shot formulations relax the vocabulary, yet most of them still fine-tune on annotated segments drawn from the same domain as the test videos, and the accuracy they report is tied to that domain: retrained out of domain, representative training-based detectors lose almost all of it (Appendix D.15) (Liberatori et al., 2024; Jiang et al., 2026). Out of domain there are, by assumption, no in-domain boundaries to fine-tune on. Training-free localization is therefore the setting itself rather than a concession within it, and this paper stays inside it: no training videos, no boundary annotations, and no gradient computed at any point.

A contrastively pretrained image–text encoder supplies most of what that setting needs. Class names become text prototypes, frames become embeddings in the same space, and the similarity between the two is already a usable relevance curve over time (Liberatori et al., 2024; Han et al., 2026b). However, such an encoder does not maintain any persistent state. Every video is scored from scratch against the same fixed text vectors, so what the encoder reveals about a category in one video is discarded before the next arrives, and the decisions that turn a curve into ranked segments are all taken with a single video in view: which frames represent a class, how far to trust a video against its peers, and where an extent begins. We introduce four components that address this limitation. Each component builds upon the previous one, and together they give the strongest training-free results we know of on both benchmarks.

The first component is a cross-video memory bank. A confidence-gated per-class buffer accumulates over the test stream, is read as a population statistic, and is blended into the frame relevance curve, so a class is scored against evidence pooled from the videos already seen rather than against its text prototype alone.

A memory bank's effectiveness depends on the quality of its contents, so we next focus on regulating admission. Frames chosen by relevance alone predominantly capture the venue, equipment, and crowd that surround an action, and a class-agnostic notion of background cannot separate those from the action itself. The second component instantiates the non-action content per class from a fixed template list and uses the resulting text contrast direction to decide which frames the memory accepts.

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

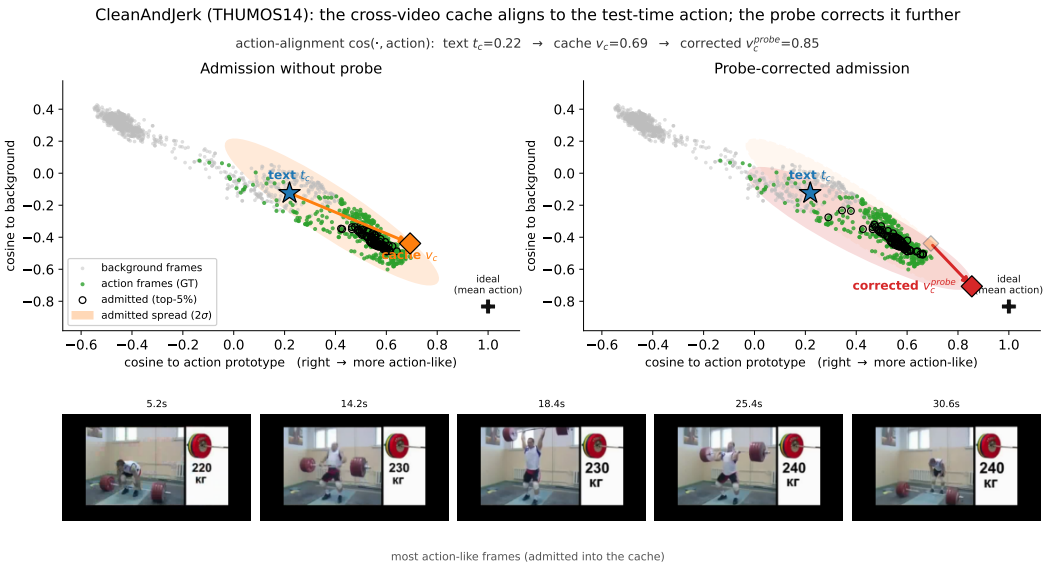


Figure 1: Frame embeddings of one CleanAndJerk test video (THUMOS14), projected onto cosine to the action prototype against cosine to background, with action frames in green and background frames in grey. The text anchor t_c is only weakly action-aligned, the parameter-free cross-video prototype v_c aligns to the test-time action distribution, and the class-conditioned contrast moves v_c^{probe} further toward action and away from background. Action-alignment rises $0.22 \rightarrow 0.69 \rightarrow 0.85$.

That contrast direction carries information at a second granularity. Read over a whole video rather than frame by frame, the dispersion of its response says how peaked that video's evidence is. The third component turns this into a scale on the scores the video emits, so that videos compete on a comparable footing in the pooled ranking. Whether peaked evidence should be promoted or demoted depends on how much of the video the action occupies, and the decode's own output supplies that estimate.

The third component rescales scores across videos; the fourth sets how much of a video lies inside an extent. We keep the frames whose relevance exceeds a quantile of that video's own curve rather than its mean, which fixes the fraction of a video declared active independently of the shape of its similarities.

Our contributions are as follows.

- **A cross-video memory.** A confidence-gated per-class buffer accumulated over the test stream, anchored on a fixed text prior, served from neighbouring classes while it is still empty, and read as a population statistic into the frame relevance curve. It holds no parameters and is never optimized.
- **Class-conditioned admission.** A per-class bank of text phrases describing that class's non-action content, encoded by the same frozen text encoder and reduced to one contrast direction per class, which decides which frames the memory admits. Many of the frames the default relevance rule writes into a class buffer fall outside any annotated instance of that class, and they share venue, equipment and crowd with the action. The templates are fixed across classes, so no per-class phrasing is written by hand and nothing is trained.
- **Video-level reliability.** The same contrast field read at video granularity. Its within-video dispersion, standardized against the videos already seen and clipped, multiplicatively scales the scores that video emits; the sign of that scaling is set by the fraction of the video the decode places inside a segment.
- **A decode threshold.** Extents are cut at a quantile of each video's own smoothed relevance curve instead of at its mean, which decouples the amount of a video declared active from the absolute level and the skew of its similarities.

We evaluate on THUMOS14 and ActivityNet under both standard seen/unseen partitions, against the training-free and test-time-adapted methods that operate on the same frozen features.

2 Related work

Zero-shot TAL with a training set. The seen/unseen protocol used here was established by Eff-Prompt (Ju et al., 2022) and developed by a line of work that varies the localization branch (Zhang et al., 2020; Nag et al., 2022b; Yan et al., 2023; Phan et al., 2024; Li et al., 2024b; Lee et al., 2024). All of it fine-tunes both localizer and classifier on in-domain segments, and the resulting domain bias is decisive: retrained out of domain, the same detectors lose almost all of their in-domain accuracy (Appendix D.15) (Liberatori et al., 2024; Jiang et al., 2026).

Training-free zero-shot TAL. The training-free branch removes the training set instead. T3AL (Liberatori et al., 2024) pseudo-labels each video from a frozen CoCa (Yu et al., 2022) model and refines its relevance curve with a BYOL-style (Grill et al., 2020) test-time objective; TEGU (Liberatori et al., 2026) adapts caption triplets with a max-margin loss, so it uses no training set but is not gradient-free; FreeZAD (Han et al., 2026b) removes optimization entirely, ranking segments with a log-decay-weighted OIC criterion and an FFT actionness recalibration, and its AdaZAD extension reintroduces test-time adaptation. OZ-TAL (Han et al., 2026a) makes the setting online and subtracts a class-agnostic background prompt from the relevance curve. Two limitations recur across this branch: boundaries are read off a single smoothed curve rather than predicted, so precision degrades at high tIoU, and absolute accuracy stays far below supervision (Appendices E, E.2).

Memory, test-time statistics, and negative prompts. Similarly to Karmanov et al. (2024) and a line of cache-based adaptation for vision–language models (Zhang et al., 2024c;a;b; Iwasawa & Matsuo, 2021), we exploit statistics of the test stream itself rather than of a training set. Inside temporal localization, LLT (Jiang et al., 2026) equips a frozen model with a learnable lookup table, and RSKP (Huang et al., 2022) maintains a trained, weakly supervised cross-video snippet memory; its finding that the highest-scoring snippets make poor class representatives is why we gate admission with a text contrast rather than with relevance alone. Negative text prompts have been used elsewhere as well, including per-class negatives *learned* for out-of-distribution detection (Li et al., 2024a).

Positioning. Our method is training-free like T3AL, TEGU and FreeZAD, with no training videos and no boundary annotations, and it differs from that line by accumulating information across the test stream instead of treating each video in isolation (Section 3.2). It differs from LLT by doing so with parameter-free statistics instead of per-video learnable residuals, and from prior negative-prompt work by building class-conditioned probes from a fixed template list without optimization and directing them to memory admission and to a video-level reliability term (Sections 3.3, 3.4) rather than to the relevance curve alone. More details in Appendix E.

3 Method

3.1 Preliminaries

Similarly to other training-free methods in our field (Liberatori et al., 2024; Han et al., 2026b; Jiang et al., 2026), we take the zero-shot setting in which the class vocabulary is partitioned into disjoint seen and unseen subsets and evaluation runs on unseen classes only, we keep both encoders of a contrastively pretrained image–text model frozen throughout (CoCa ViT-L/14; Yu et al., 2022), and we assign each video a single category read from its mean-pooled frame embedding before scoring its frames against that category's text prototype:

$$\bar{\mathbf{x}}_n = \tfrac{1}{T_n}\textstyle\sum_{t'=1}^{T_n} \mathbf{x}_{n,t'}, \qquad \hat{\mathbf{x}}_{n,t} = \tfrac{\mathbf{x}_{n,t}-\bar{\mathbf{x}}_n}{\|\mathbf{x}_{n,t}-\bar{\mathbf{x}}_n\|_2}, \tag{1}$$

$$\mathbf{q}_n = \mathrm{softmax}_c\big(\gamma\,\langle \bar{\mathbf{x}}_n/\|\bar{\mathbf{x}}_n\|_2,\ t_c\rangle\big), \qquad \hat{y}_n = \arg\max_c\ q_{n,c}, \qquad \xi_n = q_{n,\hat{y}_n}, \tag{2}$$

$$\tilde{s}_{n,t} = \gamma\,\langle \hat{\mathbf{x}}_{n,t},\ t_{\hat{y}_n}\rangle, \qquad s_{n,t} = \mathrm{minmax}\big(\mathbf{1}_\kappa * \tilde{s}_{n,\cdot}\big)_t. \tag{3}$$

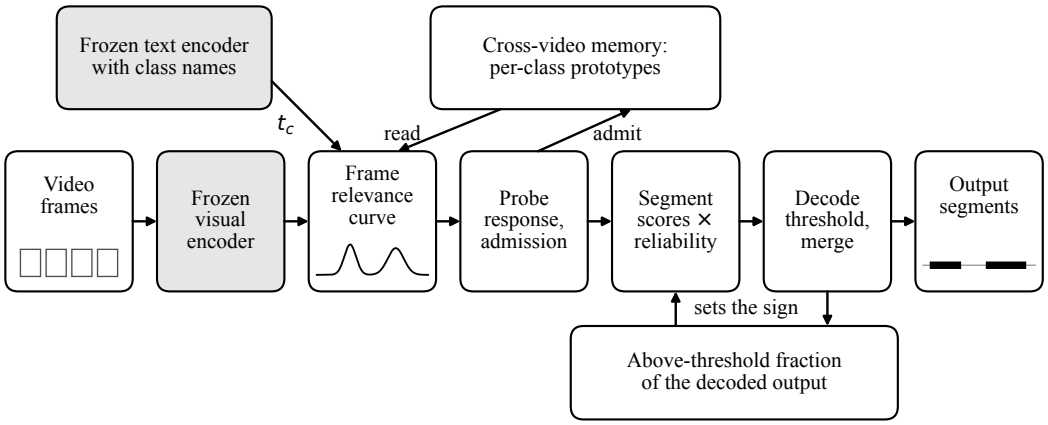


Figure 2: The four components on top of a frozen image–text encoder. The cross-video memory and its class-conditioned admission rewrite the prototype that later videos are scored against; video-level reliability rescales a video's segment scores against the rest of the stream, and the decode threshold fixes how much of that video its extents cover.

Here T_n is the number of frames sampled from video n, t_c the text embedding of class c, γ the encoder's logit scale, $\hat{y}_n$ the video's pseudo-label and ξ_n its confidence, $\mathbf{1}_\kappa$ a boxcar kernel of width κ=20 frames that suppresses high-frequency jitter, and minmax a per-video rescaling to $[0, 1]$. The two curves stay separate because later stages consume them differently: extents are cut from the smoothed $s_{n,\cdot}$, while ranking reads the raw $\tilde{s}_{n,\cdot}$. The subtraction in equation 1 is applied where the video mean is shared scene content, and omitted where it instead carries the class-discriminative first moment; a per-class cosine measured in the embedding space fixes that per benchmark, and it is on for THUMOS14 and off for ActivityNet. The only task-specific input at inference time is the list of candidate class names. More details in Appendix C.3.

Segment scorer. The scorer departs from common practice, so we give it in full. A candidate segment $[b, e]$ of length $\ell = e - b + 1$ is scored by the upper conditional value at risk of the raw relevance curve restricted to it (Rockafellar & Uryasev, 2000), multiplied by a bounded saturating function of its length:

$$\mathrm{score}(b, e) \;=\; m\big(\mathrm{CVaR}_\rho(\tilde{s}_{b:e})\big) \cdot \frac{\ell^{\,g}}{L_0^{\,g} + \ell^{\,g}}, \qquad \mathrm{CVaR}_\rho(\tilde{s}_{b:e}) = \frac{1}{k} \sum_{i \in \mathrm{top}_k(\tilde{s}_{b:e})} \tilde{s}_i, \tag{4}$$

with $k = \lceil \rho\, \ell \rceil$ and $m(x) = 2\,\mathrm{softplus}(x/2)$ a monotone map onto the positive scale the multiplicative stack requires. The inner statistic is the mean of the strongest fraction ρ of the segment's frames, that is, the k-max-mean pooling of weakly supervised temporal action localization (Paul et al., 2018; Narayan et al., 2019; Zhai et al., 2020; Islam et al., 2021) read as a fraction rather than a count: ρ specifies what share of a candidate must be strong, not how strong its single best frame is. Because that statistic is scale-free in ℓ, the duration preference is supplied explicitly by the second factor, a Hill saturation with half-saturation length L_0 and sharpness g, bounded in $(0, 1)$ and interpretable in frames. We set ρ=0.05, L_0=50 frames and g=2, shared across splits and datasets.

3.2 Cross-Video Memory

Similarly to Jiang et al. (2026), we exploit test-time statistics with a cache (Karmanov et al., 2024; Zhang et al., 2024c). Our setup differs in that the cache holds no parameters and is never optimized, and in that it is one object reused twice: a per-class statistic accumulated over the test stream, anchored on a fixed text prior, read once it holds enough samples and served from that anchor while it does not. Two questions remain: which frames a statistic admits, and how it is read. We answer both below.

The anchor. Every statistic is anchored on the class text embedding, the renormalized mean of $M{=}7$ fixed templates such as *"a video of a person [CLS]"*:

$$t_c = \mathrm{Norm}\Big(\tfrac{1}{M}\sum_{m=1}^{M} E_{\text{text}}(p_m(c))\Big). \tag{5}$$

Nothing about t_c adapts. Its role is to stabilize the pseudo-label of equation 2, and the same vector scores the relevance curve.

Statistic I: class appearance. For each class c we accumulate a fixed-size buffer of prototype candidates. Candidate z_n is the mean of test video n's top $\phi_e{=}5\%$ of frames, ranked by $\langle \hat{\mathbf{x}}_{n,t}, t_c\rangle$ against the anchor. A video's target buffer and its confidence ξ_n come from equation 2; once the buffer is full, a new candidate replaces the lowest-confidence entry only if ξ_n exceeds it. The default read is the ℓ_2-renormalized mean of the buffered candidates, and the resulting prototype v_c replaces $t_{\hat{y}_n}$ in equation 3 for subsequent videos:

$$v_c = \mathrm{Norm}\Big(\tfrac{1}{|\mathcal{B}_c|}\sum_{z\in\mathcal{B}_c} z\Big), \tag{6}$$

where $\mathcal{B}_c$ is class c's buffer of candidates. The *population read* summarizes the same evidence at finer grain. Rather than one candidate per admitted video, it keeps the individual top-ϕ_e frames themselves, pooled across every admitted video of the class up to a cap of $K{=}256$, and averages those:

$$v_c^{\text{pop}} = \mathrm{Norm}\Big(\tfrac{1}{|\mathcal{P}_c|}\sum_{\hat{\mathbf{x}}\in\mathcal{P}_c} \hat{\mathbf{x}}\Big), \qquad |\mathcal{P}_c| \le K, \tag{7}$$

with $\mathcal{P}_c$ the retained frame set. A frame is then scored by its logit-scaled dot product to v_c^{pop}, with no covariance term and no maximum over exemplars. Renormalization is not cosmetic in either read: a mean of unit vectors has norm below one, and since $\tilde{s}_{n,t}$ is a γ-scaled logit, an unnormalized prototype would shrink the whole relevance curve and shift the absolute scale that the pooled ranking depends on.

Serving an empty class. A class whose buffer is empty has no statistic to read, so the rule serves it from the anchor, interpolating over the $k{=}3$ nearest already-populated classes $\mathcal{N}_k(c)$ in text-embedding space at sharpness $\tau{=}0.1$:

$$\tilde{v}_c = \mathrm{Norm}\Big(\sum_{j\in\mathcal{N}_k(c)} w_j\, v_j\Big), \qquad w_j = \tfrac{\exp(\langle t_c, t_j\rangle/\tau)}{\sum_{j'\in\mathcal{N}_k(c)}\exp(\langle t_c, t_{j'}\rangle/\tau)}. \tag{8}$$

The anchor therefore enters twice: equation 5 fixes both the pseudo-label that selects a buffer and the neighbour set interpolated over here.

Statistic II: segment duration. The second statistic accumulates the lengths of accepted segments of class c into a Gaussian $\mathcal{N}(\mu_c, \sigma_c^2)$ and reads it by softly demoting candidates of atypical length at strength $\lambda_{\text{dur}}{=}0.3$:

$$\text{score}_{\text{dur}}(b,e) = \text{score}(b,e) - \lambda_{\text{dur}}\big(1 - w_{\text{dur}}(c,\ell)\big)\,\big|\text{score}(b,e)\big|, \quad w_{\text{dur}}(c,\ell) = \exp\Big(-\tfrac{1}{2}\big(\tfrac{\ell-\mu_c}{\sigma_c}\big)^2\Big), \tag{9}$$

with σ_c floored at $\max(0.2\mu_c, 1)$. An under-populated class falls back to the template at its simplest, $w_{\text{dur}}{=}1$ for classes holding fewer than five accepted segments. The prior is applied where per-class lengths are unimodal, which is THUMOS14. More details in Appendix B.5.

