WHICH CHECKPOINT? ASK THE OTHERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Selecting a checkpoint for domain-adaptive object detection requires evaluating target-domain performance without target annotations. Existing label-free evaluation criteria based on confidence, box stability, and prediction consistency characterize observed detections, but do not directly account for false negatives caused by missed objects. **Trajectory-based Error Decomposition (TED)** addresses this limitation by exploiting relationships between checkpoints, using detections at one checkpoint to expose possible misses at another. Our contributions are as follows. **(1)** TED performs **trajectory-based latent error reconstruction** by linking detections into shared object candidates and using their presence, absence, and duplicate detections to reconstruct TP-, FN-, and FP-like trajectories. This label-free reconstruction brings possible misses into checkpoint evaluation. **(2)** TED turns the resulting reconstruction evidence into a selection criterion through **common-mode evidence removal** by removing candidates whose presence is invariant at the threshold used to collect them, focusing comparison on candidates whose presence differs across checkpoints at that threshold. In experiments, TED achieves the lowest selection RMSE among the evaluated methods under AP_{50}, AP_{75}, and mAP. Under AP_{50}, TED achieves a selection RMSE of 0.963 AP points, compared with 1.768 for the second-best selector.

1 INTRODUCTION

The practical performance of domain-adaptive object detection (DAOD) depends not only on the adaptation algorithm but also on the checkpoint selected for deployment. Without target annotations, this choice cannot rely directly on target-domain accuracy, and source-domain validation does not reliably establish target-domain performance (You et al., 2019). Checkpoint selection is therefore a separate deployment problem that must be addressed without target labels (Yu et al., 2024).

A central difficulty in label-free detection evaluation is that observed detections and missed objects provide unequal evidence. Existing criteria use confidence, box stability, or prediction consistency to characterize detection reliability (Yang et al., 2024b; Yu et al., 2024; Yoo et al., 2025; Manoharan et al., 2025), but these signals do not directly account for false negatives caused by missed objects. A checkpoint can produce confident, stable, and consistent predictions while failing to detect many objects. Likewise, fewer detections may indicate either fewer false positives or more misses. Criteria that summarize the reliability of observed detections therefore leave unresolved whether apparently reliable predictions also cover the objects that should have been detected.

Our **key observation** is that training produces a population of checkpoints that can itself serve as a source of label-free evaluation evidence. When evaluated on the same target images, these checkpoints need not detect the same objects. An object candidate detected by other checkpoints provides a reference for interpreting its absence at the checkpoint under evaluation as a possible miss, with repeated observations lending support to that interpretation. The checkpoint population thus exposes *reachable missed-object evidence* that is unavailable from the evaluated checkpoint's detections alone. However, objects missed by every checkpoint remain unidentifiable from predictions alone.

Trajectory-based Error Decomposition (TED) implements this observation through a shared-candidate representation followed by error decomposition. It links detections across checkpoints using predicted classes and overlapping boxes, forming *candidate trajectories* that record presence, absence, and detection multiplicity. Recurrence weights candidate support, while a separate assessment

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.

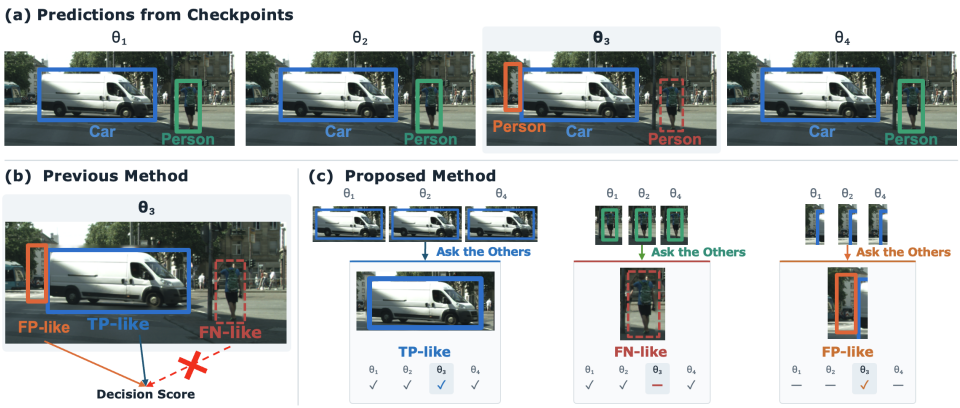

Figure 1: **Scalar scoring versus trajectory-based error evidence.** **(a)** Predictions vary across checkpoints: at θ_3, the car is detected (**blue**), an unsupported detection appears (**orange**), and the person detected by θ_1, θ_2, and θ_4 (**green**) is missed (**red dashed**). **(b)** Previous method scores θ_3 from its observed predictions, so only **TP-like** and **FP-like** evidence enters the checkpoint score while the missed person provides no evidence. **(c)** TED asks the other checkpoints to form shared candidate trajectories, interprets the absence at θ_3 as reachable **FN-like** evidence, and combines **TP-like**, **FN-like**, and **FP-like** evidence in the checkpoint score.

threshold determines presence and multiplicity at each checkpoint. Crucially, shared candidates let TED assess a checkpoint even where it produces no detection, since a candidate supported by other checkpoints but missed by this checkpoint contributes FN-like evidence of a reachable miss. Together with supported detections (TP-like) and unsupported or duplicate detections (FP-like), this FN-like term makes possible misses an explicit part of evaluation, rather than judging only the detections a checkpoint produces (Figure 1). These components are label-free proxies, not verified error counts.

However, strongly supported candidates need not help select a checkpoint. A candidate detected at every checkpoint receives maximal recurrence support but provides no difference in presence at the construction threshold. Such shared evidence is not necessarily neutral because adding identical supported-detection evidence can affect ratio-based scores unequally and even change checkpoint rankings. This motivates distinguishing **evidence of object presence from differences between checkpoints**, rather than using all reconstructed evidence indiscriminately for selection. TED therefore introduces **common-mode evidence removal** to exclude candidates whose construction-threshold presence is invariant. It then computes an F1-like score from the remaining evidence for each class and averages these scores equally. This removal targets checkpoint comparison rather than object verification, but it can also discard differences at the assessment threshold, so its benefit must be assessed empirically.

Our contributions are threefold:

- **Trajectory-based label-free error decomposition.** We link cross-checkpoint detections into shared candidate trajectories to reconstruct TP-, FN-, and FP-like evidence.
- **Separation of reconstruction and selection evidence.** We distinguish evidence useful for reconstructing latent errors from evidence useful for comparing checkpoints, removing invariant candidates before selection scoring.
- **Broad empirical validation.** TED achieves the lowest selection RMSE under AP_{50}, AP_{75}, and mAP among the evaluated methods.

2 Related Work

2.1 Unsupervised model selection under distribution shift

Unsupervised model selection estimates target performance from indirect signals. Existing methods use confidence and prediction statistics (Guillory et al., 2021; Garg et al., 2022; Deng et al., 2023;

Xie et al., 2024), distribution-shift measures (You et al., 2019; Saito et al., 2021; Yang et al., 2024a; Joo & Klabjan, 2024), and classifier agreement (Baek et al., 2022; Hu et al., 2024). Yet these checkpoint-level summaries discard the unordered spatial structure of detections. They cannot align objects across checkpoints or provide evidence when a checkpoint produces no detection. Learned accuracy predictors also require labeled synthetic meta-datasets (Deng & Zheng, 2021). TED instead constructs shared spatial candidates from saved predictions and reconstructs TP-, FN-, and FP-like evidence without target labels or an auxiliary predictor.

2.2 Label-free evaluation for detection and segmentation

BoS (Yang et al., 2024b) measures box stability under feature dropout, while DAS (Yu et al., 2024) combines parameter stability with transferability and discriminability. PCR (Yoo et al., 2025) uses consistency and reliability around NMS, CCS (Manoharan et al., 2025) compares augmented views, and CMR (Talks et al., 2026) ranks segmentation models through perturbation consistency. In addition, the original BoS and PCR pipelines fit linear score-to-mAP regressors on labeled meta-datasets, introducing auxiliary supervision and calibration data. More fundamentally, these criteria assess each checkpoint only through its own observed detections, so objects it misses leave no trace in the score. TED requires neither a performance regressor nor self-contained assessment: by linking detections across checkpoints, it turns possible misses into explicit FN-like evidence and evaluates reachable object coverage alongside detection reliability. This cross-checkpoint view connects label-free detection evaluation to training dynamics.

2.3 Training dynamics as an evaluation signal

Training dynamics reveal information that a single snapshot misses. Repeated observations capture example difficulty and ambiguity (Toneva et al., 2019; Swayamdipta et al., 2020). They also support mislabeled-data detection and label-free coreset selection (Pleiss et al., 2020; Jia et al., 2023; Zheng et al., 2025). Yet these approaches assume fixed example identities. They do not transfer directly to detection, where outputs are unordered and change across checkpoints. TED overcomes this barrier by linking detections into shared object candidates. The resulting trajectories expose reachable misses and duplicates that single-checkpoint evidence cannot reveal. This makes checkpoint dynamics directly usable for label-free detection evaluation.

3 Method

3.1 Problem formulation

Let $\mathcal{D} = \{x_i\}_{i=1}^{N}$ be an unlabeled target dataset of N images and $\Theta = \{\theta_t\}_{t=1}^{T}$ a fixed indexed collection of T detector checkpoints saved during adaptation, where θ_t denotes the t-th checkpoint. The goal is to select a checkpoint for target-domain deployment using only their predictions on the same target images. Target annotations Y_T are unavailable for scoring and hyperparameter selection. For image x_i and checkpoint θ_t, let $\mathcal{P}_t(x_i)$ denote the detection set produced by θ_t:

$$\mathcal{P}_t(x_i) = \{(b_{itk}, c_{itk}, s_{itk})\}_{k=1}^{K_{it}}, \tag{1}$$

where K_{it} is the number of detections, and b_{itk}, c_{itk}, and s_{itk} denote the predicted bounding box, class, and confidence score of detection k, respectively.

3.2 Cross-checkpoint candidate trajectories

To compare detections with possible misses, TED first forms a shared candidate reference from predictions of all checkpoints. A candidate observed at one checkpoint can then serve as a reference when it is absent at another. For image i and predicted class ℓ, let $\mathcal{V}_{i\ell}$ and $\mathcal{E}_{i\ell}$ denote the node and edge sets of the association graph:

$$\begin{aligned} \mathcal{V}_{i\ell} &= \{(t,k) \mid 1 \le t \le T,\ 1 \le k \le K_{it},\ c_{itk} = \ell,\ s_{itk} \ge \tau_{\text{cand}}\}, \\ \mathcal{E}_{i\ell} &= \{\{(t,k),(t',k')\} \subseteq \mathcal{V}_{i\ell} \mid (t,k) \ne (t',k'),\ \text{IoU}(b_{itk}, b_{it'k'}) \ge \tau_{\text{assoc}}\}. \end{aligned} \tag{2}$$

Here, the construction threshold τ_{cand} is the minimum confidence required for a detection to enter the node set, and the association threshold τ_{assoc} is the minimum box IoU required to connect two

nodes. The node set pools construction-level detections from all T checkpoints. The graph is formed over the full checkpoint collection without using temporal order or checkpoint adjacency. Each edge is an unordered pair of distinct nodes. Hence, $t = t'$ is allowed when $k \neq k'$, which permits duplicate detections from one checkpoint to enter the same component. The connected components computed by union–find define the shared candidate set $\mathcal{C} = \{C_j\}_{j=1}^{J}$, where J is the total number of candidates across target images and classes.