Transductivity, and two named configurations. Because the memory accumulates over the test stream, inference is transductive rather than strictly per-video. Every reported number comes from a single pass that includes the cold-start region at the beginning of the stream, and the benefit is a property of the accumulated memory rather than of arrival order: permuting the stream leaves mAP inside its per-split dispersion (Appendix D.10). We report two configurations. BUFFER makes both choices in the simplest way: admission gated by confidence, and the buffer read as a mean, equation 6. ADCON replaces both, with the class-conditioned admission of Section 3.3 and the population read.

3.3 CLASS-CONDITIONED ADMISSION

A memory bank's effectiveness depends on the quality of its contents, so we next focus on regulating admission. The frames that pollute a class buffer are not generic background: they share venue, equipment and crowd with the action, so a single class-agnostic background term subtracted from the scoring curve (Han et al., 2026a) neither keeps them out of the buffer nor tells them apart from the action. We instantiate the non-action content per class instead, and route it to admission.

The probe bank and the admission shift. For class c, a fixed template list is instantiated with the class name to give phrases such as *"the equipment for [CLS], unused"* or *"[CLS], but the ball is not in play"*, which the frozen text encoder maps to ℓ_2-normalized vectors $\eta_1^{(c)}, \dots, \eta_K^{(c)}$. One contrast direction per class is the difference between the mean of the class's K' positive phrases $p_j^{(c)}$ and the mean of its K probes, and the admission curve is shifted along it:

$$\hat{d}_c = \mathrm{Norm}\Big(\tfrac{1}{K'}\textstyle\sum_j p_j^{(c)} - \tfrac{1}{K}\sum_k \eta_k^{(c)}\Big), \qquad \tilde{s}^{\mathrm{adm}}_{n,t} = \tilde{s}_{n,t} + w\,\gamma\left\langle \hat{\mathbf{x}}_{n,t},\, \hat{d}_c \right\rangle. \tag{10}$$

The positive side is, in every reported run, the same $K'{=}7$ prompt-ensemble templates that build t_c in equation 5: the direction needs only a coarse action anchor from which to subtract the non-action bank. The frames written to class c's buffer are its top-ϕ_e frames under $\tilde{s}^{\mathrm{adm}}_{n,\cdot}$, so the contrast changes which frames the buffer admits, while video-level classification runs on the unshifted curve. The shift is flat over $w \in [0.8, 3.0]$, so w is a scale rather than a threshold; we use $w{=}1.2$ everywhere.

Why a difference of embeddings. A negated phrase enters only through a difference of embeddings and never as a similarity the encoder must score on its own, which matters because contrastive image–text encoders do not represent negation (Alhamoud et al., 2025). The templates are fixed across classes and datasets, so no per-class phrasing is written by hand, nothing is trained, and the encoder is the one the pipeline already loads. Controlling over the direction alone shows that the benefit primarily derives from the well-formedness of the textual contrast rather than from class identity: a systematically permuted class-to-direction assignment matches the correct-class direction, while a random matched-norm direction gives nothing. Measured on the admitted sets, the shift rotates the prototype's direction rather than cleaning its contents. More details in Appendix C.3.3.

3.4 VIDEO-LEVEL RELIABILITY

Section 3.3 reads the contrast field of equation 10 frame by frame. The same field read over a whole video answers a second question: how peaked is this video's evidence? A video whose probe response is unusually dispersed separates acting from not-acting more sharply than the stream average, and that is a statement about the video, not about any one of its frames. Writing v for the current test video, we take the within-video standard deviation of the field and standardize it against the videos that precede v in the stream:

$$\varsigma_v = \mathrm{std}_t\big(\langle \hat{\mathbf{x}}_{v,t},\, \hat{d}_{\hat{y}_v}\rangle\big), \qquad z_v = \mathrm{clip}\left(\frac{\varsigma_v - \mu_v^{\mathrm{str}}}{\sigma_v^{\mathrm{str}}},\ -3,\ +3\right), \tag{11}$$

where μ_v^{str} and σ_v^{str} are running moments of ς over the stream prefix, updated after v is scored so that no video influences its own value, and held inert for the first 20 videos. Standardizing is what makes the quantity comparable across videos, which is what a pooled cross-video ranking needs; clipping at three standard deviations keeps one outlier from dominating that ranking.

The useful direction of the scaling depends on how much of the video the action occupies. Where little of the video is active, a peaked field marks a real action and corroborates the curve; where most of it is active, a peaked field says that only a small part of a long declared extent carries the action. The decode of Section 3.5 already reports that fraction, so the estimate is available without any extra machinery: with $\mathcal{A}_v$ the set of frames it keeps,

$$f_v = \frac{|\mathcal{A}_v|}{T_v}, \qquad o_v = \mathrm{sgn}\big(\tfrac{1}{2} - f_v\big), \tag{12}$$

with the comparison strict and ties resolved to $+1$, so that o_v is never zero and the correction is never silently switched off. The reliability term then multiplies every segment score of that video:

$$s_i' = s_i \exp\big(\alpha\, o_v z_v\big). \tag{13}$$

The exponential keeps scores positive and makes the correction multiplicative, matching the multiplicative scorer of equation 4. One global strength α=0.1 is shared by both datasets and all splits, so the component is one mechanism rather than a per-corpus branch. More details in Appendix D.2.

3.5 DECODE THRESHOLD

Section 3.4 rescales scores across videos; the threshold that turns the smoothed curve into extents sets how much of each video is inside one. We keep the frames that exceed a quantile of that video's own curve rather than its mean:

$$\mathcal{A}_v = \{\, t \;:\; s_{v,t} > Q_q(s_{v,\cdot}) \,\}, \tag{14}$$

where Q_q is the q-th quantile of the video's smoothed relevance curve, and contiguous runs in $\mathcal{A}_v$ separated by at most δ_{merge}=20 frames are merged into the candidate segments that equation 4 scores. A quantile pins the fraction of a video declared active, so it is insensitive to the shape of the curve, whereas the mean moves with skew: one strong peak raises it and shrinks the extents around every other peak in the same video. We set q=0.60. More details in Appendix B.

The chain. The four components run sequentially. The memory turns per-video scoring into a population statistic over the test stream; admission decides what that population contains; the same admission field, read at video granularity, scales the scores a video emits against the rest of the stream; and the decode threshold sets how much of each video those scores are attached to. No weight is updated anywhere in the chain, and inference stays a single forward pass per video.

4 RESULTS

4.1 EXPERIMENTAL SETUP

Datasets and protocol. We evaluate on THUMOS14 and ActivityNet v1.3 under the zero-shot protocol of Liberatori et al. (2024). The vocabulary is partitioned into disjoint seen and unseen subsets, and evaluation is on videos containing unseen categories. We report both standard partitions, $50\%/50\%$ and $75\%/25\%$. Every value below is a mean over the same ten random seen/unseen resamplings, so comparisons between our own configurations are paired. The metric is mAP at temporal IoU $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ on THUMOS14 and $\{0.5, 0.75, 0.95\}$ on ActivityNet v1.3, together with the average over those thresholds.

Configuration. ADCON adds no learned parameters. One setting of every hyperparameter is used for all ten splits and both datasets (Appendix D.2); the decode threshold cuts at the 0.60 quantile of each video's smoothed curve. Inference cost, dispersion over stream orders and the significance tests are in Appendices D.9, D.10 and D.12.

Baselines. We compare against the methods that operate at the same frozen shared-encoder feature level under the same ten splits: T3AL (Liberatori et al., 2024), LLT (Jiang et al., 2026), Ti-FAD (Lee et al., 2024), FreeZAD and its test-time-adapted extension AdaZAD (Han et al., 2026b). Closed-vocabulary and fully supervised reference rows are in Appendix D.15, and per-method provenance is in Appendix D.13.

4.2 THUMOS14

Table 1 reports THUMOS14. ADCON reaches 15.4 average mAP at $50\%/50\%$ and 15.0 at $75\%/25\%$. It leads the strongest fixed-weights baseline, LLT, by 3.6 and 2.9, and the strongest test-time-adapted baseline, AdaZAD, by 1.3 and 1.0. The lead holds at every reported threshold, from 30.2 against 25.2 at tIoU 0.3 to 4.0 against 3.6 at tIoU 0.7. The margin over every baseline is significant under a video-level clustered bootstrap over the ten class splits (Appendix D.12).

4.3 ACTIVITYNET V1.3

Table 2 reports ActivityNet v1.3 under the same protocol and the same ten splits. ADCON reaches 18.8 average mAP at $75\%/25\%$ and 17.3 at $50\%/50\%$. It leads the strongest fixed-weights baseline,

Table 1: Zero-shot temporal action localization on THUMOS14. mAP (%) at temporal IoU thresholds for the $50\%/50\%$ protocol, with the average for both protocols; means over ten class splits. TF = training-free, TTA = test-time adaptation. Baseline rows are published values under this protocol, except FreeZAD, which is our reproduction on the same ten splits and reaches the published THUMOS14 level (Appendix D.14). Avg is the five-threshold mean computed from the unrounded per-threshold values; our $50\%/50\%$ row averages to 15.363 and our $75\%/25\%$ row to 14.981, the full configuration of Table 15.

			50% seen / 50% unseen						75%/25%
Method	TF	TTA	0.3	0.4	0.5	0.6	0.7	Avg	Avg
T3AL (fixed weights)	✓	✗	11.4	6.8	3.5	1.7	0.6	4.8	4.6
FreeZAD	✓	✗	20.7	13.4	8.3	4.7	2.5	9.9	10.0
LLT (fixed weights)	✓	✗	24.0	16.3	9.9	5.9	3.1	11.8	12.1
AdCon (ours)	✓	✗	**30.2**	**21.3**	**13.5**	**7.7**	**4.0**	**15.4**	**15.0**
T3AL	✗	✓	20.7	14.3	8.9	5.3	2.7	10.4	9.2
Ti-FAD	✗	✓	–	–	–	–	–	10.5	–
LLT	✗	✓	25.2	17.1	10.9	6.4	3.6	12.6	12.8
AdaZAD	✗	✓	–	–	–	–	–	14.1	14.0

Table 2: Zero-shot temporal action localization on ActivityNet v1.3. mAP (%) at temporal IoU thresholds for the $75\%/25\%$ protocol, with the average for both protocols; means over ten class splits. TTA = test-time adaptation. FreeZAD and AdaZAD are our reproductions on the same ten splits. Both protocols are tabulated in Appendix D.16.

		75% seen / 25% unseen				50%/50%
Method	TTA	0.5	0.75	0.95	Avg	Avg
T3AL (fixed weights) (Liberatori et al., 2024)	✗	26.1	13.9	2.9	14.3	13.3
LLT (fixed weights) (Jiang et al., 2026)	✗	29.3	16.7	4.2	16.7	15.4
FreeZAD (Han et al., 2026b)	✗	29.3	16.8	2.8	16.3	14.9
AdCon (ours)	✗	**31.7**	**20.3**	**4.4**	**18.8**	**17.3**
T3AL	✓	28.1	14.9	3.3	15.4	14.3
LLT	✓	29.4	17.3	4.7	17.1	15.7
AdaZAD (Han et al., 2026b)	✓	29.1	16.3	2.8	16.1	14.6

Table 3: Each component priced against the otherwise identical configuration without it, means over ten class splits. Every entry is a leave-one-out difference from the full configuration with the two new components held off, the configuration named MEMORY+ADMISSION in Table 15: 14.832 and 14.674 on THUMOS14 and 17.756 and 16.519 on ActivityNet, in the column order shown. Build-up deltas against a weaker configuration are larger or smaller than these and are reported separately in Appendix C, each against its own named reference.

	THUMOS14		ActivityNet v1.3	
Component	50/50	75/25	75/25	50/50
Cross-video memory	+1.759	+1.953	+1.816	+1.793
Class-conditioned admission	+1.606	+1.402	+0.060	+0.059
Video-level reliability	+0.059	+0.107	+0.927	+0.758
Decode threshold	+0.459	+0.114	+0.271	+0.222

Table 4: Additivity of the video-level reliability and the decode threshold. Change in average mAP, means over ten class splits. Both rows build up from MEMORY+ADMISSION (Table 15), the configuration carrying neither component: 14.832 and 14.674 on THUMOS14, 17.756 and 16.519 on ActivityNet.

	THUMOS14		ActivityNet v1.3	
	50/50	75/25	75/25	50/50
Added jointly	+0.531	+0.307	+1.063	+0.823
Sum of the separate additions	+0.518	+0.221	+1.198	+0.980

LLT, by 2.1 and 1.9, and the strongest test-time-adapted baseline, again LLT, by 1.7 and 1.6. The lead sits in the middle of the threshold grid, 20.3 against LLT's 17.3 at tIoU 0.75, while at the strictest threshold 0.95 the test-time-adapted LLT row is ahead, 4.7 against 4.4. The FreeZAD and AdaZAD rows come from running the authors' released code on these splits (Appendix D.14).

4.4 COMPONENT ANALYSIS

Table 3 prices the four components of ADCON, each against the otherwise identical configuration without it and all four against the same reference (Table 15). Every entry is positive, on both corpora and at both protocols. Figure 1 shows the geometry the first two of them act on, the cross-video prototype and the class-conditioned admission that rotates it toward the test video's action frames.

Table 4 turns to the two components introduced here, the video-level reliability and the decode threshold. Adding them together moves average mAP by close to the sum of their separate contributions in all four settings, so the two combine without redundancy.

5 CONCLUSION

We built training-free zero-shot temporal action localization out of four components on a frozen image–text encoder, each following from the one before it. A cross-video memory collects per-class evidence across the test stream, gated by confidence, anchored on a fixed text prior, and read as a population statistic into the frame relevance curve. A class-conditioned admission rule, one text contrast direction per class built from a fixed template list, decides which frames that memory accepts. A video-level reliability term reads the same contrast field at video granularity and multiplicatively scales the scores a video emits against the rest of the stream, with the sign of the scaling taken from the fraction of the video the decode declares active. A decode threshold cuts extents at a quantile of each video's own smoothed curve instead of at its mean. The chain improves over the state of the art on both benchmarks at both class splits.

No weight is updated anywhere, and every statistic the method keeps is read off the test stream it is already processing. Any localizer that emits a relevance curve and a set of ranked extents can carry the same four components.

AI USE STATEMENT

We used generative AI tools for editing text for readability. We have not used generative AI tools for any of the tasks for which disclosure is required: no part of the theoretical framing, the mathematical claims, the hypotheses, the methodology or experiment design, the implementation, the data cleaning or the interpretation of results was produced with such tools, and no synthetic data was generated. We have reviewed all AI-assisted work: every edited passage was checked against the measurements it describes by the authors, and no claim, number or table entry originates from a generative model. We take responsibility for the final content of this work, including text, claims and artifacts.

REPRODUCIBILITY STATEMENT

We use THUMOS14 and ActivityNet v1.3 in the reference versions and with the pre-processing released with the official implementation of Liberatori et al. (2024)[1]. ADCON adds no learned parameters, and one setting of every hyperparameter is used for all ten class splits and both datasets. The pipeline, the prompt and probe banks, the hyperparameter values and their sensitivity, the baseline provenance and the significance procedure are specified in Appendices B, C.3, D.2, D.13 and D.12. Our code will be released publicly upon acceptance.

ETHICS STATEMENT

We have reviewed the ICLR code of ethics and adhered to it. Our experiments use only the public THUMOS14 and ActivityNet v1.3 benchmarks under their released licenses, these contain no personally identifying annotations, and we collect no new data. Temporal action localization supports surveillance and behavioral-monitoring applications, and a zero-shot method that generalizes to unseen actions widens that surface; our contribution is an offline analysis method rather than a deployed system. ADCON performs no gradient computation and runs in a single forward pass on one commodity GPU, so its energy footprint is small relative to the detectors it is compared against (Appendix D.9).

REFERENCES

Josiah Aklilu, Xiaohan Wang, and Serena Yeung-Levy. Zero-shot action localization via the confidence of large vision-language models. *arXiv preprint arXiv:2410.14340*, 2024. CVPR 2025 VidLLMs Workshop.

Kumail Alhamoud, Shaden Alshammari, Yonglong Tian, Guohao Li, Philip Torr, Yoon Kim, and Marzyeh Ghassemi. Vision-language models do not understand negation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Piyush Bagad, Makarand Tapaswi, and Cees G. M. Snoek. Test of time: Instilling video-language models with a sense of time. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. arXiv:2301.02074.

Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-NMS – improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 5561–5569, 2017.

Massimo Bosetti, Shibingfeng Zhang, Benedetta Liberatori, Giacomo Zara, Elisa Ricci, and Paolo Rota. Text-enhanced zero-shot action recognition: A training-free approach. In *International Conference on Pattern Recognition (ICPR)*, 2024. arXiv:2408.16412.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 961–970, 2015.

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6299–6308, 2017. doi: 10.1109/CVPR.2017.502.

[1] https://github.com/benedettaliberatori/T3AL

Feng Cheng and Gedas Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 503–521, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19830-4.

Jia-Run Du, Kun-Yu Lin, Jingke Meng, and Wei-Shi Zheng. Towards completeness: A generalizable action proposal generator for zero-shot temporal action localization. In Apostolos Antonacopoulos, Subhasis Chaudhuri, Rama Chellappa, Cheng-Lin Liu, Saumik Bhattacharya, and Umapada Pal (eds.), *Pattern Recognition*, pp. 252–267, Cham, 2025. Springer Nature Switzerland. ISBN 978-3-031-78444-6.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6202–6211, 2019. doi: 10.1109/ICCV.2019.00630.

Chun-Mei Feng, Kai Yu, Yong Liu, Salman Khan, and Wangmeng Zuo. Diverse data augmentation with diffusions for effective test-time prompt tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2704–2714, 2023.