Graph connectivity is transitive. If one detection overlaps a second and the second overlaps a third, all three can enter one candidate even when the first and third do not meet the association threshold directly. This chaining can merge nearby objects. We retain union–find because temporal one-to-one matching reduces multi-object merging but increases fragmentation. Appendix C.7 reports this trade-off.

For a candidate C_j from image i and checkpoint t, we define the construction occurrence, assessment count, and assessment occurrence as follows:

$$z_{jt}^{\mathrm{cand}} = \mathbf{1}[\exists k \text{ with } (t,k) \in C_j]\,,$$
$$n_{jt}^{\mathrm{det}} = \sum_{k \mid (t,k)\in C_j} \mathbf{1}[s_{itk} \geq \tau_{\mathrm{det}}]\,, \qquad z_{jt}^{\mathrm{det}} = \mathbf{1}\left[n_{jt}^{\mathrm{det}} > 0\right]. \tag{3}$$

Here, the assessment threshold τ_{det} is the minimum confidence required for an assigned detection to count when evaluating a checkpoint. The variable z_{jt}^{cand} records whether candidate j occurs at checkpoint t under candidate construction. The variable z_{jt}^{det} records assessment-level presence, while n_{jt}^{det} preserves detection multiplicity. Candidate construction and checkpoint assessment use different thresholds, so $z_{jt}^{\mathrm{cand}} = 1$ does not imply $z_{jt}^{\mathrm{det}} = 1$. The case $z_{jt}^{\mathrm{cand}} = 1$ and $z_{jt}^{\mathrm{det}} = 0$ means that the checkpoint provides candidate evidence without an assessment-level detection. Such evidence supports candidate recurrence through z_{jt}^{cand}, while $z_{jt}^{\mathrm{det}} = 0$ contributes FN-like evidence in the following decomposition. With a single threshold, weak evidence would either be excluded from recurrence support or counted as a successful detection. Candidate construction can therefore be present at every checkpoint even when assessment-level presence varies.

3.3 Candidate recurrence and error decomposition

A candidate observed at many checkpoints has stronger cross-checkpoint support than one observed only rarely. We quantify this support by the recurrence of candidate j

$$f_j = \frac{1}{T}\sum_{t=1}^{T} z_{jt}^{\mathrm{cand}}. \tag{4}$$

Here, f_j is the fraction of checkpoints that provide construction-level evidence for candidate j. The post-hoc analysis in Table 3 shows that candidates with larger f_j correspond to ground-truth objects more often. We therefore use f_j directly as the candidate's support strength. It remains a relative evidence weight rather than a calibrated probability that the candidate is a real object.

The checkpoint-specific interpretation of a candidate depends jointly on this cross-checkpoint support and the checkpoint's own assessment-level detections. We define its error evidence by

$$\mathbf{e}_{jt} = \Big[\underbrace{f_j z_{jt}^{\mathrm{det}}}_{\text{TP-like}} \quad \underbrace{f_j(1 - z_{jt}^{\mathrm{det}})}_{\text{FN-like}} \quad \underbrace{(1-f_j)n_{jt}^{\mathrm{det}}}_{\text{Background FP-like}} \quad \underbrace{f_j(n_{jt}^{\mathrm{det}} - z_{jt}^{\mathrm{det}})}_{\text{Duplicate FP-like}}\Big]^{\top}. \tag{5}$$

When f_j is large, detecting the candidate produces strong TP-like evidence through the first term. Missing the same strongly supported candidate instead produces strong FN-like evidence through the second term. When f_j is small, a detection has little support from the other checkpoints, so the third term treats it primarily as background FP-like evidence. The difference $n_{jt}^{\mathrm{det}} - z_{jt}^{\mathrm{det}}$ counts detections beyond the first. The fourth term therefore assigns duplicate FP-like evidence when a strongly supported candidate receives multiple detections. Thus, the same detection is interpreted differently depending on how consistently the candidate appears across checkpoints.

Summing the candidate-level evidence gives the reconstructed error evidence for checkpoint t

$$\mathbf{E}_t^{\mathrm{rec}} = \sum_{j=1}^{J} \mathbf{e}_{jt}. \tag{6}$$

Within each class, TP-like and FN-like evidence sum to the fixed total recurrence mass, so they are complementary rather than independent. The FP-like terms assess weakly supported and duplicate detections separately. All four components are label-free proxies rather than verified error counts. They cover only candidates supported by at least one checkpoint. Objects missed by every checkpoint never enter $\mathcal{C}$ and remain unidentifiable from predictions alone. Appendix A formalizes this identifiability boundary, and Appendix C.5 reports the reachable ground-truth coverage of the checkpoint sets.

3.4 Common-mode evidence removal

The reconstruction above uses recurrence to determine how strongly each candidate contributes to error evidence. Checkpoint selection, however, requires evidence that distinguishes checkpoints. These two objectives need not favor the same candidates. Consider a candidate with $f_j = 1$. It appears above τ_{cand} at every checkpoint, giving maximal cross-checkpoint support for its existence. Yet its construction-level occurrence is identical across checkpoints and therefore provides no presence difference for distinguishing them.

Such shared evidence is not necessarily neutral in the final selection score. In the special case where the candidate also has exactly one assessment-level detection at every checkpoint, Eq. (5) adds the same TP-like evidence $[1 \quad 0 \quad 0 \quad 0]^\top$ to every checkpoint. Because the F1-like score in Eq. (8) is nonlinear, this identical addition changes checkpoint scores by different amounts and can even reverse their ranking, as summarized in Appendix A.2.

To focus scoring on candidates whose construction-level occurrence varies, TED applies the mask

$$a_j = \mathbf{1}\{f_j < 1\}, \qquad \mathbf{E}_t^{\text{cmp}} = \sum_{j=1}^{J} a_j \mathbf{e}_{jt}. \tag{7}$$

The mask retains a candidate if it is absent from at least one checkpoint at τ_{cand} and removes it if it is observed at all T checkpoints. A removed candidate is excluded only from the comparison evidence $\mathbf{E}_t^{\text{cmp}}$ used for selection; removal does not imply that the candidate is false or uninformative for reconstruction.

Importantly, invariance here is defined at the construction threshold τ_{cand}. Even when $f_j = 1$, assessment-level presence z_{jt}^{det} and multiplicity n_{jt}^{det} can vary across checkpoints because they are defined at τ_{det}. The removal rule can therefore discard some assessment-level differences. Table 4 evaluates the complete removal rule, including this trade-off, empirically.

3.5 Class-wise checkpoint scoring

After common-mode evidence removal, we convert the retained candidate evidence into a scalar score for each checkpoint. We apply the masked sum in Eq. (7) separately to the candidates of each predicted class. For class ℓ and checkpoint t, let $TP_{\ell t}$ and $FN_{\ell t}$ denote the first and second components of this class-restricted sum, and let $FP_{\ell t}$ denote the sum of its background and duplicate FP-like components. We keep the two FP-like components separate in the decomposition because they describe different failure modes, but add them for scoring because both represent excess detection evidence. These quantities remain label-free proxy masses rather than verified error counts.

Let L be the number of classes with at least one retained candidate, indexed by $\ell = 1, \ldots, L$; the same classes are used for every checkpoint. For each class and checkpoint, $P_{\ell t}$ is a precision-like score that penalizes unsupported or duplicate detections through $FP_{\ell t}$, whereas $R_{\ell t}$ is a recall-like score that penalizes missed candidates through $FN_{\ell t}$. Their harmonic mean $F_{\ell t}$ gives the class-wise checkpoint score. We then average these scores over the L retained classes to obtain the checkpoint score q_t, giving every included class equal weight rather than allowing classes with many candidates to dominate. Finally, $\hat{t}$ denotes the selected checkpoint, defined as the checkpoint with the largest q_t:

$$\begin{aligned} P_{\ell t} &= \frac{TP_{\ell t}}{TP_{\ell t} + FP_{\ell t} + \epsilon}, \quad R_{\ell t} = \frac{TP_{\ell t}}{TP_{\ell t} + FN_{\ell t} + \epsilon}, \\ F_{\ell t} &= \frac{2P_{\ell t}R_{\ell t}}{P_{\ell t} + R_{\ell t} + \epsilon}, \qquad q_t = \frac{1}{L}\sum_{\ell=1}^{L} F_{\ell t}, \qquad \hat{t} = \arg\max_t q_t. \end{aligned} \tag{8}$$

Here, $\epsilon > 0$ prevents division by zero. Appendix B.1 provides the complete selection pseudocode.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate five domain shifts: Cityscapes→Foggy Cityscapes (C2F) for weather change (Cordts et al., 2016; Sakaridis et al., 2018), Sim10k→Cityscapes (S2C) for synthetic-to-real adaptation (Johnson-Roberson et al., 2017; Cordts et al., 2016), KITTI→Cityscapes (K2C) for cross-dataset driving-domain adaptation (Geiger et al., 2012; Cordts et al., 2016), PASCAL VOC→Clipart1k (V2C) for real-to-art adaptation (Everingham et al., 2010; Inoue et al., 2018), and xView→DOTA (X2D) for cross-dataset aerial adaptation (Lam et al., 2018; Xia et al., 2018). Source annotations are used only for adaptation, and target annotations remain hidden until checkpoint selection is complete.

Main evaluation and checkpoint sets. All detector training, checkpoint generation, and selector evaluation use a unified implementation based on the Align and Distill (ALDI), ALDI++ (Kay et al., 2025), and DAS (Yu et al., 2024) codebases. The main evaluation combines the five shifts with Mean Teacher (MT) (Tarvainen & Valpola, 2017), Adaptive Teacher (AT) (Li et al., 2022), ALDI++ (Kay et al., 2025), and Contrastive Mean Teacher (CMT) (Cao et al., 2023) using Faster R-CNN (Ren et al., 2015), yielding 20 setups. We additionally evaluate MT and AT with Deformable-DETR (Zhu et al., 2021) and YOLOv5 (Jocher, 2022) across the five shifts, yielding 20 additional setups and 40 distinct setups in total. Figure 2 reports the fully matched baseline comparison on the 20 Faster R-CNN setups. Appendix Table 6 reports the 20 additional Deformable-DETR and YOLOv5 setups, while the remaining main mechanism and ablation analyses use all 40 setups. Within each setup, every selector evaluates the same target images and prespecified checkpoint set.

Implementation details. TED uses $\tau_{\text{assoc}} = 0.40$, $\tau_{\text{cand}} = 0.05$, and $\tau_{\text{det}} = 0.50$. Inspired by the default Faster R-CNN inference settings, we fixed this configuration before accessing target annotations and applied it unchanged across all shifts, adaptation methods, detectors, and evaluation metrics. Section 4.4 evaluates its sensitivity, and Appendix B.2 provides further implementation details.

Baselines and applicability. We compare TED with 16 label-free baselines and two temporal references. We reproduced them from the corresponding repositories and integrated them into our workspace. Nine Classification-derived baselines use confidence or prediction statistics: AC and PS (Hendrycks & Gimpel, 2017), ES (Saito et al., 2019), DIV and IM (Liang et al., 2020), BNM (Cui et al., 2020), ATC (Garg et al., 2022), SND (Saito et al., 2021), and MaNo (Xie et al., 2024). Seven detection-oriented baselines use spatial or detector-specific evidence: CCS-2View (Manoharan et al., 2025), CMR-Det-Hard and CMR-Det-Soft (Talks et al., 2026), BoS (Yang et al., 2024b), DAS (Yu et al., 2024), and PCR-C and PCR-R (Yoo et al., 2025). CMR-Det-Hard/Soft are our detection adaptations of Consistency Model Ranking (CMR), and PCR-C/R denote the consistency and reliability components of PCR, respectively. Selecting the first or last saved checkpoint provides two temporal references that require no prediction-based criterion.