Sourabh Vasant Gothe, Vibhav Agarwal, Sourav Ghosh, Jayesh Rajkumar Vachhani, Pranay Kashyap, and Barath Raj Kandur Raja. What's in the flow? exploiting temporal motion cues for unsupervised generic event boundary detection. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6926–6935, 2024. doi: 10.1109/WACV57701.2024.00679.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21271–21284, 2020.

Akshita Gupta, Aditya Arora, Sanath Narayan, Salman Khan, Fahad Shahbaz Khan, and Graham W. Taylor. Open-vocabulary temporal action localization using multimodal guidance. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2024. arXiv:2406.15556.

Chaolei Han, Hongsong Wang, Xin Gong, and Jie Gui. Oz-tal: Online zero-shot temporal action localization. *arXiv preprint arXiv:2605.09976*, 2026a.

Chaolei Han, Hongsong Wang, Jidong Kuang, Lei Zhang, and Jie Gui. Training-free zero-shot temporal action detection with vision-language models. *IEEE Transactions on Multimedia*, pp. 1–12, 2026b. doi: 10.1109/TMM.2026.3701531.

De-An Huang, Shijia Liao, Subhashree Radhakrishnan, Hongxu Yin, Pavlo Molchanov, Zhiding Yu, and Jan Kautz. LITA: Language instructed temporal-localization assistant. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 202–218, 2024.

Linjiang Huang, Liang Wang, and Hongsheng Li. Weakly supervised temporal action localization via representative snippet knowledge propagation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2203.02925.

Jeongseok Hyun, Su Ho Han, Hyolim Kang, Joon-Young Lee, and Seon Joo Kim. Exploring scalability of self-training for open-vocabulary temporal action localization. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 9388–9397, 2025. doi: 10.1109/WACV61041.2025.00911.

Haroon Idrees, Amir R. Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2016.10.018. URL https://www.sciencedirect.com/science/article/pii/S1077314216301710.

Ashraful Islam, Chengjiang Long, and Richard Radke. A hybrid attention mechanism for weakly-supervised temporal action localization. In *AAAI Conference on Artificial Intelligence*, 2021.

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

Han Jiang, Haoyu Tang, Xiaoxuan Mu, Chen Li, and Jihua Zhu. Memory matters: Boosting training-free zero-shot temporal action localization with a learnable lookup table. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9762–9772, 2026.

Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 105–124, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19833-5.

Chen Ju, Zeqian Li, Peisen Zhao, Ya Zhang, Xiaopeng Zhang, Qi Tian, Yanfeng Wang, and Weidi Xie. Multimodal prompting for low-shot temporal action localization. *arXiv preprint arXiv:2303.11732*, 2023.

Hyolim Kang, Jinwoo Kim, Taehyun Kim, and Seon Joo Kim. UBoCo: Unsupervised boundary contrastive learning for generic event boundary detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20073–20082, 2022. doi: 10.1109/CVPR52688.2022.01944.

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

Kanchan Keisham, Thenukan Pathmanathan, and Thangarajah Akilan. ConTrans: Learning text-enhanced local-global temporal representations for zero-shot temporal action localization. *arXiv preprint arXiv:2605.30689*, 2026. Canadian AI 2026.

Yearang Lee, Ho-Joong Kim, and Seong-Whan Lee. Text-infused attention and foreground-aware modeling for zero-shot temporal action detection. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 9864–9884. Curran Associates, Inc., 2024. doi: 10.52202/079017-0316. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/13250eb13871b3c2c0a0667b54bad165-Paper-Conference.pdf.

Tianqi Li, Guansong Pang, Xiao Bai, Wenjun Miao, and Jin Zheng. Learning transferable negative prompts for out-of-distribution detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17584–17594, 2024a.

Zeqian Li, Shangzhe Di, Zhonghua Zhai, Weilin Huang, Yanfeng Wang, and Weidi Xie. Universal video temporal grounding with generative multi-modal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 71787–71816, 2025.

Zhiheng Li, Yujie Zhong, Ran Song, Tianjiao Li, Lin Ma, and Wei Zhang. Detal: Open-vocabulary temporal action localization with decoupled networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):7728–7741, 2024b. doi: 10.1109/TPAMI.2024.3395778.

Benedetta Liberatori, Alessandro Conti, Paolo Rota, Yiming Wang, and Elisa Ricci. Test-time zero-shot temporal action localization. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18720–18729, 2024. doi: 10.1109/CVPR52733.2024.01771.

Benedetta Liberatori, Alessandro Conti, Lorenzo Vaquero, Paolo Rota, Yiming Wang, and Elisa Ricci. Zero-shot temporal action localization through textual guidance. In *2026 IEEE 20th International Conference on Automatic Face and Gesture Recognition (FG)*, pp. 1–9. IEEE, 2026. doi: 10.1109/fg67764.2026.11557050.

Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 3–21, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01225-0.

Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3889–3898, 2019. doi: 10.1109/ICCV.2019.00399.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980–2988, 2017.

Wei Lin, Muhammad Jehanzeb Mirza, Mateusz Kozinski, Horst Possegger, Hilde Kuehne, and Horst Bischof. Video test-time adaptation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22952–22961, 2023.

Shuming Liu, Chen Zhao, Fatimah Zohra, Mattia Soldan, Alejandro Pardo, Mengmeng Xu, Lama Alssum, Merey Ramazanova, Juan León Alcázar, Anthony Cioppa, Silvio Giancola, Carlos Hinojosa, and Bernard Ghanem. Opentad: a unified framework and comprehensive study of temporal action detection. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 2616–2626, 2025. doi: 10.1109/CVPRW67362.2025.00247.

Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022. doi: 10.1109/TIP.2022.3195321.

Junyi Ma, Erhang Zhang, Yin-Dong Zheng, Yuchen Xie, Yixuan Zhou, and Hesheng Wang. EgoLoc: A generalizable solution for temporal interaction localization in egocentric videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3717912. arXiv:2508.12349.

Sachit Menon and Carl Vondrick. Visual classification via description from large language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Liliane Momeni, Mathilde Caron, Arsha Nagrani, Andrew Zisserman, and Cordelia Schmid. Verbs in action: Improving verb understanding in video-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Sauradip Nag, Mengmeng Xu, Xiatian Zhu, Juan-Manuel Perez-Rua, Bernard Ghanem, Yi-Zhe Song, and Tao Xiang. Multi-modal few-shot temporal action detection via vision-language meta-adaptation. *arXiv preprint arXiv:2211.14905*, 2022a.

Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 681–697, Cham, 2022b. Springer Nature Switzerland. ISBN 978-3-031-20062-5.

Sayak Nag, Orpaz Goldstein, and Amit K. Roy-Chowdhury. Semantics guided contrastive learning of transformers for zero-shot temporal activity detection. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6243–6253, 2023. doi: 10.1109/WACV56688.2023.00618.

Sanath Narayan, Hisham Cholakkal, Fahad Shahbaz Khan, and Ling Shao. 3C-Net: Category count and center loss for weakly-supervised action localization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Sujoy Paul, Sourya Roy, and Amit K. Roy-Chowdhury. W-TALC: Weakly-supervised temporal activity localization and classification. In *European Conference on Computer Vision (ECCV)*, 2018.

Thinh Phan, Khoa Vo, Duy Le, Gianfranco Doretto, Donald Adjeroh, and Ngan Le. ZEETAD: Adapting Pretrained vision-language model for zero-shot end-to-end temporal action detection. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7046–7055, 2024. doi: 10.1109/WACV57701.2024.00689.

Shraman Pramanick, Effrosyni Mavroudi, Yale Song, Rama Chellappa, Lorenzo Torresani, and Triantafyllos Afouras. Enrich and detect: Video temporal grounding with multimodal LLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24297–24308, 2025.

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 Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/radford21a.html`.

Asif Raza, Bang Yang, and Yuexian Zou. Zero-shot temporal action detection by learning multimodal prompts and text-enhanced actionness. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(11): 11000–11012, 2024. doi: 10.1109/TCSVT.2024.3414275.

Shuhuai Ren, Linli Yao, Shicheng Li, Xu Sun, and Lu Hou. TimeChat: A time-sensitive multimodal large language model for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14313–14323, 2024.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3): 21–41, 2000.

Dingfeng Shi, Yujie Zhong, Qiong Cao, Jing Zhang, Lin Ma, Jia Li, and Dacheng Tao. React: Temporal action detection with relational queries. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 105–121, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-20080-9.

Dingfeng Shi, Yujie Zhong, Qiong Cao, Lin Ma, Jia Li, and Dacheng Tao. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18857–18866, June 2023.

Mike Zheng Shou, Stan Weixian Lei, Weiyao Wang, Deepti Ghadiyaram, and Matt Feiszli. Generic event boundary detection: A benchmark for event segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8075–8084, 2021. doi: 10.1109/ICCV48922.2021.00797.

Zheng Shou, Hang Gao, Lei Zhang, Kazuyuki Miyazawa, and Shih-Fu Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, pp. 162–179, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01270-0.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 14274–14289, 2022.

Asim Unmesh, Ramesh Kaki, Rahul Jain, Mayank Patel, and Karthik Ramani. Exploring vision-language models for open-vocabulary zero-shot action segmentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026. Program paper ThI2I.215; arXiv:2602.21406.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. Open-vocabulary action localization with iterative visual prompting. *IEEE Access*, 13:56908–56917, 2025.

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

Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos: Versatile action models for video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 142–160, 2024a.

Xiaohan Wang, Yuhui Zhang, Orr Zohar, and Serena Yeung-Levy. Videoagent: Long-form video understanding with large language model as agent. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 58–76, 2024b.

Mengmeng Xu, Chen Zhao, David S. Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10156–10165, 2020. doi: 10.1109/CVPR42600.2020.01017.

Shen Yan, Xuehan Xiong, Arsha Nagrani, Anurag Arnab, Zhonghao Wang, Weina Ge, David Ross, and Cordelia Schmid. Unloc: A unified framework for video localization tasks. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13623–13633, 2023. doi: 10.1109/ICCV51070.2023.01253.

Zaiquan Yang, Yuhao Liu, Gerhard Hancke, and Rynson W.H. Lau. Unleashing the potential of multimodal llms for zero-shot spatio-temporal video grounding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. arXiv:2509.15178.

Zuhao Yang, Yingchen Yu, Yunqing Zhao, Shijian Lu, and Song Bai. TimeExpert: An expert-guided video LLM for video temporal grounding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24286–24296, 2025b.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022.

Yuanhao Zhai, Le Wang, Wei Tang, Qilin Zhang, Junsong Yuan, and Gang Hua. Two-stream consensus network for weakly-supervised temporal action localization. In *European Conference on Computer Vision (ECCV)*, 2020.

Ce Zhang, Simon Stepputtis, Katia Sycara, and Yaqi Xie. Dual prototype evolving for test-time generalization of vision-language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. Code: github.com/zhangce01/DPE-CLIP.

Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 492–510, Cham, 2022a. Springer Nature Switzerland. ISBN 978-3-031-19772-7.

Erhang Zhang, Junyi Ma, Yin-Dong Zheng, Yixuan Zhou, and Hesheng Wang. Zero-shot temporal interaction localization for egocentric videos. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 19554–19561, 2025. arXiv:2506.03662.

Lingling Zhang, Xiaojun Chang, Jun Liu, Minnan Luo, Sen Wang, Zongyuan Ge, and Alexander Hauptmann. Zstad: Zero-shot temporal activity detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 879–888, 2020. doi: 10.1109/CVPR42600.2020.00096.

Lingling Zhang, Xiaojun Chang, Jun Liu, Minnan Luo, Zhihui Li, Lina Yao, and Alex Hauptmann. Tn-zstad: Transferable network for zero-shot temporal activity detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3848–3861, 2023. doi: 10.1109/TPAMI.2022.3183586.

Marvin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 38629–38642, 2022b.

Taolin Zhang, Jinpeng Wang, Hang Guo, Tao Dai, Bin Chen, and Shu-tao Xia. BoostAdapter: Improving vision-language test-time adaptation via regional bootstrapping. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 67795–67825, 2024b.

Yabin Zhang, Wenjie Zhu, Hui Tang, Zhiyuan Ma, Kaiyang Zhou, and Lei Zhang. Dual memory networks: A versatile adaptation approach for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28718–28728, 2024c.

Minghang Zheng, Xinhao Cai, Qingchao Chen, Yuxin Peng, and Yang Liu. Training-free video temporal grounding using large-scale pre-trained models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–37, 2024.

Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 12993–13000, 2020.

APPENDIX ROADMAP

Appendix A fixes the protocol: datasets, class splits, metric, frozen encoders, every hyperparameter, and the full inference pass. Appendix B records the base-pipeline detail the method inherits or supplies as a drop-in. Appendix C fixes the named reference configurations and the delta convention every number in it is read against, then analyses the four components one at a time, in method order: cross-video memory, class-conditioned admission, video-level reliability, decode threshold. Appendix D carries the remaining measurements: additivity of the two new components, hyperparameter sensitivity, backbone generality, the localization-bottleneck decomposition, inference cost, dispersion and significance, baseline provenance, the ActivityNet reproduction note, closed-vocabulary reference rows, and the full ActivityNet table. Appendix E gives the extended related work.

A EXPERIMENTAL PROTOCOL

A.1 DATASETS, CLASS SPLITS, AND EVALUATION DEPTH

The two benchmarks differ on exactly the axes the components are configured against. THUMOS14 carries eight instances per video at about one percent of the clip each, so a within-video cut has many short positive runs to separate and a per-class length distribution with usable variance. ActivityNet carries one instance covering a third of the clip, so the same cut produces a single near-covering run, the length distribution is broad and close to the clip length, and the strict $\{0.5, 0.75, 0.95\}$ grid prices the placement of that one run rather than the ordering of many. Ten resamplings per setting make every comparison among our own configurations paired at the split level, which is the unit the significance tests of Appendix D.12 resample at. *The protocol is a single fixed grid: ten class resamplings per setting, unseen-only evaluation, and one metric convention per benchmark.*

Table 5: The two benchmarks, their class splits, and the evaluation grid. Both follow the class-split protocol of Liberatori et al. (2024): the vocabulary is partitioned into disjoint seen and unseen subsets, evaluation is on videos containing unseen categories only, and every reported figure is a mean over ten independent seen/unseen resamplings of that vocabulary. Instance statistics are per evaluation video.

	THUMOS14	ActivityNet v1.3
Annotated categories	20	200
Unseen categories, $50\%/50\%$	10	100
Unseen categories, $75\%/25\%$	5	50
Resamplings per setting	10	10
Evaluation videos, $50\%/50\%$	203	train $\cup$ val of unseen classes
Median action instances per video	8	1
Mean action instances per video	–	1.15
Median instance length, fraction of clip	$\approx 1\%$	33%
90th-percentile instance length	–	95%
tIoU thresholds	$\{0.3, 0.4, 0.5, 0.6, 0.7\}$	$\{0.5, 0.75, 0.95\}$
Reported metric	per threshold and average	per threshold and average

Table 6: Frozen encoders and the one feature-level treatment that differs between the two benchmarks. Both encoders are frozen at every stage and no gradient is computed at training or at test time; a video costs one forward pass.

	THUMOS14	ActivityNet v1.3
Image–text model	CoCa ViT-L/14	CoCa ViT-L/14
Image encoder	frozen	frozen
Text encoder	frozen	frozen
Gradient computation, train or test	none	none
Forward passes per video	1	1
Video-mean subtraction, equation 1	on	off
Frame similarities computed on	$\hat{\mathbf{x}}_{n,t}$	ℓ_2-normalized raw embeddings
Second encoder checked	CLIP ViT-L/14 (Appendix D.3)	–

A.2 Frozen encoders and per-dataset feature treatment

The subtraction of equation 1 removes the temporal mean of a video's frame embeddings and renormalizes, which on THUMOS14 removes shared scene content and on ActivityNet removes the class-discriminative first moment; the measurement that fixes the switch is the per-class cosine between the action and the non-action content of the same class, -0.451 against $+0.946$ (Appendix C.3.5). Where the subtraction is applied it shifts frame–text cosines toward zero, so any stage reading an absolute similarity is calibrated against the shifted range on THUMOS14 and the raw cosine range on ActivityNet; every stage reading a within-video rank is unaffected either way. The same pipeline run on a second frozen encoder keeps its margin (Appendix D.3). *The encoders are frozen and identical across benchmarks, and the single feature-level difference is the video-mean subtraction, set by a measured property of each corpus.*

A.3 Every hyperparameter and its value

Three groups of values sit in the table for different reasons. The relevance-curve, scorer and memory knobs are round-number defaults carried over from the reproduced baseline and from prior work (Liberatori et al., 2024; Han et al., 2026b); the contrast weight, the template bank and the read-out were all fixed on THUMOS14; and the reliability strength, its sign threshold and the decode quantile are the settings the sweeps of Appendices C.4 and C.5 bracket, each sitting inside a plateau rather than on a peak found per split. Only three rows differ by benchmark, and each is set on a measured property of that benchmark rather than on its mAP: the subtraction on the action/non-action cosine, the duration statistic on the unimodality of the per-class length distribution, and the ActivityNet gap bridge on the 1.15 instances per video the corpus carries. *One value per knob covers all twenty runs of each benchmark, and the three per-dataset rows are each justified by a property of the corpus.*

Table 7: Every hyperparameter of the deployed pipeline. One value is shared by all ten splits of both settings and, except where the last column says otherwise, by both benchmarks; nothing is re-tuned per split, and the protocol supplies no validation set on which it could be. Robustness of the shared values is swept in Appendix D.2.