BoS, PCR, and DAS respectively require dropout, pre-NMS, and training-side or source-domain signals unavailable through the common detector interface. We therefore evaluate all 16 baselines and both temporal references on the 20 Faster R-CNN setups using identical target images and checkpoint sets. On the 20 additional Deformable-DETR and YOLOv5 setups, we evaluate the 12 compatible baselines and both temporal references, excluding BoS, DAS, PCR-C, and PCR-R. Analyses over all 40 setups are restricted to TED variants and do not claim a fully matched external comparison. Following DAS, we use direct label-free ranking and the native fit-free BoS and PCR statistics.

Evaluation protocol. For each setup s, the label-free score selects a checkpoint $\hat{t}_s$ before ground truth is accessed, and this choice remains fixed during evaluation. For each average precision (AP) metric $m \in \{AP_{50}, AP_{75}, \text{mAP}\}$, let $AP^{\star}_{s,m}$ be the best performance among the checkpoints in setup s. Across S setups, we report the root mean squared error relative to these per-setup oracles, in

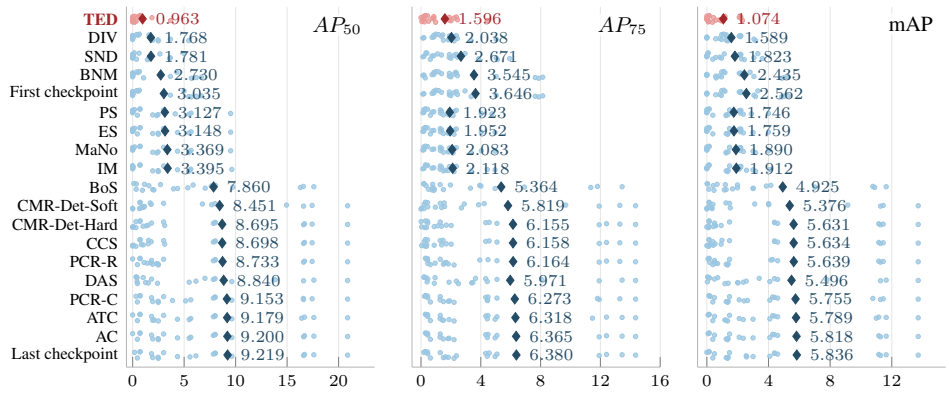

Figure 2: **Label-free checkpoint selection on twenty Faster R-CNN setups.** The horizontal axes report oracle gaps in AP points (lower is better), and the vertical axis lists the checkpoint selectors. From left to right, the panels evaluate AP_{50}, AP_{75}, and mAP. Each small circle represents one setup, and the larger diamond marks the oracle-relative RMSE across the twenty setups.

AP points:

$$\mathrm{RMSE}_m = \sqrt{\frac{1}{S}\sum_{s=1}^{S}\left[100\left(AP^{\star}_{s,m} - AP_{s,m}(\hat{t}_s)\right)\right]^2}. \tag{9}$$

RMSE emphasizes large selection errors and equals zero when the selector recovers an oracle checkpoint for every setup. The same $\hat{t}_s$ is evaluated under all three metrics without metric-specific reselection. Here, S denotes the number of setups in the corresponding comparison: $S = 20$ for Figure 2 and Appendix Table 6, and $S = 40$ for the mechanism and ablation analyses over the union of the two grids.

4.2 Main results: label-free checkpoint selection

Main comparison. As shown in Figure 2, TED achieves the lowest RMSE under AP_{50}, AP_{75}, and mAP: 0.963, 1.596, and 1.074, compared with 1.768, 1.923, and 1.589 for the respective strongest competitors. DAS is the most direct DAOD-specific comparator, while BoS and PCR use their native fit-free statistics. These detection-oriented competitors exhibit maximum AP_{50} gaps of 17.597–20.877 points, versus 3.677 for TED. A plausible explanation is that pseudo-label feedback can keep a checkpoint's observed predictions confident and internally consistent even as its object coverage deteriorates, a failure that criteria based only on observed detections cannot register. TED also uses recurrence as positive support, but does not treat it as sufficient evidence of checkpoint quality. Cross-checkpoint absence contributes FN-like evidence, unsupported or duplicate detections contribute FP-like evidence, and invariant candidates are removed before selection. This broader comparison may explain the smaller tail errors of TED, although errors shared by every checkpoint remain unobservable. All three metrics evaluate the same checkpoint choice per setup. Supplementary Figures 5 and 6 show the target AP_{50} trajectories and fixed selections for all 40 setups.

Generalization across detector families. The main comparison fixes Faster R-CNN so that every baseline can be evaluated on the same setups. To test whether the result depends on this detector, we additionally evaluate MT and AT checkpoints from YOLOv5 and Deformable-DETR. Appendix B.4 reports the 20 additional setups, ten per detector family, and shows that TED's selection performance generalizes beyond Faster R-CNN.

4.3 Trajectory reconstruction and checkpoint comparison

Selection score. Table 1 compares alternative checkpoint scores while holding the shared candidate set, recurrence weights, and removal mask fixed. Precision-like scoring uses only $\widehat{P}$ and excludes FN-like evidence, while recall-like scoring uses $\widehat{R}$ and accounts for missed candidates. F1-like

Score	$\mathrm{RMSE}_{AP50} \downarrow$	$\mathrm{RMSE}_{AP75} \downarrow$	$\mathrm{RMSE}_{mAP} \downarrow$
F1-like (TED)	0.743	1.391	0.881
Precision-like $\widehat{P}$	6.674	3.923	3.775
Recall-like $\widehat{R}$	1.258	2.184	1.391
AP-like	9.185	6.372	5.871

Table 1: **Selection-score ablation.** RMSE on 40 distinct setups. All scores use the same candidate set, recurrence weights, and mask $a_j = \mathbf{1}\{f_j < 1\}$.

Component	$\mathbf{E}^{\mathrm{rec}}$, $\rho \uparrow$	$\mathbf{E}^{\mathrm{cmp}}$, $\rho \uparrow$
TP-like	0.767	0.530
FN-like	0.767	0.529
FP-like	0.789	0.525

Table 2: **Reachable trajectory agreement.** Mean setup-wise Spearman correlations with GT error trajectories across 40 distinct setups.

scoring takes their harmonic mean, whereas AP-like scoring integrates a confidence-ranked proxy precision–recall curve. Recall-like scoring has substantially lower RMSE_{AP50} than precision-like scoring (1.258 versus 6.674), indicating that explicitly accounting for reachable FN-like evidence supports more reliable checkpoint selection than scoring observed detections alone. F1-like scoring further lowers the RMSE to 0.743 by balancing missed candidates against unsupported and duplicate detections. The AP-like variant has the highest RMSE_{AP50} of 9.185, supporting the fixed-threshold F1-like scoring for the present proxy construction. Confidence-ranked aggregation may be more sensitive to proxy-assignment errors across score cutoffs.

Reconstruction agreement and selection utility. Table 2 reports agreement before common-mode removal as $\mathbf{E}^{\mathrm{rec}}$ and after removal as $\mathbf{E}^{\mathrm{cmp}}$, while Figure 3 visualizes the before-removal evidence. The before-removal proxies follow the relative variation of the GT error trajectories across TP-, FN-, and FP-like components, indicating that cross-checkpoint candidates provide meaningful evidence for reconstructing latent errors. Table 4, however, shows that retaining all of this evidence does not produce effective checkpoint selection. For candidates with $f_j = 1$, construction-level presence is shared across checkpoints, while their evidence in Eq. (5) can still vary because detections may cross τ_{det}. Moreover, identical shared TP-like mass can move nonlinear F1-like scores by different amounts across checkpoints (Appendix A.2). These observations motivate evaluating a comparison score that removes candidates with $f_j = 1$. Table 4 measures the net effect on the observed trajectories, including the assessment-level differences discarded by removal. Empirically, common-mode removal lowers reconstruction agreement while improving selection. Together, these results suggest that object-presence evidence can support error reconstruction while contributing less reliably to checkpoint ranking. On these trajectories, restricting the comparison score to candidates with varying construction-level presence improves selection despite reducing reconstruction agreement. Supplementary Figures 7–10 show the corresponding GT and reconstructed TP-, FN-, and FP-like trajectories across all 40 setups.

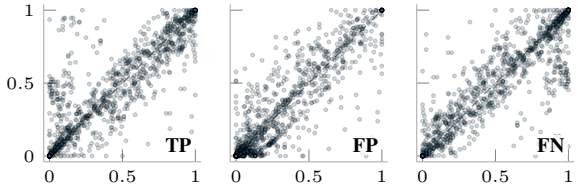

Figure 3: **Before-removal reconstruction versus GT.** Each panel plots normalized GT evidence on the x-axis and normalized reconstructed evidence on the y-axis.

Recurrence as object-presence evidence. A candidate observed at another checkpoint is not necessarily a real object. Across 40 setups, the setup-averaged fraction of candidates matched to GT increases monotonically from 2.8% for $f_j < 0.1$ to 32.2% for $0.9 \leq f_j < 1$ and 64.6% for $f_j = 1$ (Table 3). Recurrence therefore provides a useful monotonic weight for object-presence support, although even highly recurrent candidates are far from reliably real, which is why f_j serves as a relative weight rather than a probability.

Common-mode evidence removal and shared contributions. Tables 3 and 4 reveal an apparent tension. Candidates with $f_j = 1$ have the highest GT-matching rate, yet removing exactly these candidates gives the best selection result among the evaluated rules. This distinction helps explain that the higher GT-matching rate of $f_j = 1$ candidates makes them stronger evidence of object presence, but their construction-level presence does not distinguish checkpoints. Removing them

f_j	$[0, .1)$	$[.1, .2)$	$[.2, .3)$	$[.3, .4)$	$[.4, .5)$	$[.5, .6)$
GT fraction (%)	2.8	5.2	9.9	14.9	19.6	23.4

f_j	$[.6, .7)$	$[.7, .8)$	$[.8, .9)$	$[.9, 1)$	$\{1\}$
GT fraction (%)	25.4	28.9	30.5	32.2	64.6

Table 3: **GT correspondence across recurrence bins.** Setup averages over 40 distinct setups.

Rule	$\mathrm{RMSE}_{AP50} \downarrow$	$\mathrm{RMSE}_{AP75} \downarrow$	$\mathrm{RMSE}_{mAP} \downarrow$
No removal	6.105	4.278	3.891
Random (count-matched)	6.105	4.317	3.918
Random (recurrence-mass-matched)	5.961	4.180	3.816
Entropy soft rule	1.246	1.711	1.245
Variance soft rule	1.045	1.592	1.106
Remove $f \geq 0.90$	1.072	1.568	1.106
Remove $f = 1$ (TED)	0.743	1.391	0.881

Table 4: **Naturally occurring invariant-candidate controls.** RMSE on 40 distinct setups. Random-removal entries average 100 independent removal seeds per setup.

Table 5: **Threshold sensitivity.** RMSE on 40 distinct setups. Stars mark the fixed main configuration.