Stage	Symbol or switch	Value	Per-dataset
Relevance curve	smoothing width κ	20 frames	✗
	merge gap δ_{merge}	20 frames	✗
	logit scale γ	the encoder's own	✗
	video-mean subtraction, equation 1	on / off	✓
Text anchor	prompt templates M	7	✗
	anchor read	renormalized template mean	✗
Segment scorer	CVaR fraction ρ	0.05	✗
	Hill half-saturation L_0	50 frames	✗
	Hill sharpness g	2	✗
Cross-video memory	admission top-fraction ϕ_e	0.05	✗
	prototype buffer size	4 candidates per class	✗
	population cap	256 snippets per class	✗
	cold-start neighbours k	3	✗
	cold-start sharpness τ	0.1	✗
Class-conditioned admission	contrast weight w	1.2	✗
	positive bank K'	7 (the prompt ensemble)	✗
	non-action bank K	19 phrases, 5 families	✗
	read-out	difference direction, equation 10	✗
Duration statistic	strength λ_{dur}	0.3	✗
	σ_c floor	$\max(0.2\mu_c, 1)$	✗
	refit period / minimum observations	every 5 / at least 5 segments	✗
	enabled	THUMOS14 only	✓
Video-level reliability	covariate	dispersion of the probe response	✗
	strength α	0.10	✗
	z clip	3.0	✗
	warm-up	20 videos	✗
	sign threshold θ	0.50	✗
Decode	cut rule	quantile of the video's own smoothed curve	✗
	decode quantile q	0.60	✗
	gap bridge / instance cap	1600 frames / one per video	✓

A.4 THE FULL INFERENCE PASS

Table 8: The streaming state of the inference pass: what each stage reads and what it writes. Every read uses the state as it stood before the current video arrived, and the corresponding write happens after the video's own output has been emitted, so no video influences its own score.

Stage	Reads	Writes
Video-level pseudo-labeling	text anchors t_c	nothing
Frame relevance	prototype buffer of the pseudo-label class	nothing
Admission	contrast direction $\hat{d}_c$	prototype buffer, population
Decode threshold and merge	the video's own smoothed curve	nothing
Segment scoring	duration Gaussian $\mathcal{N}(\mu_c, \sigma_c^2)$	accepted lengths
Video-level reliability	Welford mean/sd of the covariate	the same moments

Figure 2 draws the stages and Table 8 gives the state each one touches; the block below is the complete deployed procedure in the order the implementation executes it, reading every constant from Table 7. The decode threshold and merge produce the extents, the segment scorer supplies their scores, and the reliability factor is applied last so that what it scales is the final ranked value.

```
Input : frozen image encoder E_img, frozen text encoder E_txt,
        class names C, test-video stream v_1 .. v_N
State : per-class prototype buffer B_c, duration Gaussian D_c,
        running mean/sd of the reliability covariate (Welford)

Once, before the stream, for every class c in C:
```

```
  t_c   <- Norm( mean_m E_txt( template_m(c) ) )
  eta_c <- E_txt( non-action templates instantiated with c )
  d_c   <- Norm( mean_j pos_c[j]  -  mean_k eta_c[k] )

for n = 1 .. N:
  X       <- E_img( frames(v_n) )            # one forward pass
  x       <- (X - mean_t X)/||.||  if subtraction on, else X/||.||
  y, xi   <- argmax_c , max_c softmax( g <mean_t X, t_c> )
  p       <- read(B_y)           if B_y populated
             interpolate(t, B)   else if any class populated
             t_y                 otherwise
  s_raw   <- g <x, p>                        # ranking curve
  s_sm    <- minmax( boxcar_kappa * s_raw )  # extraction curve
  s_adm   <- s_raw + w * g <x, d_y>          # admission curve

  # ==== cross-video memory: admit ============================
  z_n     <- mean of top-phi_e frames of x ranked by s_adm
  write z_n into B_y, replacing the lowest-confidence entry
          if xi exceeds it

  # ==== decode threshold and merge ===========================
  cut     <- quantile_q( s_sm )
  R       <- merge( runs of s_sm above cut, gap <= delta_merge )
  R       <- bridge(R, anet_gap), keep one run  if ActivityNet

  # ==== segment scores =======================================
  for [b,e] in R with length L:
      sc  <- m( CVaR_rho( s_raw[b:e] ) ) * L^g/(L0^g + L^g)
      sc  <- sc - lambda_dur (1 - w_dur(y, L)) |sc|  if enabled

  # ==== video-level reliability ==============================
  cov_n   <- fraction of v_n's frames inside some interval of R
  sign_n  <- -1 if cov_n > theta else +1
  zz      <- clip( welford_z( dispersion of <x, d_y> ), +-clip )
  mult_n  <- 1.0 if fewer than warmup videos seen,
             else exp( alpha * sign_n * zz )

  emit { (b, e, y, sc[b,e] * mult_n) : [b,e] in R }
  fold v_n into the Welford moments, its lengths into D_y

Output: all emitted segments, ranked per class by their score
        across videos.
```

Every state update is one-sided in the stream. A video reads the prototype buffer, the duration Gaussian and the Welford moments as they stood before it arrived and is folded in only afterwards, so no video influences its own score; the reliability multiplier is exactly 1.0 until the warm-up prefix has passed, which is what keeps a small-sample standard deviation from turning the transform into noise amplification. The multiplier is a positive scalar applied per video, so it preserves the within-video ordering of scores exactly while changing the pooled cross-video interleaving that per-class AP reads. All reported numbers come from this single pass with the cold-start prefix included; there is no warm-up set and no second pass. *The deployed method is one forward pass per video plus streaming per-class state, with every read strictly preceding the corresponding write.*

B Base pipeline detail

B.1 Notation, and the two relevance curves

Two curves are carried separately because later stages consume them differently. Extraction thresholds the smoothed $s_{n,\cdot}$ against a within-video statistic and therefore depends only on the within-video ordering of relevance; ranking evaluates equation 4 on the raw $\tilde{s}_{n,\cdot}$ and pools the result across all videos of a class, so it depends on the absolute scale. The consequence is used repeatedly below: an intervention that changes only the within-video ordering moves extraction, one that changes only

Table 9: Symbols used throughout the appendix. Relevance curves always carry a video/frame subscript, so $s_{n,t}$ is a curve value and s_i a segment start time; a candidate interval is written $[b, e]$ in local frame indices and an emitted segment (s_i, e_i, c_i, p_i) in video time.

Symbol	Meaning
$\hat{\mathbf{x}}_{n,t}$	frame embedding, after equation 1 where it applies
t_c	class text anchor, the renormalized M-template mean, equation 5
$\hat{y}_n,\ \xi_n$	video-level pseudo-label and its confidence, equation 2
$\tilde{s}_{n,t}$	raw relevance curve; read by ranking
$s_{n,t}$	smoothed, min-max rescaled curve; read by extraction
$\tilde{s}^{\mathrm{adm}}_{n,t}$	admission curve, equation 10
$v_c,\ \tilde{v}_c$	class prototype and its cold-start proxy, equation 8
$\hat{d}_c$	class contrast direction, equation 10
$[b, e],\ \ell$	candidate interval in local frame indices, and its length
(s_i, e_i, c_i, p_i)	an emitted segment: start, end, category, confidence
q	decode quantile of the smoothed curve
θ	threshold on the decode statistic that sets the reliability sign
$\alpha,\ z_n$	reliability strength, and video n's standardised covariate

the per-video scale moves the pooled ranking, and the video-level reliability factor of Appendix C.4 is built to be the second kind exactly. *The raw and smoothed curves are distinct objects, and which one a stage reads determines whether it can move extraction or ranking.*

B.2 What is inherited, what is a drop-in, and what is claimed

Table 10: Provenance of every element of the pipeline. *Inherited* elements are reproduced unchanged from the baseline we compare against; the *drop-in* scorer replaces one inherited element without being part of the claim; the four *claimed* components are the subject of Appendix C.

Element	Provenance	Defined in
Frozen-encoder relevance curve and equation 1	inherited (Han et al., 2026b)	Section 3.1
Video-level pseudo-labeling from text prototypes	inherited (Liberatori et al., 2024; Han et al., 2026b)	equation 2
Segment extraction by a within-video cut	inherited (Han et al., 2026b)	Section 3.1
Log-decay context weighting ω_j	inherited; reproduced baseline only	equation 15
Prompt-ensemble anchor	inherited (Radford et al., 2021)	equation 5
Upper-CVaR scorer with a bounded duration factor	drop-in, ours	equation 4
Per-class duration statistic	drop-in, ours	equation 9
Cross-video memory	claimed; storage design after Jiang et al. (2026)	Section 3.2
Class-conditioned admission	claimed	Section 3.3
Video-level reliability	claimed	Appendix C.4
Decode threshold	claimed	Appendix C.5

The one inherited element that needs restating is the context weighting, because the mean-subtraction of equation 1 changes the scale it operates on. For a candidate spanning frames $[b, e]$ of length $L = e - b + 1$, the context window on each side has length $l = \lfloor L/4 \rfloor$, so the penalty region scales with the candidate rather than being a fixed number of frames, and the j-th context frame outward from either boundary receives weight

$$\omega_j = \frac{1/\log(j+1)}{\sum_{j'=1}^{l} 1/\log(j'+1)}, \qquad j = 1, \ldots, l, \qquad \sum_{j=1}^{l} \omega_j = 1, \tag{15}$$

monotonically decreasing in j and normalized to sum to one, so the two outer sums of the inherited criterion are weighted *means* of context relevance rather than sums: the penalty does not grow with l, and hence not with L, which is what keeps that criterion comparable across candidates of different length. Frames falling outside the video are excluded and equation 15 is renormalized over the surviving indices, so a candidate touching the first or last frame is not penalised for missing context. Our deployed pipeline scores segments with equation 4 instead, which supplies its own length dynamics, and the two scorers are interchangeable at equal accuracy (Appendix C.2.1). *Four*

elements are claimed, two are drop-in replacements measured against the inherited ones, and the rest is reproduced unchanged.

B.3 SEGMENT EXTRACTION AND THE ACTIVITYNET RE-ASSEMBLY

Table 11: The ActivityNet decoder is the within-video cut plus two operations, each priced by leave-one-out at $75\%/25\%$ over ten splits, the depth and setting the ActivityNet headline numbers use, except the instance-cap row, which is priced at $50\%/50\%$ as its cell says and on the reproduced FreeZAD configuration of Appendix D.14 at 14.9. The memory row is the leave-one-out from MEMORY+ADMISSION at 17.756 (Table 15) that Table 3 reports as $+1.816$. The gap bridge is compared against a strided gap and the memory against switching it off, everything else held fixed.

Operation	Replaced by	Change in average mAP
Gap bridge below 1600 frames (`anet_gap`)	a strided gap	-4.053
One instance per video (`anet_count_cap`)	fixed m=0.7 cut, no cap	-1.3 (at $50/50$)
Cross-video memory	memory off	-1.816

Candidate segments are the maximal runs of frames above the decode cut; runs separated by a gap of at most δ_{merge}=20 frames are merged and single-frame runs are discarded. A within-video cut is what makes the procedure robust to the wide variation in action density across videos, and it also makes the per-video minmax rescaling of equation 3 inert wherever the cut is a within-video statistic, since minmax is an increasing affine map and the predicate is invariant under any such map. Rescaling matters only where the cut is an absolute constant. On ActivityNet, whose instances are long and cover most of the clip, the same cut is followed by the two operations in the table: dips are bridged up to a large gap and the single dominant run is retained, which reassembles one clean covering segment per video. The gap bridge is the larger of the two by a wide margin, consistent with the benchmark's 1.15 ground-truth instances per video, since a proposer emitting several disjoint candidates fragments the one instance that is there. *Every ActivityNet number we report comes from this decoder, and its gap bridge is set by the corpus's instance count rather than by a sweep on mAP.*

B.4 THE ANCHOR, SERVING EMPTY CLASSES, AND THE PROMPT ENSEMBLE

Table 12: Video-level pseudo-label accuracy per prompt template and for the seven-template ensemble (THUMOS14 $50\%/50\%$, mean $\pm$ std over the ten resamplings). The ensemble is computed once at initialization and used throughout, both for pseudo-labeling and as the anchor a class with an empty buffer is served from.

Prompt template	Pseudo-label accuracy
"a video of a person {c}"	0.862 ± 0.036
"a person performing {c}"	0.852 ± 0.050
"a clip showing {c}"	0.841 ± 0.038
"a video of action {c}"	0.836 ± 0.040
"{c} in a video"	0.832 ± 0.046
"someone is {c}"	0.829 ± 0.049
"{c}"	0.824 ± 0.062
single-template mean	0.839
ensemble (7 templates)	$\mathbf{0.869 \pm 0.035}$

Individual templates span 0.824 to 0.862 and disagree on about 10% of videos; the ensemble reaches 0.869, above the best single template and 3.0 points over the single-template average. Averaging templates therefore regularizes the anchor: it attains near-top accuracy without committing to any one phrasing. Because the same vector also scores the per-frame relevance curve, its effect on localization is proportionally larger than the accuracy gain alone, 12.5 to 13.1 average mAP (Table 16). The anchor enters the pipeline at three points: it fixes the pseudo-label of equation 2 that selects which buffer a video writes to, it is the vector a class with an empty buffer is served from, and it

supplies the neighbour set $\mathcal{N}_k(c)$ that equation 8 interpolates over from the k=3 populated classes of highest text similarity, with the proxy ℓ_2-renormalized before use and a fallback to t_c when no class is populated. *One fixed seven-template ensemble serves as classifier, prototype fallback and neighbour metric, and it is the strongest of the eight text read-outs measured.*

B.5 THE DURATION STATISTIC: SCHEDULE, FALLBACK, AND WHERE IT IS ENABLED

Table 13: The per-class duration statistic. *Upper block*: its schedule and its fallback for under-populated classes. *Lower block*: the benchmark properties the unimodal Gaussian assumes, and the measured effect on each benchmark. The THUMOS14 pair is a build-up over rows C+I and C+I+D of Table 16, 12.5 and 12.8, rather than a leave-one-out from the full configuration. The ActivityNet pair is measured at $50\%/50\%$ over ten splits on the arm where it was priced there (an earlier `ens_cache` arm that predates the CVaR scorer and sits 1.8 average mAP below the memory-plus-admission configuration, which measures 16.5 at this setting and depth).

	THUMOS14	ActivityNet v1.3
Strength λ_{dur}	0.3	0.3
Refit period / minimum observations	every 5 / at least 5	every 5 / at least 5
Fallback below the minimum	w_{dur}=1, inert	w_{dur}=1, inert
Median instances per video	8	1
Median instance length, fraction of clip	≈1%	33%
Average mAP with the statistic off	12.5	**14.75**
Average mAP with the statistic on	**12.8**	13.82
Configuration the pair is measured on	C+I	`ens_cache`
Enabled	✓	✗

Action categories have characteristic duration distributions, and the Gaussian $\mathcal{N}(\mu_c, \sigma_c^2)$ of equation 9 accumulates them from the lengths of accepted segments, refitted every five segments once five have been observed for class c, with σ_c floored at $\max(0.2\mu_c, 1)$ to avoid a hard cutoff. With λ_{dur}=0.3 the penalty removes at most three tenths of a candidate's score magnitude, so a duration-plausible segment is untouched while one of atypical length loses rank. Two properties of ActivityNet decide the last row. The Gaussian is unimodal and presumes a tight characteristic length, whereas ActivityNet categories range from seconds to minutes, so the fitted density is nearly flat and its self-estimated median an uninformative summary; and because the decoder emits essentially one near-clip-length run per video, there is no length-based ordering within a video to perform, so equation 9 only rescales that lone segment's score and thereby perturbs the cross-video ranking under the $\{0.5, 0.75, 0.95\}$ grid. The sign in the table follows. *The duration statistic is enabled on the benchmark whose per-class length distribution satisfies the unimodality it fits, and disabled on the one that violates it.*

B.6 WHY THE DURATION FACTOR MUST SATURATE

Both ladders are monotone and point the same way. An unbounded power degrades further the faster it grows, and within the bounded family a later half-saturation degrades further the later it arrives. Both orderings follow from the length distribution of THUMOS14 instances, which are short enough that a duration preference must already be close to saturated by the time it reaches a typical action length; a factor that is still rising there is a length preference competing with the contrastive term rather than a tie-break among candidates of comparable strength. This is also why the inner statistic is a fixed *fraction* of the candidate rather than a fixed number of frames: CVaR_ρ averages the top $\lceil \rho\ell \rceil$ values and is therefore scale-free in ℓ, which leaves the second factor as the only place a duration preference is expressed and makes it inspectable in frames. *The length preference has to saturate by a typical action length, and L_0=50 frames with g=2 is the setting at which it does.*

Table 14: The bounded Hill factor of equation 4 against unbounded and slower-saturating alternatives, paired against the inherited criterion on THUMOS14. *Upper block*: replacing the Hill factor by a power ℓ^{α}. *Lower block*: moving the half-saturation length L_0 at the deployed sharpness. Deployed values in bold. The reference for every row is the deployed length factor, the $L_0{=}50$ row that reads $+0.00$ by construction.

Length factor	Setting	Paired change in average mAP
power ℓ^{α}	$\alpha{=}0$	-0.37
	$\alpha{=}1$	-1.82
	$\alpha{=}1.5$	-2.06
	$\alpha{=}2$	-2.25
	$\alpha{=}2.5$	-2.36
Hill, half-saturation L_0	$\mathbf{L_0{=}50}$ **frames**	$\mathbf{+0.00}$
	$L_0{=}100$ frames	-0.67
	$L_0{=}200$ frames	-1.54

Table 15: The named configurations every delta in this section is measured against, THUMOS14, average mAP over ten class splits. BUFFER gates admission on the video-level confidence alone and reads its four-slot buffer as a mean; MEMORY+ADMISSION gates admission on the class-conditioned contrast and reads the pooled population; the full configuration is MEMORY+ADMISSION plus the video-level reliability factor and the decode threshold, and is the configuration Table 1 reports at one decimal as 15.4 and 15.0.

Name	Admission and memory read	50%/50%	75%/25%
Reproduced baseline (FreeZAD)	text anchor only, no memory	9.9	10.0
BUFFER	confidence-gated, buffer read as a mean	13.515	13.4
MEMORY+ADMISSION	contrast-gated, population read	14.832	14.674
Full configuration (ADCON)	the same, plus reliability and decode	15.363	14.981

C PER-COMPONENT ANALYSIS

C.1 REFERENCE CONFIGURATIONS AND THE DELTA CONVENTION

Two delta conventions appear in this section and they answer different questions, so one is fixed as the default and the other is always announced. *Every delta in this section is a leave-one-out difference from the full configuration of Table 15 unless the table's own caption names a different reference, and any table that builds a configuration up from a weaker one names that weaker configuration in its caption.* Because the video-level reliability factor and the decode threshold act on stages the other two components do not touch and compose additively with them (Appendix D.1), the leave-one-out reference for the cross-video memory and the class-conditioned admission is the full configuration with those two held off, namely MEMORY+ADMISSION; each of their captions says so and quotes its value. A build-up delta and a leave-one-out delta for the same component differ in general, and here they differ systematically: a component removed from a configuration that contains its partner is worth more than the same component added to a configuration that does not, because the two halves of the memory mechanism are complementary (Appendix C.2.5). Both numbers are therefore reported for the memory and for the admission, each against its own named reference. On ActivityNet v1.3 the two references are MEMORY+ADMISSION, 17.756 at 75%/25% and 16.519 at 50%/50%, and the full configuration, 18.819 and 17.342. *Leave-one-out from the full configuration is the default, a build-up names its starting configuration, and the four reference configurations are fixed in Table 15.*

C.2 CROSS-VIDEO MEMORY

The memory is the single largest term in the table and it is separable from everything else in it, and Figure 3 plots the build-up the rows below trace. Holding the other four switches off, turning the memory on moves average mAP from 9.9 to 12.4, and that $+2.5$ reappears at the same magnitude

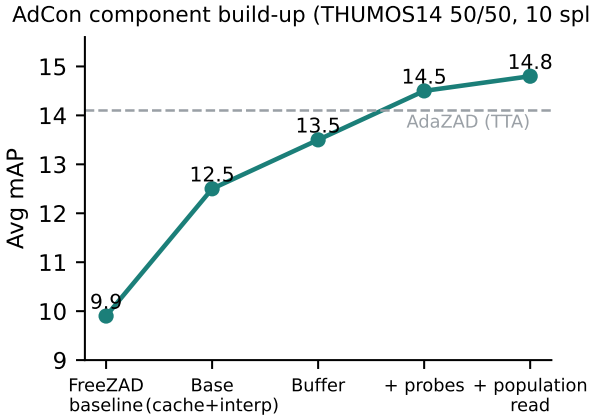


Figure 3: Component build-up on THUMOS14 ($50\%/50\%$, mean over ten splits), every point a row of Table 16: the reproduced FreeZAD baseline (9.9, row 1); "Base (cache + interp)", the cross-video memory with the cold-start interpolation (12.5, row C+ I), which is the memory alone at 12.4 plus 0.1 from the interpolation branch; BUFFER (13.5, row C+D+P); the class-conditioned admission; and the population read, reaching MEMORY+ADMISSION at 14.8 before the two components of Appendices C.4 and C.5 are added, already above the test-time-adapted AdaZAD (14.1, dashed). The axis is a build-up, so each step is measured from the point to its left and not by leave-one-out.

in every block: no combination of the duration statistic, the prompt ensemble, the cold-start fallback or the write-gate temperature substitutes for it, and none of them changes its size by more than 0.4. The cold-start fallback moves average mAP by at most 0.2 wherever it appears, which is the behaviour a fallback branch should have once the stream has populated the buffers, and the write-gate temperature moves it by at most 0.1 as well and is not carried in the final method; both are folded into the reference configuration for that reason. What the memory changes is which vector the relevance curve is computed against, and Appendix C.2.4 shows it never changes which class a video is assigned.

The memory therefore carries two deltas against two named references, and the pair behaves the way the convention of Appendix C.1 anticipates. As a build-up from the reproduced FreeZAD baseline of Table 15 it is worth $+2.5$, 9.9 to 12.4, which is the quantity this factorial measures and the second step of Figure 3. By leave-one-out from MEMORY+ADMISSION it is worth $+1.759$, 14.832 down to 13.073 when the memory is switched off and the relevance curve falls back to the text anchor, and $+1.953$ at $75\%/25\%$, 14.674 down to 12.721 under the same removal, which are the quantities Table 3 reports. The leave-one-out figure is the smaller of the two here because the configuration it is removed from already contains the class-conditioned admission and the prompt ensemble, both of which recover part of what the memory supplies, whereas the build-up starts from a configuration that contains neither. *Accumulating a per-class prototype across the test stream is worth $+2.5$ average mAP as a build-up from the reproduced baseline and $+1.759$ by leave-one-out from the memory-plus-admission configuration, and composes additively with every other base-pipeline switch.*

C.2.1 SEGMENT SCORER AND MEMORY ARE SEPARABLE

The two axes act independently. The CVaR scorer is equal-accuracy to the inherited criterion, a -0.2 difference with the memory off and none with it on, and the memory contributes $+2.4$ under the inherited scorer and $+2.6$ under ours, landing on the same 13.5 either way. This is the measurement that lets us present the scorer as a drop-in rather than as part of the claim: it supplies the length dynamics equation 15 otherwise supplies, in an interpretable form, without being the source of any reported gain. At $75\%/25\%$ the same substitution is worth -0.08. *The scorer and the memory occupy orthogonal slots, so the memory's contribution is the same under either scorer.*

Table 16: Factorial over the base-pipeline switches on THUMOS14 ($50\%/50\%$, mean over ten splits). C = cross-video memory (`cache_mode=positive`), I = cold-start interpolation, T = confidence-calibration temperature (a rescale of the per-class *write* gate confidence, not of the ranking score; it is not part of the final method), D = per-class duration statistic, P = seven-template prompt ensemble. I and T are no-ops whenever C is disabled, so the 2^5 factorial collapses to the 20 distinct configurations below and the 12 cache-disabled cells that are identical to a row already listed are omitted. Avg is the five-threshold mean of tIoU $\{0.3, 0.4, 0.5, 0.6, 0.7\}$ used throughout on THUMOS14, three of which are shown. This table is a build-up rather than a leave-one-out: every Δ is measured against row 1, the reproduced FreeZAD baseline of Table 15 at 9.9 average mAP, with all five switches off. †: previously-reported COMBO_KITCHEN_SINK.

C	I	T	D	P	0.3	0.5	0.7	Avg	Δ
					20.7	8.3	2.5	9.9	–
			✓		21.2	8.5	2.5	10.1	+0.2
				✓	22.2	9.1	2.7	10.8	+0.9
			✓	✓	22.8	9.4	2.7	11.1	+1.2
✓					24.7	10.7	3.3	12.4	+2.5
✓				✓	26.2	11.5	3.4	13.1	+3.2
✓			✓		25.1	11.0	3.4	12.6	+2.7
✓			✓	✓	26.8	11.8	3.6	13.5	+3.6
✓		✓			24.6	10.6	3.3	12.3	+2.4
✓		✓		✓	26.1	11.5	3.4	13.1	+3.2
✓		✓	✓		25.0	10.9	3.4	12.6	+2.7
✓		✓	✓	✓	26.7	11.8	3.6	13.5	+3.6 †
✓	✓				25.0	10.8	3.3	12.5	+2.6
✓	✓			✓	26.3	11.5	3.5	13.1	+3.2
✓	✓		✓		25.4	11.1	3.5	12.8	+2.9
✓	✓		✓	✓	26.9	11.8	3.6	13.5	+3.6
✓	✓	✓			24.9	10.7	3.3	12.4	+2.5
✓	✓	✓		✓	26.2	11.5	3.4	13.1	+3.2
✓	✓	✓	✓		25.3	11.0	3.4	12.7	+2.8
✓	✓	✓	✓	✓	26.8	11.8	3.6	13.5	+3.6 †

Table 17: Segment scorer crossed with the cross-video memory (THUMOS14 $50\%/50\%$, mean over ten splits, base-pipeline switches held on). This is a build-up across the memory axis, not a leave-one-out: the memory-off column is the cache-disabled configuration with the duration statistic and the prompt ensemble on (row D+P of Table 16, 11.1 under the inherited scorer) and the memory-on column is BUFFER of Table 15, with its buffer-mean read, under each scorer.

Scorer	memory off	memory on (BUFFER)
Log-decay OIC (inherited)	11.1	13.5
Upper CVaR (ours, drop-in)	10.9	13.5

C.2.2 WHICH FRAMES THE MEMORY SHOULD STORE

Two things follow. Sourcing the prototype from boundary or low-similarity frames is strongly destructive, -3.5 and -5.3 average mAP, so the memory is best served by clean, high-similarity exemplars; and reselecting among these categories by the base relevance score does not improve on the deployed rule, which matches the best category to within 0.01 average mAP. On the deployed configurations that selectivity comes from the within-video top-fraction ranking rather than from the video-level write gate, and replacing the gate's input with the pre-subtraction features leaves the numbers unchanged at both settings. What this study varies is *which* frames enter the prototype along the score-ranked axis, with the prototype's *direction* essentially fixed, so it prices frame selection; direction is the degree of freedom the class-conditioned admission of Appendix C.3 moves, which is why a clean-content prototype and the contrast direction are two degrees of freedom rather

Table 18: Prototype-content attribution on THUMOS14 ($50\%/50\%$, ten splits), measured on BUFFER. Each row rebuilds the class prototype from a single frame category and re-scores with everything else held fixed. The categories are cut from ground-truth segments and are a measurement device, not a pipeline input: the deployed rule is the confidence top-fraction row, which is BUFFER of Table 15 at 13.515 and is the reference every other row is read against.

Prototype source	Average mAP
Top-scoring action frames	13.52
Confidence top-fraction (deployed, BUFFER)	**13.515**
All action frames	13.29
Interior action frames	12.77
Mid-similarity action frames	11.64
Action-side boundary frames	10.01
Low-similarity action frames	8.23

than two estimates of one. *The memory's content should be clean high-similarity exemplars, and the within-video score-ranked axis that selects them is already at its best setting.*

C.2.3 NON-PARAMETRIC AND INSTANCE-LEVEL READ-OUTS

Table 19: Training-free memory read-outs as drop-in replacements for the mean prototype (THUMOS14 $50\%/50\%$, mean over ten splits, on BUFFER). Average mAP is the localization metric; cls-acc and N_k-skew characterise the memory used as a standalone video recognizer, lower N_k-skew meaning less hubness. Non-parametric read-outs use buffer capacity 64. The first row is BUFFER of Table 15 at 13.515, the reference the other eight are read against.

Read-out	Average mAP	memory cls-acc	N_k-skew
Mean prototype (deployed, BUFFER)	**13.515**	0.49	1.29
Max (nearest exemplar)	12.76	**0.57**	0.93
k-NN	13.22	0.56	0.99
Medoid	13.37	0.45	1.25
Intermodal-KL affinity	13.22	0.56	**0.77**
MTA-inlier selection	13.50	0.49	1.20
TDA negative cache	11.04	0.49	1.29
InMaP (α=0.25)	12.92	0.49	1.34
Hubness penalty (w=1)	13.04	0.49	1.39

A mean collapses a class's exemplars into a single vector, so the first question is whether the limited within-video headroom above is an artifact of averaging. It is not: replacing the mean with fixed non-parametric read-outs over a larger retained exemplar buffer, keeping everything else fixed, leaves localization mAP at or below the mean prototype in every row, the closest of them, an inlier-selection rule, matching it to within 0.02 and the best of the non-parametric block, the medoid, landing 0.15 below. The second column shows the two axes are decoupled. A nearest-exemplar read-out raises the memory's standalone recognition accuracy from 0.49 to 0.57 and cuts its N_k-skewness from 1.29 to 0.93, and an intermodal-KL affinity cuts it to 0.77, yet neither converts into mAP, because the memory scores the frame-relevance curve while the video label is assigned by the text ensemble. The only row that moves localization is the one that folds text logits back in. *A mean prototype for localization and the text ensemble for recognition is the best of the nine read-outs measured, so the memory stays a simple top-fraction mean rather than a heavier exemplar store.*

C.2.4 THE MEMORY IS A POOR VIDEO RECOGNIZER, SO CLASSIFICATION STAYS ON TEXT

Used for frame-level relevance after classification the memory is effective; used as the classifier it is not, and the errors are dominated by hubness. A single class is predicted for 93 of 203 evaluation videos although it is the true label for 19 of them, giving a k-occurrence skewness of 2.03 and

Table 20: The memory used *as* the video-level recognizer, against the deployed text ensemble (THUMOS14, ten splits). The last row grows the memory with ground-truth-selected action and background exemplars and is a ceiling probe, not a configuration the method can run.

Video-level recognizer	accuracy	N_k-skew	per-frame action/background AUC
Text prototype ensemble (deployed)	$\mathbf{0.869}$	–	0.77
Nearest memory prototype	0.49	2.03	–
Memory grown with ground-truth-clean exemplars	0.845	–	0.77

a predicted-label entropy well below uniform. The ceiling probe sharpens the division of labour: growing the memory with ground-truth-clean exemplars raises its recognition to 0.845, still below the text ensemble's 0.869, while its per-frame action/background separation saturates at the same ROC-AUC the deployed relevance curve already reaches, and adding a background prototype does not move it. The memory is therefore a source of relevance rather than of labels or boundaries, which is why equation 2 stays on the text prototypes and cross-video state enters strictly afterwards. *Video-level classification is kept on text because the memory's own recognition is hubness-dominated, and the memory never changes which class a video is assigned.*

C.2.5 The population read of the memory, and its composition with admission

Table 21: The two memory slots, admission and read, priced both ways on the same ten splits. *Upper block*: added to BUFFER at $75\%/25\%$ (13.4), alone and in the other's presence. *Lower block*: removed from the memory-plus-admission configuration (14.8 at $50\%/50\%$, 14.7 at $75\%/25\%$).

Slot	alone, over BUFFER	with the other present
Class-conditioned admission	$+0.90$	$+1.40$
Population read	-0.09	$+0.42$
	Δ at $50\%/50\%$	Δ at $75\%/25\%$
Removing class-conditioned admission	-1.61	-1.40
Removing the population read	-0.52	-0.42

The population read retains the individual top-ϕ_e snippets un-pooled across all admitted videos up to a per-class cap of 256 and scores a frame by one logit-scaled dot product against the ℓ_2-normalized mean of that population: one mean vector, no covariance term and no maximum over exemplars, which is why it does not coincide with the max read-out of Appendix C.2.3. The table shows the two slots are one mechanism with two halves. Each is worth more in the other's presence than alone, and removal from that configuration prices both higher than their build-up deltas over BUFFER, so we quote leave-one-out deltas throughout and carry both conventions here. The reason is the division of labour between the slots: admission changes *what* the population contains and the read changes *how* it is summarised, so a read over an uncurated population and a curated population read by a single mean are each missing half of the mechanism. *Admission and read are complementary halves of one memory mechanism, and each is worth more with the other present at both settings.*

C.3 Class-conditioned admission

The three rows built from the shared template list gain between $+0.79$ and $+0.98$ over BUFFER, the wrong-class and permuted directions at or slightly above the class's own, while the matched-norm random direction gains nothing. Any instantiation of the shared template list points along the same acting-versus-not-acting axis, and that axis is what the mechanism uses; a matched-norm vector that is not an instantiation of the list points nowhere useful. Two design consequences follow directly. The templates are shared across classes rather than authored per class, because it is the well-formedness of the contrast and not class identity that pays; and the qualifier for the bank is *class-instantiated*, since the class slot makes the phrases concrete while the shared list is what makes them work. The same reading explains why a single global background prompt subtracted from the scoring curve is worth less than this construction routed to admission, $+0.4$ against $+0.79$ over BUFFER: the frames that pollute a class buffer share venue, equipment and crowd with the action,

Table 22: What the contrast direction has to be (THUMOS14 50%/50%, ten resamplings). The reference for this table is not the full configuration but BUFFER of Table 15 with its buffer-mean read, 13.515 average mAP, and Δ is measured against that value. Each row adds the same shift at the same admission step with everything else held fixed, replacing the class's own contrast direction in equation 10 by a different unit vector of the same norm and re-running the stream.

Direction used to shift the admission curve	Average mAP	Δ over BUFFER
BUFFER, no shift (reference)	13.515	–
A different class's contrast direction	14.498	+0.983
A fixed permutation of the class-to-direction assignment	14.448	+0.933
The class's own contrast direction (deployed)	14.309	+0.794
A random matched-norm direction	13.258	−0.257

so the direction that separates them has to be instantiated with the class. *What the component needs is a well-formed textual acting-versus-not-acting axis, not the correct class's copy of it.*

C.3.1 WHERE THE CONTRAST IS APPLIED

Table 23: Scope ablation: the contrast routed to one pipeline stage at a time, on the same ten splits per setting, THUMOS14. All rows share the reference of the first row, which is MEMORY+ADMISSION of Table 15 with the admission mechanism removed and the population read retained, 13.226 and 13.272. That row is a different configuration from BUFFER, which admits on confidence and reads its buffer as a mean and measures 13.515 at 50%/50%. The deployed configuration applies the shift at both stages and reaches the 14.832 and 14.674 of Table 15, which is the +1.606 and +1.402 leave-one-out delta of Table 3.

Contrast applied to	50%/50%	75%/25%
Nothing: mechanism off, population read (reference)	13.226	13.272
The relevance curve only	13.276	13.309
Buffer admission only	14.759	14.614
Both stages (deployed)	**14.832**	**14.674**

The ordering is the whole result: routing the same direction to buffer admission moves average mAP by more than an order of magnitude further than routing it to the scoring curve, at both settings. The component is therefore an intervention on *what the memory stores*, and it reaches localization only through the prototypes that later videos read against, which is exactly the path Figure 2 draws from admission into the memory and back into the curve. This is also why the component cannot be described as a curve correction: applied to the curve the same vector with the same weight moves the metric by a small fraction of what it moves through admission, and the two routings differ in nothing else. *The contrast pays through buffer admission, not through the scoring curve.*

C.3.2 ADMISSION CROSSED WITH THE MEMORY READ ON ACTIVITYNET

Table 24: The mechanism crossed with the memory read on ActivityNet 75%/25%, ten splits. The *deployed read* row is the configuration every ActivityNet number we report uses, so its with-the-mechanism cell is MEMORY+ADMISSION at 17.756 and the without-it cell at 17.696 is the leave-one-out reference behind the +0.060 of Table 3; the *larger read* row widens the retained population.

Memory read	with the mechanism	without it
Deployed read	**17.756**	17.696
Larger read	17.455	17.567

The best of the four cells is the deployed read with the mechanism on, which is the configuration the ActivityNet results use, and the two factors are not interchangeable: widening the read moves the

metric down in both columns, so the mechanism cannot be traded for a larger population and a larger population cannot substitute for the mechanism. Read together with Appendix C.3.1 this locates the component at the same place on both corpora, the admission stage, and it is the reason the contrast is enabled on ActivityNet at the same weight $w{=}1.2$ as on THUMOS14 rather than being switched per dataset. How much non-action content a textual contrast can remove on each corpus is a measured property of the representation, given in Appendix C.3.5. *The deployed read with the mechanism on is the strongest of the four cells, and the read size is not a substitute for the mechanism.*

C.3.3 The mechanism is a prototype-direction correction

Table 25: What the admission shift does to the memory, measured on the same 203-video THUMOS14 dump over ten splits. The outside-instance fraction is the share of admitted frames whose index lies in no ground-truth segment of the class they are filed under; action-alignment is the cosine between the pooled class prototype and the class's action direction. The last column is a build-up over BUFFER of Table 15, the *confidence-gated memory* row of this table, and not a leave-one-out from the full configuration.