(a) Association threshold τ_{assoc} with $(\tau_{\mathrm{cand}}, \tau_{\mathrm{det}}) = (0.05, 0.50)$

τ_{assoc}	$\mathrm{RMSE}_{AP50} \downarrow$	$\mathrm{RMSE}_{AP75} \downarrow$	$\mathrm{RMSE}_{mAP} \downarrow$
0.30	0.763	1.393	0.882
0.40*	0.743	1.391	0.881
0.50	0.984	1.504	0.969
0.60	1.015	1.616	1.124
0.70	0.988	1.580	1.054

(b) Confidence thresholds with $\tau_{\mathrm{assoc}} = 0.40$

τ_{cand}	τ_{det}	$\mathrm{RMSE}_{AP50} \downarrow$	$\mathrm{RMSE}_{AP75} \downarrow$	$\mathrm{RMSE}_{mAP} \downarrow$
0.05	0.30	0.829	1.516	1.070
0.05	0.40	1.356	1.541	1.214
0.05	0.50*	0.743	1.391	0.881
0.10	0.30	1.035	1.164	0.973
0.10	0.40	0.532	1.159	0.884
0.10	0.50	0.707	1.316	0.934

from the comparison score does not treat them as false objects. Table 4 shows that, on the evaluated trajectories, the selection benefit of this choice outweighs the assessment-level differences it discards. The other rules in Table 4 are consistent with this view. The near-invariant cutoff $f_j \geq 0.90$ and the entropy and variance rules also remove or down-weight candidates whose presence varies, and matched random removal discards such candidates at random. All of these alternatives yield higher RMSE_{AP50} than exact removal (1.045–6.105 versus 0.743). Appendix C.2 defines these controls, and Appendix C.4 provides a controlled illustration of shared-evidence sensitivity.

4.4 Parameter sensitivity and robustness

Table 5 varies each threshold around the fixed main configuration: panel (a) varies τ_{assoc} with $(\tau_{\mathrm{cand}}, \tau_{\mathrm{det}}) = (0.05, 0.50)$, and panel (b) varies the confidence thresholds with $\tau_{\mathrm{assoc}} = 0.40$. Across all tested settings, RMSE_{AP50} ranges from 0.532 to 1.356 and remains below that of the second best. The fixed configuration is not optimal in every case. For example, $(\tau_{\mathrm{cand}}, \tau_{\mathrm{det}}) = (0.10, 0.40)$ yields lower RMSE_{AP50} and RMSE_{AP75}. We nevertheless report the configuration fixed before target annotations were accessed, since selecting thresholds post hoc from these results would require target labels. Additionally, the association-rule comparison is deferred to Appendix C.7.

5 Discussion and Future Work

TED currently selects offline from a fixed checkpoint trajectory. Extending its trajectory construction and scoring to an evolving run could support label-free online selection and early stopping. A second direction is cross-architecture model selection. Our evaluation compares checkpoints within a detector and does not establish score comparability across architectures, which would require accounting for detector-specific confidence scales and prediction structures.

6 Conclusion

Detections from multiple checkpoints provide evidence for candidates missed by an individual checkpoint. TED links these detections into shared candidates and represents supported detections, missed candidates, and unsupported or duplicate detections as separate components. For selection, it distinguishes evidence of object presence from differences between checkpoints, excludes candidates observed at every checkpoint under the construction threshold, and averages the resulting scores across object classes. With the candidate set and scoring rule fixed, common-mode evidence removal reduces overall selection error despite weakening agreement with ground-truth error trajectories. Error reconstruction and checkpoint selection therefore require separate evaluation.

Reproducibility Statement

Section 3 specifies the scoring method, while Appendix A clarifies its observable scope and scoring intuition. Appendix B.1 provides the complete pseudocode of selection procedure. Section 4 and Appendix B describe the datasets, checkpoint sets, baseline implementations, hyperparameters, and evaluation protocol. Appendix C provides additional ablations and robustness analyses.

AI Use Statement

AI tools were used to assist with manuscript drafting and revision, portions of the experimental code, and the preparation of figures and tables. Manuscript assistance included improving language, organizing the presentation, and clarifying notation and technical explanations. The authors take responsibility for the final manuscript, code, figures, tables, and reported results.

References

Christina Baek, Yiding Jiang, Aditi Raghunathan, and J. Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Shengcao Cao, Dhiraj Joshi, Liang-Yan Gui, and Yu-Xiong Wang. Contrastive mean teacher for domain adaptive object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23839–23848, 2023.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3941–3950, 2020.

Weijian Deng and Liang Zheng. Are labels always necessary for classifier accuracy evaluation? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15069–15078, 2021.

Weijian Deng, Yumin Suh, Stephen Gould, and Liang Zheng. Confidence and dispersity speak: Characterizing prediction matrix for unsupervised accuracy estimation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7658–7674. PMLR, 2023.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2): 303–338, 2010.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *International Conference on Learning Representations*, 2022.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1134–1144, 2021.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017.

Dapeng Hu, Mi Luo, Jian Liang, and Chuan-Sheng Foo. Towards reliable model selection for unsupervised domain adaptation: An empirical study and a certified baseline. In *Advances in Neural Information Processing Systems*, volume 37, 2024. Datasets and Benchmarks Track.

Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5001–5009, 2018.

Qingrui Jia, Xuhong Li, Lei Yu, Jiang Bian, Penghao Zhao, Shupeng Li, Haoyi Xiong, and Dejing Dou. Learning from training dynamics: Identifying mislabeled data beyond manually designed features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8041–8049, 2023.

Glenn Jocher. YOLOv5 by ultralytics. Software, 2022. URL `https://github.com/ultralytics/yolov5`.

Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 746–753, 2017.

Taejong Joo and Diego Klabjan. IW-GAE: Importance weighted group accuracy estimation for improved calibration and model selection in unsupervised domain adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 22509–22529. PMLR, 2024.

Justin Kay, Timm Haucke, Suzanne Stathatos, Siqi Deng, Erik Young, Pietro Perona, Sara Beery, and Grant Van Horn. Align and distill: Unifying and improving domain adaptive object detection. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL `https://openreview.net/forum?id=ssXSrZ94sR`.

Darius Lam, Richard Kuzma, Kevin McGee, Samuel Dooley, Michael Laielli, Matthew Klaric, Yaroslav Bulatov, and Brendan McCord. xview: Objects in context in overhead imagery, 2018.