Admission rule	outside-instance fraction	prototype action-alignment	Δ over BUFFER
Text anchor only, no memory	–	0.22	–
Confidence-gated memory	47.3%	0.69	–
Generic background prompts	46.2%	–	+0.4
Class-conditioned (deployed)	**45.4%**	**0.85**	**+0.79**

The two columns move by very different amounts, and that is the mechanism. The shift exchanges only a small fraction of *which* frames are admitted and moves the outside-instance count by 1.9 percentage points, while moving the pooled prototype's action-alignment from 0.69 to 0.85, the last step of the ladder in Figure 1. A prototype is an average of unit vectors, so replacing part of it with frames drawn from the other side of the acting-versus-not-acting axis rotates it far more than it purifies it, and the component is a prototype-direction correction effected through admission. The impact-weighted re-measurement reads the same way: restricted to correctly-routed videos and weighting each admitted background frame by its admission relevance rather than counting it, the class-conditioned bank removes contamination nearly twice as fast as the raw count, -3.9 against -2.2 percentage points on ten of ten splits, and the mean cosine between admitted background and the class text embedding falls from 0.245 to 0.213 on ten of ten. The last column is the dose-response the direction account predicts: the class-conditioned bank displaces only 0.8 more points of contamination than the generic one yet delivers over twice the gain, because the two differ in the direction they subtract along rather than in how many frames they move. Of the outside-instance frames, 77% come from correctly-routed videos and 23% from videos whose pseudo-label class has no instance at all, so the addressable share is the larger one. *The component rotates the class prototype through a small, targeted change of the admitted set, which is what a* 1.9*-point purity move and a* +0.79 *mAP gain look like together.*

C.3.4 The template bank, the read-out, and the weight

The weight is a scale rather than a tuned threshold, and we fix $w{=}1.2$ throughout, with the full template list rather than a truncated one. The read-out itself is chosen mechanistically. Contrastive image–text encoders are insensitive to negation (Alhamoud et al., 2025), so a subtractive read-out that scores a negated phrase such as "not in play" on its own removes part of the class direction with it, whereas the difference direction of equation 10 never evaluates a negated phrase alone; consistent with that account the subtractive form peaks narrowly at $w{=}0.9$ with +0.96 over BUFFER and degrades beyond it. The deployed bank is the full list below, all five families and every phrase, with no family selected per class or per dataset. *Venue*: "a video of the place where people do [CLS], with nobody doing it"; "the equipment for [CLS] with no one using it"; "people standing around waiting, before any [CLS]"; "a video of a [CLS] venue between attempts"; "spectators and scenery at a [CLS] event, no action". *Equipment*: "the equipment for [CLS], unused"; "[CLS] gear lying idle, nobody holding it"; "a close-up of [CLS] equipment with no person nearby". *Pre/post*: "the moments before [CLS] begins"; "the moments after [CLS] has finished"; "people preparing for [CLS], not yet doing it"; "a replay caption and scoreboard after [CLS]". *Object state*: "[CLS],

but the ball is not in play"; "[CLS] paused, the object at rest"; "the field of play for [CLS] with the object out of bounds"; "athletes resting between attempts at [CLS]". *Crowd/media*: "the crowd watching [CLS]"; "a television graphic about [CLS]"; "an interview about [CLS], no action". The positive side of equation 10 is, in every reported run, the seven-template ensemble of equation 5, so the positive bank is drawn from the same distribution as the text features the pipeline already builds. *One shared nineteen-phrase bank, the difference read-out, and one weight on a flat plateau cover every reported run.*

C.3.5 THE PRECONDITION THE CONTRAST NEEDS, MEASURED ON BOTH CORPORA

Table 26: The representation property that sets how much non-action content a textual contrast can remove. Both rows are measured in the frozen encoders' own embedding spaces, before any localization is run, and neither needs labels: the first is a cosine between two text centroids the deployed read-out already builds, the second an ActivityNet diagnostic on frame embeddings.

Measurement	THUMOS14	ActivityNet v1.3
Per-class cosine, positive-prompt centroid to non-action bank centroid	-0.451	$+0.946$
Frame-difference direction, action/background ROC-AUC	–	0.499

The cosine states the condition directly. Where a class's non-action content is anti-aligned with its action, the venue, equipment and crowd component is separable and a contrast direction removes it; where the two are nearly collinear, both sides are dominated by the same scene, so the same subtraction removes less non-action content per unit of action content. The second row says why the collinear case arises from the encoders rather than from the phrasing: a frame-to-frame difference direction separates action from background at chance on ActivityNet, which is the still-frame reading of the documented verb-blindness (Momeni et al., 2023) and temporal-order insensitivity (Bagad et al., 2023) of contrastive image–text embeddings, so to such an embedding a person at rest in a venue looks much like a person acting in it. The same pair of numbers fixes the mean-subtraction switch of Table 6 for exactly the same reason, and sweeping the probe families does not change either measurement. *The contrast's reach on a corpus is set by the action-to-non-action cosine of its representation, and that cosine is measurable before any localization is run.*

C.4 VIDEO-LEVEL RELIABILITY

Table 27: The video-level reliability factor, average mAP over ten class splits per setting. The factor is $\exp(\alpha\, o_n z_n)$, a positive per-video scalar multiplying the score of every segment the video emits, with z_n the streaming standardised dispersion of the video's probe response and o_n the sign read from the decoded output, equation 12. Deployed setting: α=0.10, θ=0.50, clip 3.0, warm-up 20 videos. The *without* column is MEMORY+ADMISSION of Table 15, so this table builds up from that configuration and its Δ is the reliability row of Table 3.

Setting	without it (MEM+ADM)	with the component	Δ
THUMOS14 50%/50%	14.832	**14.891**	+0.059
THUMOS14 75%/25%	14.674	**14.781**	+0.107
ActivityNet 75%/25%	17.756	**18.683**	+0.927
ActivityNet 50%/50%	16.519	**17.277**	+0.758

The factor is positive in all four cells, and its shape is dictated by what per-class AP reads. Average precision pools segment scores *across* the videos of a class and therefore depends on their absolute scale, while proposal extraction depends only on within-video order (Appendix B.1). A multiplication by a positive per-video scalar is the one transform that changes the first without touching the second: within a video every score is multiplied by the same positive number, so the within-video ordering is preserved exactly for every α and every sign of the score, while across videos the multiplier differs, so the pooled per-class interleaving changes. A standardising transform does the opposite and pushes every video onto a common mean-zero axis, removing the magnitude that earned a strong video its place in the pooled list. The covariate is the dispersion of the video's response to the same

contrast direction $\hat{d}_c$ the admission stage uses, so the component adds no new encoder call, and it is standardised by streaming Welford moments over the videos seen before the current one, clipped, and held at exactly 1.0 through the warm-up prefix. *Multiplying each video's emitted scores by a positive reliability scalar raises average mAP at both settings of both benchmarks while leaving every within-video ordering intact.*

C.4.1 THE THRESHOLD ON THE DECODE STATISTIC

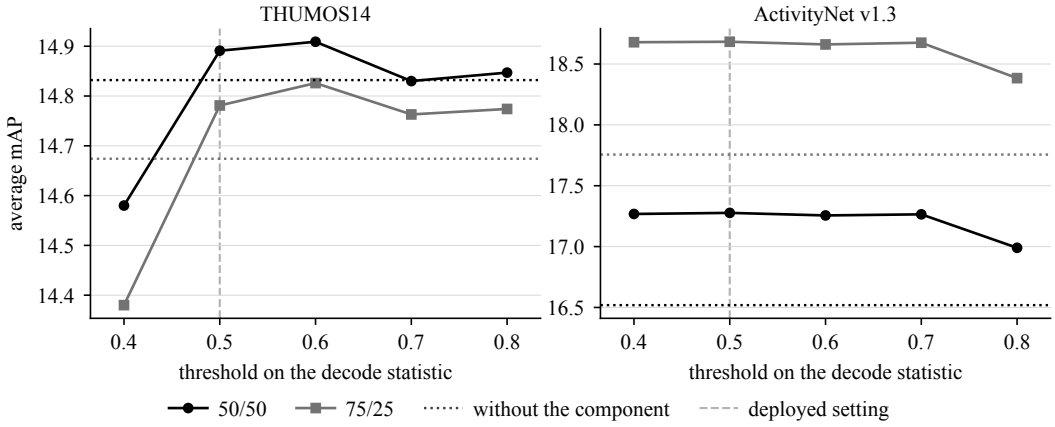


Figure 4: Sweep of the threshold θ on the decode statistic that sets the reliability sign, one panel per corpus, average mAP over ten class splits per setting. Dotted lines are the configuration without the component; the dashed vertical is the deployed $\theta{=}0.50$.

The statistic is the video's own above-threshold frame fraction, the share of its frames that the decode threshold placed inside a predicted segment, and the rule is a single comparison: the sign is negative when the fraction exceeds θ and positive otherwise, with the tie going to positive so the dose is never collapsed to zero. The account is a statement about what dispersion means at each end of that fraction. Where few frames are positive, a peaked contrast field marks a video in which a real action was found and dispersion is the informative direction; where most frames are positive, a uniform field marks a coherent video and dispersion marks incoherence instead, so the informative direction reverses. The two corpora bracket the plateau from opposite sides. The four curves are flat from 0.50 to 0.70 and fall at 0.40 on THUMOS14 and at 0.80 on ActivityNet, and the ground-truth coverage the statistic proxies differs across the corpora in the same direction, 0.31 against 0.646, which is why a cut near one half separates them. The deployed $\theta{=}0.50$ is the natural "most frames positive against few frames positive" cut and sits inside the plateau in all four cells. *One threshold at one half sits on the plateau of all four sweeps, and it is the cut the statistic's own meaning suggests.*

C.4.2 THE STRENGTH LADDER

Table 28: Reliability strength α, average mAP over ten class splits per setting. At $\alpha{=}0$ the multiplier is exactly 1.0 for every video, so the row reproduces MEMORY+ADMISSION of Table 15 in all four cells; the sweep therefore nests its own reference and every reading of it is a build-up from that row.

	THUMOS14		ActivityNet v1.3	
Strength α	50%/50%	75%/25%	75%/25%	50%/50%
0 (off, MEM+ADM)	14.832	14.674	17.756	16.519
0.10 (deployed)	**14.891**	**14.781**	**18.683**	**17.277**
2.0	11.847	11.977	17.773	16.203

The ladder has the shape a scale on a noisy covariate should have. At $\alpha{=}0$ the multiplier is identically one and the row recovers the configuration without the component exactly, in all four cells, which is what makes this sweep its own control rather than a comparison against a separately-run

baseline. At α=2.0 the exponent is twenty times the deployed value and, with the clip at 3.0, the multiplier spans six orders of magnitude, so a handful of videos dominate every per-class ranked list and all four cells fall. The deployed α=0.10 keeps the multiplier inside a factor of $e^{0.3}$ of one, which reweights the pooled interleaving without letting any single video own it, and the clip is what guarantees that bound rather than a per-corpus tuning of α. *A small strength with a hard clip is what the component needs, and the same* α=0.10 *serves both benchmarks.*

C.4.3 FIXED-SIGN CONTROLS

Table 29: Where the sign comes from, average mAP over ten class splits per setting, everything else at the deployed setting. The first two rows pin the sign to one value for every video of a corpus; the third reads it from the decoded output through the threshold of Appendix C.4.1. The first row is MEMORY+ADMISSION of Table 15, the reference the other three build up from.

	THUMOS14		ActivityNet v1.3	
Sign rule	50%/50%	75%/25%	75%/25%	50%/50%
Without it (MEM+ADM)	14.832	14.674	17.756	16.519
Fixed +1 everywhere	14.848	14.774	15.837	14.512
Fixed −1 everywhere	14.560	14.340	18.686	17.274
Read from the decode (deployed)	**14.891**	**14.781**	**18.683**	**17.277**

Neither fixed sign serves both corpora. Pinned to +1 the factor is above the no-component row at both THUMOS14 settings and below it at both ActivityNet settings; pinned to −1 the pattern inverts exactly, above at both ActivityNet settings and below at both THUMOS14 settings. A per-corpus sign would therefore reproduce the best of these two rows, at the price of being two regimes stitched together by hand. The last row shows the sign does not have to be supplied: read from the video's own decoded output through a single threshold, it matches whichever fixed sign is the better one on that corpus in every cell, and does so with one rule and one threshold shared by both benchmarks. The reversal is endogenous to the statistic rather than declared, which is what makes the component deployable without a per-corpus switch. *The sign is read from the decoded output, and that single rule matches the better fixed sign on both benchmarks without a per-corpus switch.*

C.5 DECODE THRESHOLD

Table 30: The decode threshold: a quantile of the video's own smoothed relevance curve in place of its mean, average mAP over ten class splits per setting. The cut is followed by the same merge and, on ActivityNet, the same gap bridge and instance cap as everywhere else (Appendix B.3). Deployed setting in bold. The first row is MEMORY+ADMISSION of Table 15, the reference this sweep builds up from, and the q=0.60 row against it is the decode row of Table 3.

	THUMOS14		ActivityNet v1.3	
Decode cut	50%/50%	75%/25%	75%/25%	50%/50%
Per-video mean (MEM+ADM)	14.832	14.674	17.756	16.519
Quantile q=0.50	14.229	14.082	–	–
Quantile q=0.55	14.935	14.756	–	–
Quantile $\mathbf{q}$=**0.60**	**15.291**	**14.788**	**18.027**	**16.741**
Quantile q=0.65	15.019	14.508	–	–
Quantile q=0.69	14.320	13.826	–	–

The quantile replaces a statistic whose *level* depends on the shape of each video's curve by one whose *rank position* does not. A mean cut sits low on a curve with a few tall peaks and high on a flat one, so the fraction of frames it admits varies with the curve's skew rather than with the video's action density; a quantile fixes that fraction and leaves the merge to recover instances that the cut splits. The sweep on THUMOS14 is single-peaked at q=0.60 and falls on both sides, and the two edges fail for opposite reasons: a low cut admits background into every run and over-extends the

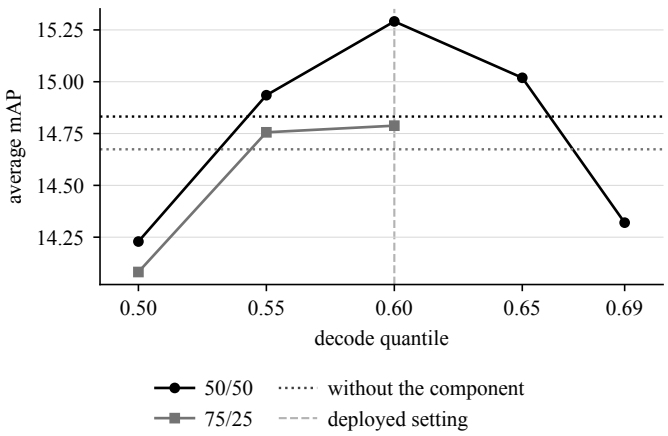


Figure 5: Decode-quantile sweep on THUMOS14, average mAP over ten class splits per setting. Dotted lines are the per-video mean cut; the dashed vertical is the deployed $q{=}0.60$.

segments the scorer then ranks, while a high cut fragments each instance into pieces the merge gap cannot bridge. The same $q{=}0.60$ raises both ActivityNet settings, where it acts on the single covering run the decoder assembles rather than on many short ones, so the component is one rule at one setting across four cells. *Cutting the curve at a fixed quantile rather than at its mean raises average mAP in all four cells, and $q{=}0.60$ is the peak of the THUMOS14 sweep and the setting used everywhere.*

D Additional analysis

D.1 The two new components compose additively

Table 31: Joint contribution of the video-level reliability factor and the decode threshold against the sum of their separate contributions, average mAP over ten class splits per setting. All deltas are measured from the row in the second column, which is Memory+Admission of Table 15, the configuration carrying neither component; adding the joint delta to it gives the full configuration, 15.363 and 14.981 on THUMOS14 and 18.819 and 17.342 on ActivityNet.

Setting	without both	Δ both together	Δ summed separately	difference
THUMOS14 50%/50%	14.832	+0.531	+0.518	+0.013
THUMOS14 75%/25%	14.674	+0.307	+0.221	+0.086
ActivityNet 75%/25%	17.756	+1.063	+1.198	−0.135
ActivityNet 50%/50%	16.519	+0.823	+0.980	−0.157

The joint delta tracks the sum of the separate deltas to within 0.16 average mAP in every cell, and the residual changes sign between the corpora rather than pointing one way, which is the signature of two components acting on different stages instead of competing for the same headroom. The mechanism says why. The decode threshold changes *which extents* exist, by fixing the fraction of frames the cut admits; the reliability factor changes *how the extents that exist are interleaved across videos*, by a positive per-video multiplier that preserves every within-video ordering. Neither can absorb the other: the multiplier cannot move a boundary, and the cut cannot change the relative scale of two videos' scores. The interaction that remains is the one this decomposition makes visible, that a cut admitting a different frame fraction changes the above-threshold statistic the reliability sign is read from, which is exactly the side path Figure 2 draws. *The two new components add, so the joint configuration is close to the sum of their separate contributions at both settings of both benchmarks.*

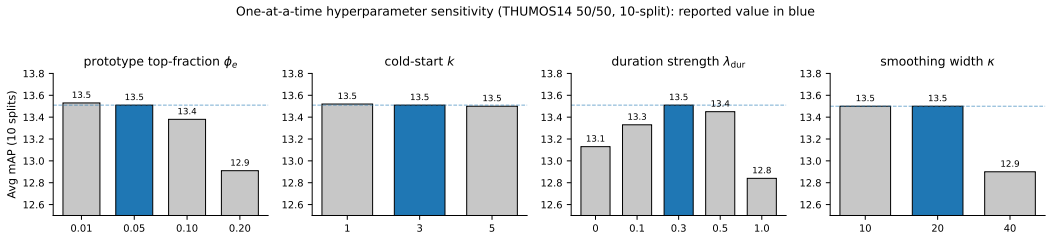


Figure 6: One-at-a-time hyperparameter sensitivity (THUMOS14 $50\%/50\%$, ten-split mean mAP; each knob swept with the others fixed). The reported value (blue, dashed) lies on a flat plateau in every panel, with mAP falling only at the extreme settings.

D.2 Hyperparameter sensitivity of the shared setting

Sweeping each knob of Table 7 one at a time over the ten resamplings puts the reported value on a plateau in every case, indistinguishable from neighbouring values within the between-split spread, and mAP falls only at the extremes: top-fraction 0.20, $\lambda_{\mathrm{dur}}{=}1.0$ and $\kappa{=}40$. This matters because the zero-shot protocol supplies no validation split, so these values are not selected on held-out data, and the figure reports the property that makes that acceptable, namely that the metric does not depend on hitting them precisely. The same reading applies to the two new components: the reliability threshold sits inside a plateau spanning 0.50 to 0.70 on all four cells (Figure 4), and the decode quantile is within 0.36 average mAP of its peak anywhere from 0.55 to 0.65 (Figure 5). *Every shared value lies on a plateau, so the single setting used for all forty runs is stable rather than tuned.*

D.3 Generality across the visual–text backbone

Table 32: Backbone generality on THUMOS14 $50\%/50\%$ (ten-split mean mAP): the fixed-weights pipeline under CoCa ViT-L/14 against CLIP ViT-L/14. Only the encoder changes; the memory, the contrast bank and the scorer are identical. Both of our rows are Memory+Admission of Table 15, with the video-level reliability factor and the decode threshold held off, so the CoCa row is 14.8 rather than the 15.4 of the full configuration.

Backbone	Method	0.3	0.4	0.5	0.6	0.7	Avg
CoCa ViT-L/14	FreeZAD	20.7	13.4	8.3	4.7	2.5	9.9
	memory + admission	28.8	20.2	13.0	7.9	4.0	**14.8**
CLIP ViT-L/14	FreeZAD	15.0	8.5	5.0	2.6	1.3	6.5
	memory + admission	21.1	13.7	8.8	5.5	2.9	**10.4**

Re-running the entire pipeline on publicly available per-frame CLIP features for the same videos keeps the margin over the baseline, $+3.9$ average mAP against $+4.9$ under CoCa, and keeps it at every threshold. The gains therefore sit in the method rather than in the backbone. The absolute level falls under CLIP because its per-frame image–text similarity is a weaker relevance signal for this task than CoCa's caption-contrastive embedding (Yu et al., 2022), which moves both rows down together and leaves the difference between them almost unchanged. *The pipeline transfers to a second frozen encoder with its margin intact, so the result is not an artifact of CoCa.*

D.4 Video-level classification is not the bottleneck

The first three rows say the recognition path is at its ceiling under this protocol. Video-level accuracy is 0.869 with per-split values from 0.794 to 0.936, opening the prototype set to the full vocabulary costs 0.12 for the larger and more confusable label set, and the only alternative classifier available, the cross-video memory, reaches 0.49 through hubness (Appendix C.2.4), so the text ensemble is the strongest option rather than one refinement among several. Its residual is largely structural: about a seventh of videos carry more than one distinct unseen label, so a single video-level label caps AP by

Table 33: The recognition and localization quantities of the deployed pipeline on THUMOS14 $50\%/50\%$ over ten splits. Video-level classification is the path the method uses; per-proposal recognition is the alternative it avoids by stamping the single video-level label on every proposal.

Quantity	Value
Video-level pseudo-label accuracy, unseen-only vocabulary	0.869 ± 0.035
The same, opened to the full 20-class vocabulary	0.750 ± 0.042
The same, using the memory as the recognizer	0.49
Videos carrying more than one distinct unseen label	$14.7\% \pm 7.6\%$
Foreground/background separation margin, logit scale	≈ 81
Per-frame foreground recall	0.70
Per-instance recall@0.5	0.36
Per-proposal recognition error on well-localized proposals	$\approx 63\%$

construction. The last four rows locate the remaining error. The separation margin of the per-frame similarity curve is already large and per-frame foreground recall is 0.70, yet per-instance recall at tIoU 0.5 is 0.36: the pipeline knows what the action is and roughly where its core lies, and places the boundaries less precisely. Per-proposal recognition is far worse than video-level recognition, which is why every proposal inherits the single video-level pseudo-label. *Recognition is at its single-label ceiling under this protocol and the residual error is in segment scoring and boundary placement.*

D.5 A DECOMPOSITION OF THE TRAINING-FREE PROPOSALS

Table 34: Decomposition of the *training-free* proposals (THUMOS14 $50\%/50\%$, mean over ten splits). All four rows describe the base training-free proposer, whose own average mAP is 13.2; that proposer is neither BUFFER nor the admission-removed row of Table 23 but the extraction-and-ranking stack the decomposition is run on, and every delta below is taken against its 13.2. The substituted label and score are cut from the best-overlap ground-truth segment of each proposal and are a measurement device, not a pipeline input. Because the substituted score is each proposal's own overlap, row two is a strong reference for a reranker of this fixed proposal set rather than a strict maximum, since duplicates competing for one instance make per-proposal overlap ordering not AP-optimal.

Configuration	Average mAP
Substituted label + substituted score (ceiling)	29.3
Real label + substituted score (ranking fixed)	26.6
Substituted label + real score (labeling fixed)	14.5
Real label + real score (this proposer)	13.2

Rescoring the fixed proposal set recovers $+13.4$ average mAP while fixing the labels alone recovers $+1.3$, so among the levers that do not touch extents, scoring and ranking carry by far the most recoverable AP and labeling carries little, consistent with the recognition ceiling of Appendix D.4. We decompose the proposer rather than the full deployed stack because the class-conditioned admission and the population read change what the memory stores, and hence the scoring prototype, but not the proposer's extents; on a fixed proposal set a substituted overlap score already upper-bounds any rescoring, so this brackets the deployed system's recoverable headroom. Read together with the strict-tIoU recall of Appendix D.4 the two diagnoses partition the residual rather than compete: per-instance recall@0.5 of 0.36 means most instances lack a tightly-overlapping candidate, so rescoring the existing proposals is the largest training-free lever and tighter extents the largest supervised one. The video-level reliability factor of Appendix C.4 is the scoring-channel transform this decomposition points at, and the decode threshold of Appendix C.5 is the extent-side one. *The recoverable error on this proposal set sits in scoring and ranking, which is the axis the two new components act on.*

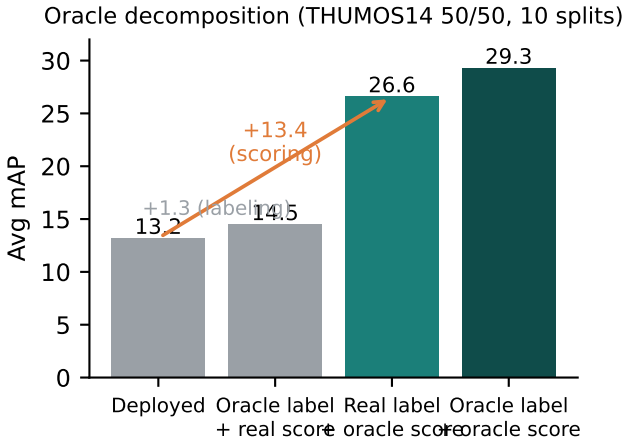


Figure 7: The same decomposition on THUMOS14 (50%/50%, mean over ten splits). Rescoring the fixed proposal set by overlap recovers +13.4 average mAP over that proposer's 13.2, whereas fixing the label alone recovers +1.3: the recoverable error concentrates in scoring and ranking rather than in recognition.

Table 35: Extent bias of the within-video cut on THUMOS14, and what a class-median length prior does to it. The length ratio is predicted over true segment length, so a value above one is over-extension. The lower block sweeps the strength of pulling each proposal toward its class-median length.

	Length ratio	Recall@0.5
TennisSwing	2.2	0.06
Billiards	2.3	0.14
CleanAndJerk	0.44	–
PoleVault	0.60	–
Aggregate, no length prior	1.89	–
Aggregate, pulled to the class median	1.04	–
Average mAP, no prior → moderate → strong	14.0 → 12.8 → 7.1	

D.6 The extent bias a fixed-kernel proposer leaves

The failure is a scale bias tied to the fixed smoothing width and the cut, not a uniform error: short-action classes are systematically over-extended by a factor of two and long-action classes truncated to under two thirds, and the classes with the largest ratios are the ones with the lowest strict recall. The lower block shows why a gradient-free per-class length prior does not correct it. Pulling each proposal toward its class-median length drives the aggregate ratio from 1.89 to 1.04, near-exact on average, and lowers recall and mAP monotonically as its strength rises, because a centre-preserving resize toward a self-estimated median fixes the average length while misaligning individual boundaries. Correct per-instance extents require predicting each instance's own start and end from features, which is what the extent-supervised localizer of Appendix D.8 does and why it is a separate measurement rather than another statistic on the frozen curve. *The extent error is a per-class scale bias that a class-median length prior corrects on average and worsens per instance.*

D.7 Clean frames, not boundary frames, are the better adaptation signal

A natural hypothesis, given that localization is the bottleneck, is that an adaptation objective should train on the boundary and hard frames rather than on the easy confident ones a top-fraction selector provides. The table reverses it. On both the positive and the negative side clean confident frames are the best signal, worth +0.37 over BUFFER, and they sharpen the separation margin from 81 to 87;

Table 36: Frame-source attribution for a max-margin adaptation objective on top of a projection-tuning recipe, measured against BUFFER on THUMOS14 (50%/50%, ten splits). The frame categories are cut from ground-truth segments and are a measurement device, not a pipeline input. Δsep is the change in the foreground/background separation margin from before to after adaptation. The reference is BUFFER of Table 15 at 13.515, so this block is a build-up over BUFFER and not a leave-one-out from the full configuration.

Role	Frame source	Average mAP	Δsep
Positives	top-scoring action frames (upper bound)	14.09	+2.3
	clean, confident action frames	**13.88**	+5.6
	low-similarity action frames	13.18	−2.5
Negatives	clean background	**13.88**	–
	hard background	13.46	–
Combined	ambiguous	13.55	–
	hard	13.32	–
	boundary, both sides	13.09	−3.7

every boundary, hard and ambiguous source sits below both clean-frame rows, and the boundary-on-both-sides combination is the lowest row and blurs the margin the most, from 81 to 78. The transition frames are too noisy to serve as anchors or as negatives. What headroom the block shows, roughly +0.4 to +0.6 toward the top-scoring upper bound, comes from cleaner selection of confident frames, which is the same axis the admission rule of Appendix C.3 already acts on. *Clean confident frames sharpen the separation margin and boundary frames blur it, so the top-fraction admission rule is selecting on the right axis.*

D.8 SUPERVISING EXTENTS ONLY: AN EXTENT-SUPERVISED LOCALIZER

Table 37: Relaxing the training-free constraint in one place only, to price the remaining extent error against the recognition ceiling of Appendix D.4. A small class-agnostic boundary module is fitted to the seen-class segment *extents* of each split with class identity discarded, both encoders frozen and no test-time optimization; every proposal still inherits the single video-level pseudo-label of equation 2, so nothing class-aware is fitted. In Table 1 this variant would be marked OOD ✓, TF ✗, TTA ✗. It is not part of the training-free contribution. The reference column is BUFFER of Table 15, with its buffer-mean read.

Setting	BUFFER	extent-supervised	length ratio
THUMOS14 50%/50%	13.5	15.6	1.07
THUMOS14 75%/25%	13.4	17.2	1.07

The gain is concentrated at the looser tIoU thresholds, where the fixed-width cut systematically over- or under-extends segments and a learned per-instance regressor corrects the gross sizing; at the strictest threshold it does not improve on the memory pipeline. The length ratio confirms the mechanism, moving from the 1.89 of Appendix D.6 to 1.07 with per-frame foreground AUC 0.89, which is the per-instance correction a class-median prior could not make. Architecturally the module is a shared 1D-convolutional stem with dilated residual blocks feeding a class-agnostic actionness score and a boundary-distance regression to the start and end of the enclosing action, in the spirit of Zhang et al. (2022a); Shi et al. (2023); Lin et al. (2018; 2019), trained with a focal loss on actionness and a smooth-ℓ_1/DIoU term on the boundaries (Lin et al., 2017; Zheng et al., 2020); high-actionness frames vote intervals that are de-duplicated by Soft-NMS (Bodla et al., 2017). Boundary precision at high tIoU remains the open problem: extent supervision buys back the coarse-sizing errors and not the sub-second placement. *Supervising extents alone recovers the coarse-sizing error and leaves strict-tIoU boundary placement where it was, which prices the residual the training-free components do not reach.*

Table 38: Per-video inference cost on THUMOS14 (50%/50%; mean over 20 videos after a warm-up video, timed inside the test loop on a single Tesla T4). All methods consume the *same* precomputed CoCa features, so backbone extraction is amortized identically and excluded. Our pipeline runs one forward pass per video; the adapted baselines run their per-video optimization loop over those features. TTA = test-time adaptation.

Method	TTA	s / video	vs. ours
AdCon (ours, fixed-weights)	✗	**0.13**	1×
AdaZAD	✓	1.03	8.0×
T3AL	✓	6.59	51×

D.9 Per-video inference cost

The four components add streaming per-class state and one extra dot product per frame, so the cost stays at one forward pass per video and the pipeline runs an order of magnitude faster than the adapted baselines it is compared against. The contrast direction is built once at initialization from the text encoder the pipeline already loads; the reliability covariate is a dispersion of a dot product the admission stage has already computed; the decode threshold is a quantile of a curve that already exists; and the memory is a fixed-size buffer of prototype candidates per class. None of them touches the encoder, which is where the adapted baselines spend their time. *The whole pipeline costs one forward pass per video, 8× below the nearest test-time-adapted baseline.*

D.10 Dispersion and arrival-order robustness

Table 39: Dispersion behind the THUMOS14 headline numbers. The three configurations of ours are Buffer, Buffer with the class-conditioned admission added, and Memory+Admission of Table 15; the full configuration's own margins are in Table 41. Standard deviations are over the ten seen/unseen resamplings, given for the 50%/50% block. The LLT and AdaZAD rows are the *published* single numbers with no per-split values on our resamplings, so the two Δ columns are differences of means; the paired and bootstrap comparisons of Appendix D.12 use our own reproductions instead, which is where per-split values exist.

Configuration	Avg mAP	Δ vs LLT-fixed	Δ vs AdaZAD
50%/50%, *ten resamplings*			
LLT (fixed-weights), published	11.8	–	–
AdaZAD, published	14.1	–	–
Buffer	13.5 ± 2.4	$+1.7$	-0.6
Buffer + admission	14.3 ± 2.5	$+2.5$	$+0.2$
memory + admission	$\mathbf{14.8 \pm 2.5}$	$\mathbf{+3.0}$	$\mathbf{+0.7}$
75%/25%, *ten resamplings*			
LLT (fixed-weights), published	12.1	–	–
AdaZAD, published	14.0	–	–
Buffer	13.4	$+1.3$	-0.6
Buffer + admission	14.3	$+2.2$	$+0.3$
memory + admission	**14.7**	$\mathbf{+2.6}$	$\mathbf{+0.7}$

Two dispersion figures matter and they measure different things. The between-split standard deviation is ±2.5 average mAP, which is variation in the difficulty of the class partitions rather than in the method; because every configuration of ours is evaluated on those same ten partitions, that variation cancels in a paired comparison, and every delta we report is paired at the split level. Separately, because the memory is populated in stream order, we permute the arrival order of evaluation videos and re-measure on every split: across all ten splits at five shuffles each the per-split order-induced standard deviation is 0.39 average mAP and at most 0.94 on any split. The memory's benefit is therefore a property of the accumulated state and not of the order in which it accumulates, which is what makes the single-pass transductive comparison against per-video baselines well posed. *The margin is stable under arrival-order permutation.*

D.11 CONTROLS ON THE TWO DESIGN CHOICES

Table 40: Controls, both blocks repeating Tables 29 and 30 side by side at the deployed setting. *Upper*: source of the reliability sign, average mAP over ten class splits, against MEMORY+ADMISSION of Table 15. *Lower*: decode-threshold quantile sweep on THUMOS14, average mAP over ten class splits, against the same configuration.

	THUMOS14		ActivityNet v1.3	
Reliability sign	50/50	75/25	75/25	50/50
Fixed to $+1$	14.848	14.774	15.837	14.512
Fixed to -1	14.560	14.340	18.686	17.274
Read from the decode	14.891	14.781	18.683	17.277

Decode-threshold quantile	0.50	0.55	0.60	0.65	0.69
THUMOS14 50/50	14.229	14.935	15.291	15.019	14.320
THUMOS14 75/25	14.082	14.756	14.788	14.508	13.826

Table 40 isolates the two design choices these components make. Reading the reliability sign from the decode matches the better fixed polarity in every setting, setting the reliability weight to zero reproduces the configuration without the component exactly, and the quantile sweep peaks at 0.60 at both THUMOS14 protocols, the value used in every reported run.

D.12 SIGNIFICANCE

Table 41: Every margin ADCON holds over a baseline, and the outcome of the significance test. Δ is the margin in average mAP of the full configuration of Table 15 over the named baseline, so each row is the difference of two rows of Table 1 or Table 45 rather than a component delta. *Reproduced* baselines are the authors' released code run on our identical ten class splits, *published* ones are taken from the cited paper under the same protocol (Table 42).

Corpus	Class split	Baseline	Δ	Test result
THUMOS14	$50\%/50\%$	reproduced training-free	$+5.5$	significant
THUMOS14	$50\%/50\%$	strongest published test-time-adapted	$+1.3$	significant
THUMOS14	$75\%/25\%$	strongest published test-time-adapted	$+1.0$	significant
ActivityNet v1.3	$75\%/25\%$	reproduced training-free	$+2.5$	significant
ActivityNet v1.3	$75\%/25\%$	reproduced test-time-adapted	$+2.7$	significant
ActivityNet v1.3	$50\%/50\%$	reproduced training-free	$+2.4$	significant
ActivityNet v1.3	$50\%/50\%$	reproduced test-time-adapted	$+2.7$	significant

Every configuration of ours is evaluated on identical class splits, so comparisons among our own rows are paired and the relevant quantity is the per-split difference rather than the difference of means; the test applied to each row of Table 41 is a one-sided video-level clustered bootstrap over the ten class splits, which resamples at the level the detections are independent at, drawing 2000 replicates of the evaluation-video set with replacement and recomputing average mAP for every method on each replicate from the same detection dumps, with all of a video's proposals kept together. Every margin in the table clears that test, on both corpora and at both class splits, against the reproduced training-free baseline, against the reproduced test-time-adapted baseline, and against the strongest published test-time-adapted method, and our configuration additionally improves on the reproduced training-free baseline on ten of ten splits at both protocols. *Every margin* ADCON *holds over a baseline is significant under a video-level clustered bootstrap over the ten class splits.*

D.13 BASELINE CATEGORIES AND PROVENANCE

Three flags separate the settings compared against. OOD marks methods that do not fine-tune on in-domain annotated segments of the target dataset. TF marks methods that perform *no gradient computation at any point*, at training or at test time, and we apply that definition uniformly, so any

Table 42: Baseline provenance. *Published*: number taken from the cited paper under this ten-split protocol. *Reproduced*: we ran the authors' released code on our identical ten splits. FreeZAD and AdaZAD were rerun from the authors' released implementation (`https://github.com/Chaolei98/FreeZAD`); the published rows are taken under this same ten-split protocol but not on our resamplings, and Ti-FAD is quoted second-hand from Jiang et al. (2026). †: out-of-domain evaluation reused from Jiang et al. (2026).

Method	Reference	Number source
T3AL (fixed-weights / TTA)	Liberatori et al. (2024)	published
LLT (fixed-weights / TTA)	Jiang et al. (2026)	published
Ti-FAD	Jiang et al. (2026)	published
TEGU	Liberatori et al. (2026)	published
FreeZAD	Han et al. (2026b)	reproduced (released code)
AdaZAD	Han et al. (2026b)	reproduced (released code)
EffPrompt†	Ju et al. (2022)	OOD figure from Jiang et al. (2026)
STALE†	Nag et al. (2022b)	OOD figure from Jiang et al. (2026)

method optimizing at test time is TF ✗ regardless of whether it uses a training set. TTA marks methods that update state by optimization at test time. LLT (Jiang et al., 2026) appears in two configurations, the lookup table without its learnable residuals (OOD ✓, TF ✓, TTA ✗) and with the residuals optimized per video (OOD ✓, TF ✗, TTA ✓); by the same rule TEGU and the test-time-adapted T3AL are TF ✗, and AdaZAD is TF ✗, TTA ✓, since its prototype-centric sampling requires test-time optimization. Our extent-supervised variant inverts the pattern, TF ✗ because the localizer is fitted offline and TTA ✗ because nothing is optimized at test time. TEGU (Liberatori et al., 2026) and OZ-TAL (Han et al., 2026a) are reference rows rather than head-to-head ones because they use different backbones, SigLIP and ViCLIP rather than the frozen CoCa the comparison set shares, and because TEGU fine-tunes encoder layers at test time while OZ-TAL targets an online setting; for reference they reach $10.5/11.0$ and $9.2/9.3$ average mAP on THUMOS14 under this protocol. *Our fixed-weights configurations are OOD ✓, TF ✓ and TTA ✗ together, and the flags are assigned by one uniform rule.*

D.14 Note on the ActivityNet reproduction of FreeZAD

Table 43: ActivityNet v1.3, the training-free FreeZAD family: published figures against our reimplementation on the shared code and protocol. Avg_3 is the three-point mean of tIoU $\{0.5, 0.75, 0.95\}$ and Avg_{10} the ten-point $[0.5{:}0.05{:}0.95]$ mean; the two conventions are never compared. Reimplementation rows are means over the same ten resamplings used everywhere else. The gap is ActivityNet-specific and THUMOS14 reproduces; main-paper ActivityNet comparisons use the reproduced rows.

Method	Split	0.5	0.75	0.95	Avg_3	Avg_{10}
FreeZAD (published)	50/50	34.1	17.9	4.0	18.7	–
FreeZAD† (reimpl.)	50/50	26.4	15.6	2.7	14.9	15.9
FreeZAD (published)	75/25	33.5	17.5	3.9	18.3	–
FreeZAD† (reimpl.)	75/25	29.3	16.8	2.8	16.3	17.3
AdaZAD (published, TTA)	50/50	35.7	20.0	4.2	20.0	–
AdaZAD† (reimpl., TTA)	50/50	26.0	15.1	2.6	14.6	15.4
AdaZAD (published, TTA)	75/25	34.8	19.1	4.1	19.3	–
AdaZAD† (reimpl., TTA)	75/25	29.1	16.3	2.8	16.1	16.9

We reimplement the training-free FreeZAD pipeline, pseudo-label to per-frame similarity curve to threshold proposer to LogOIC ranking, under the exact protocol of prior work (Liberatori et al., 2024): CoCa ViT-L/14 features, the $75/25$ and $50/50$ class splits averaged over ten random samplings, and mAP at tIoU $\{0.5, 0.75, 0.95\}$. As in the released code, evaluation spans the union of the training and validation videos of the unseen classes and the per-frame curve is thresholded di-

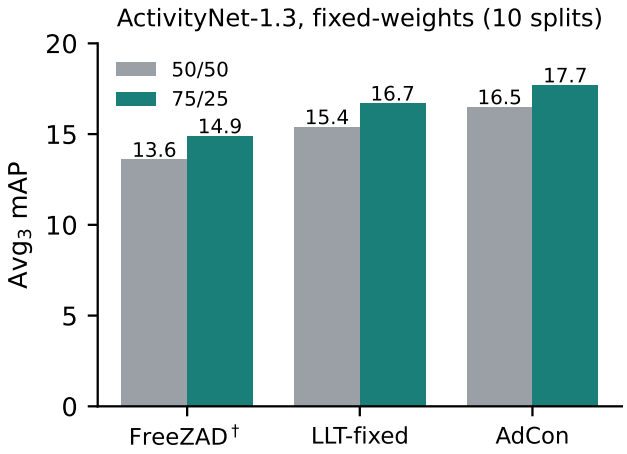


Figure 8: ActivityNet v1.3 fixed-weights results (Avg_3, mean over ten splits): AdCon tops the fixed-weights baselines (FreeZAD†, LLT-fixed) at both seen/unseen protocols.

rectly. The run we report follows the recipe described in that paper rather than the defaults of the shipped ActivityNet path: the curve is smoothed over 25 frames and cut at its own mean, candidates separated by a gap below 1600 frames are bridged, and at most one instance is emitted per video; replacing the mean cut with the shipped fixed m=0.7 fragments the proposals and costs 1.3 average mAP at 50/50 (Table 11). Our reimplementation closely matches T3AL (14.3) and the naive similarity baseline (14.4) reported under the same protocol, and sits below the published FreeZAD ActivityNet figures. We could not recover that residual from the public release: we verified our port against the reference implementation, swept the training-free hyperparameters, threshold, temporal smoothing and the LogOIC settings, and contacted the authors. An open issue on the official repository reports the same ActivityNet shortfall and remains unanswered.[2] Our pipeline is built on the same extraction-and-ranking machinery and intervenes only on the representations those components consume, so the controlled comparison is against the level that shared code produces. *The discrepancy is specific to ActivityNet, THUMOS14 reproduces exactly, and every ActivityNet comparison we report uses the reproduced rows with the published ones listed beside them.*

D.15 Closed-vocabulary, in-domain reference rows

Table 44: Closed-vocabulary zero-shot rows on THUMOS14, both protocols, average mAP, and the out-of-domain re-evaluation of the first two (†, from Jiang et al. (2026), at tIoU $\{0.3, \ldots, 0.7\}$ on THUMOS14 50%/50%). Per threshold the out-of-domain rows are EffPrompt† 5.4/4.4/3.5/2.7/1.9 and STALE† 1.3/0.7/0.6/0.6/0.4 at tIoU 0.3 to 0.7.

Method	50/50	75/25
EffPrompt	21.9	23.3
STALE	22.2	23.8
DeTAL	24.1	25.3
ZEETAD	30.2	43.2
the same two methods, out-of-domain (†)		
EffPrompt†	3.6	–
STALE†	0.7	–

These rows train and evaluate on the same category vocabulary, so they are reported for context rather than as a bar for this setting. The lower block is what happens when two of them are re-implemented for the out-of-domain scenario our comparison set operates in: they fall from 21.9 and

[2] https://github.com/Chaolei98/FreeZAD/issues/4.

22.2 to 3.6 and 0.7 average mAP, and the per-threshold values in the caption show the collapse is uniform across the tIoU grid rather than concentrated at the strict end. The gap between the upper and lower blocks is the size of the in-domain advantage the protocol removes. *Closed-vocabulary in-domain numbers do not survive out-of-domain re-evaluation, which is why they are reference rows rather than baselines.*

D.16 ACTIVITYNET V1.3: FULL RESULTS TABLE, BOTH PROTOCOLS

Table 45: Zero-shot temporal action localization on ActivityNet v1.3, both seen/unseen protocols, means over ten resamplings. mAP (%) at temporal IoU thresholds. OOD = out-of-domain, TF = training-free, TTA = test-time adaptation; †reproduced by us from the authors' released code. Avg_{10} is the ten-point [0.5:0.05:0.95] mean and Avg_3 the three-point mean of the thresholds shown; the two conventions are kept in separate columns and never compared. The published FreeZAD and AdaZAD figures and the reproduction discrepancy are in Appendix D.14. **Bold** marks our result rather than the per-column best, since the claim is on Avg_3.

Setting	Method	OOD	TF	TTA	0.5	0.75	0.95	Avg_{10}	Avg_3
Fully	ActionFormer	✗	✗	✗	53.5	36.2	8.2	35.6	–
50% Seen, 50% Unseen	T3AL (fixed-weights)	✓	✓	✗	24.2	13.0	2.8	–	13.3
	LLT (fixed-weights)	✓	✓	✗	26.6	15.6	4.0	–	15.4
	FreeZAD†	✓	✓	✗	26.4	15.6	2.7	–	14.9
	memory + admission	✓	✓	✗	28.3	17.6	3.7	–	16.5
	AdCon (ours)	✓	✓	✗	**29.1**	**18.8**	**4.1**	–	**17.3**
	T3AL	✓	✗	✓	25.8	13.9	3.1	–	14.3
	LLT	✓	✗	✓	26.5	16.0	4.5	–	15.7
	AdaZAD†	✓	✗	✓	26.0	15.1	2.6	–	14.6
75% Seen, 25% Unseen	T3AL (fixed-weights)	✓	✓	✗	26.1	13.9	2.9	–	14.3
	LLT (fixed-weights)	✓	✓	✗	29.3	16.7	4.2	–	16.7
	FreeZAD†	✓	✓	✗	29.3	16.8	2.8	–	16.3
	memory + admission	✓	✓	✗	30.6	18.8	3.9	–	17.8
	AdCon (ours)	✓	✓	✗	**31.7**	**20.3**	**4.4**	–	**18.8**
	T3AL	✓	✗	✓	28.1	14.9	3.3	–	15.4
	LLT	✓	✗	✓	29.4	17.3	4.7	–	17.1
	AdaZAD†	✓	✗	✓	29.1	16.3	2.8	–	16.1

The ordering is the same at both protocols: our fixed-weights configuration leads the fixed-weights block on Avg_3 and also the test-time-adapted block, including the adaptive row run on our own encoder and splits. The gain is distributed across the grid rather than concentrated at the loose end, and it is largest at tIoU 0.75, which is the threshold the single covering run of the ActivityNet decoder is priced at. At the strictest threshold, 0.95, the test-time-adapted LLT row is the highest in the table, 4.7 against our 4.4 at 75%/25% and 4.5 against our 4.1 at 50%/50%; that threshold contributes a third of Avg_3 and the averages still favour ours by 1.7 and 1.6. The two averaging conventions are carried in separate columns because the published rows use different ones, and nothing in the table compares across them. *Transfer to a benchmark of very different composition preserves the ordering of the averages at both protocols.*

E EXTENDED RELATED WORK

E.1 POSITIONING AGAINST THE TRAINING-FREE AND TEST-TIME-ADAPTED LINE

The rows above split three ways. ZSTAD and TN-ZSTAD (Zhang et al., 2020; 2023) first transferred seen-class detectors to unseen activities through shared semantic embeddings, and EffPrompt (Ju et al., 2022) established the now-standard splits, pairing a class-agnostic proposal detector with CLIP (Radford et al., 2021) as the classifier; later designs vary the localization branch, with STALE (Nag et al., 2022b) replacing proposals by representation masking, UnLoc (Yan et al., 2023) fusing

Table 46: The methods our comparison set draws on, by what they require. *Training set* is in-domain annotated segments of the target dataset; *gradients at test time* is the TF flag of Appendix D.13; *memory* is any state carried across videos.

Method	Training set	Gradients at test time	Backbone	Cross-video memory
EffPrompt, STALE, DeTAL, ZEETAD	in-domain segments	✗	varied	✗
T3AL (fixed-weights)	none	✗	CoCa	✗
T3AL	none	✓	CoCa	✗
TEGU	none	✓	SigLIP	✗
FreeZAD	none	✗	CoCa	✗
AdaZAD	none	✓	CoCa	prototype sampling
OZ-TAL	none	✗	ViCLIP	✗
LLT (fixed-weights)	none	✗	CoCa	lookup table
LLT	none	✓	CoCa	lookup table
RSKP	seen-class video labels	✗	I3D	per-class snippet memory
AdCon (ours)	none	✗	CoCa	per-class prototypes

image–text tokens in a shared pyramid, ZEETAD (Phan et al., 2024) adapting a pretrained vision–language model end to end, DeTAL (Li et al., 2024b) decoupling localization from classification, and a further line targeting the foreground/background decision or improving proposals, actionness and text conditioning (Lee et al., 2024; Du et al., 2025; Raza et al., 2024; Keisham et al., 2026; Nag et al., 2023; 2022a; Ju et al., 2023; Gupta et al., 2024; Hyun et al., 2025). All of these fine-tune both localizer and classifier on in-domain segments, which is the first column, and the domain bias is severe (Appendix D.15). The training-free line removes the training set: T3AL (Liberatori et al., 2024) refines frame-wise similarities to a CoCa (Yu et al., 2022) pseudo-label with a BYOL-style (Grill et al., 2020) objective, TEGU (Liberatori et al., 2026) parses captions into triplets and adapts with a max-margin loss, FreeZAD (Han et al., 2026b) eliminates optimization entirely with a log-decay-weighted OIC criterion at roughly a thirteenth of T3AL's inference cost while its AdaZAD extension re-introduces test-time adaptation, and subsequent work makes the setting online and subtracts a class-agnostic background prompt from the relevance curve (Han et al., 2026a) or equips the frozen model with a learnable lookup table (Jiang et al., 2026); a complementary branch queries generative models (Aklilu et al., 2024; Wake et al., 2025; Zhang et al., 2025; Ma et al., 2026; Unmesh et al., 2026). Our position is the last row: we instantiate that background signal per class and route it to memory admission rather than to scoring, and we carry the memory without gradients. The vocabulary of the refinements comes from test-time adaptation (Wang et al., 2021; Iwasawa & Matsuo, 2021; Zhang et al., 2022b; Shu et al., 2022; Feng et al., 2023; Karmanov et al., 2024; Zhang et al., 2024c;b;a; Lin et al., 2023) and from the language-prior line (Menon & Vondrick, 2023; Bosetti et al., 2024; Zheng et al., 2024; Li et al., 2024a) and the time-aware video-LLM line (Ren et al., 2024; Huang et al., 2024; Yang et al., 2025b; Pramanick et al., 2025; Li et al., 2025; Yang et al., 2025a; Wang et al., 2024b;a); the closest published relative of our memory is RSKP (Huang et al., 2022), weakly supervised and trained, whose own finding that the highest-scoring snippets make poor class representatives is the measured precedent for gating admission with a text contrast rather than with relevance alone. *Ours requires neither a training set nor a gradient at test time, and the cross-video state it carries is accumulated from the test stream itself.*

E.2 Supervised temporal action localization

Table 47: Supervised TAL families, and what this paper takes from each. The anchor-free row supplies the fully-supervised ActionFormer reference row of Table 45 and the stem of the extent-supervised localizer of Appendix D.8; the event-boundary row supplies the outer–inner contrastive score the inherited ranking criterion is built on.

Family	Representative work	What this paper takes from it
Snippet backbones	Carreira & Zisserman (2017); Feichtenhofer et al. (2019)	the frozen-feature recipe
Proposal-based heads	Lin et al. (2018; 2019); Xu et al. (2020)	boundary-probability regression, Appendix D.8
Anchor-free detectors	Zhang et al. (2022a); Shi et al. (2023)	the 35.6 reference row and the head's stem
Set prediction	Liu et al. (2022); Shi et al. (2022); Cheng & Bertasius (2022); Liu et al. (2025)	–
Event boundaries	Shou et al. (2021); Kang et al. (2022); Gothe et al. (2024); Shou et al. (2018)	the outer–inner contrastive score

Temporal action localization predicts the start, end and category of every action instance in an untrimmed video, and the dominant supervised recipe extracts snippet features with a frozen 3D backbone and trains a detection head on top: proposal-based pipelines score candidate windows

through boundary probability maps, dense start–end matching or sub-graph localization, while anchor-free detectors drop the proposal stage and regress centres and offsets or relative boundaries, setting the fully-supervised reference on THUMOS14 (Idrees et al., 2017) and ActivityNet-1.3 (Caba Heilbron et al., 2015), whose ActivityNet value Table 45 quotes at 35.6 Avg_{10}. The last row is the branch our extent-supervised localizer sits closest to, since it predicts where a boundary lies without reference to the vocabulary; it differs in being supervised by seen-class extents with class labels discarded rather than by generic boundary annotations, and in being composed with a zero-shot recognizer rather than the closed-set classifier the supervised methods assume. *We borrow the frozen-feature recipe and the outer–inner contrastive score from this line and relax its shared closed-vocabulary, fully-supervised assumption.*