Yu-Jhe Li, Xiaoliang Dai, Chih-Yao Ma, Yen-Cheng Liu, Kan Chen, Bichen Wu, Zijian He, Kris Kitani, and Peter Vajda. Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7581–7590, 2022.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6028–6039. PMLR, 2020.

Avinaash Manoharan, Xiangyu Yin, Domenik Helm, and Chih-Hong Cheng. Cumulative consensus score: Label-free and model-agnostic evaluation of object detectors in deployment. *arXiv preprint arXiv:2509.12871*, 2025.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q. Weinberger. Identifying mislabeled data using the area under the margin ranking. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL `https://proceedings.neurips.cc/paper_files/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html`.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8050–8058, 2019.

Kuniaki Saito, Donghyun Kim, Piotr Teterwak, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Tune it the right way: Unsupervised validation of domain adaptation via soft neighborhood density. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9184–9193, 2021.

Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018. doi: 10.1007/s11263-018-1072-8.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9275–9293. Association for Computational Linguistics, 2020.

Joshua Talks, Kevin Marchesini, Luca Lumetti, Federico Bolelli, and Anna Kreshuk. Unsupervised source-free ranking of biomedical segmentation models under distribution shift. In *European Conference on Computer Vision*, 2026. Accepted manuscript.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019.

Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3974–3983, 2018.

Renchunzi Xie, Ambroise Odonnat, Vasilii Feofanov, Weijian Deng, Jianfeng Zhang, and Bo An. Mano: Exploiting matrix norm for unsupervised accuracy estimation under distribution shifts. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Jianfei Yang, Hanjie Qian, Yuecong Xu, Kai Wang, and Lihua Xie. Can we evaluate domain adaptation models without target-domain labels? In *International Conference on Learning Representations*, 2024a.

Yang Yang, Wenhai Wang, Zhe Chen, Jifeng Dai, and Liang Zheng. Bounding box stability against feature dropout reflects detector generalization across environments. In *International Conference on Learning Representations*, 2024b.

Seungju Yoo, Hyuk Kwon, Joong-Won Hwang, and Kibok Lee. Automated model evaluation for object detection via prediction consistency and reliability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19764–19773, 2025.

Kaichao You, Ximei Wang, Mingsheng Long, and Michael I. Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7124–7133. PMLR, 2019.

Hengfu Yu, Jinhong Deng, Wen Li, and Lixin Duan. Towards unsupervised model selection for domain adaptive object detection. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Haizhong Zheng, Elisa Tsai, Yifu Lu, Jiachen Sun, Brian Bartoldson, Bhavya Kailkhura, and Atul Prakash. Elfs: Label-free coreset selection with proxy training dynamics. In *International Conference on Learning Representations*, 2025.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=gZ9hCDWe6ke`.

A Scope and Scoring Intuition

We clarify what prediction-only evidence can reveal and summarize why evidence shared by every checkpoint is not necessarily neutral in the nonlinear selection score. These observations motivate the empirical controls in Tables 4 and 8.

A.1 Observable scope of prediction-only error evidence

Predictions alone cannot determine full checkpoint-wise FN counts. Decomposing the misses at checkpoint t gives

$$\mathrm{FN}_t^{\mathrm{full}} = \mathrm{FN}_t^{\mathrm{reach}} + N_{\text{miss-all}}, \tag{10}$$

where $N_{\text{miss-all}}$ counts objects missed by every checkpoint and is therefore unobservable from predictions alone. For example, adding to the annotations an object whose same-class overlap with every prediction is below the evaluation threshold leaves every checkpoint's predictions unchanged but increases every checkpoint's FN count by one. Such an object never enters the shared candidate set. TED therefore addresses only *reachable* misses—objects exposed by at least one checkpoint—and does not claim to recover full FN counts or AP rankings from predictions alone. Table 10 reports the observable GT coverage of the evaluated checkpoint populations, and Table 1 evaluates whether the resulting FN-like evidence helps selection.

A.2 Sensitivity of F1-like rankings to shared evidence

For one class, let x_t denote checkpoint t's TP-like mass and let $r_t > 0$ denote its combined FN- and FP-like mass, omitting the numerical stabilizers in Eq. (8). Adding the same TP-like mass $c \geq 0$ to every checkpoint gives $\widetilde{F}_t(c) = 2(x_t + c)/(2(x_t + c) + r_t)$ and changes the score by

$$\Delta\widetilde{F}_t(c) = \widetilde{F}_t(c) - \widetilde{F}_t(0) = \frac{2cr_t}{(2x_t + r_t)(2x_t + r_t + 2c)}. \tag{11}$$

The same c therefore shifts checkpoints by different amounts according to their existing composition (x_t, r_t) and can change their ranking. A candidate with $f_j = 1$ and exactly one assessment-level detection at every checkpoint realizes this common TP-like addition. Candidates with $f_j = 1$ need not satisfy that special case, because their multiplicity and presence at τ_{det} can still vary; the naturally occurring removal controls in Table 4 measure the net effect without imposing this simplification.

A.3 Why invariant candidates can reflect confidence calibration

The two thresholds play different roles. The lower threshold $\tau_{\mathrm{cand}} = 0.05$ determines whether a detection supports a shared candidate, whereas the higher threshold $\tau_{\mathrm{det}} = 0.50$ determines whether that candidate counts as present when TP-, FN-, and FP-like evidence is computed. Consider one candidate detected with confidence 0.60 at checkpoint A and 0.40 at checkpoint B. It is present at the construction threshold in both checkpoints,

$$z_{jA}^{\mathrm{cand}} = z_{jB}^{\mathrm{cand}} = 1, \qquad f_j = 1, \tag{12}$$

but it crosses the assessment threshold only at checkpoint A:

$$z_{jA}^{\mathrm{det}} = 1, \qquad z_{jB}^{\mathrm{det}} = 0. \tag{13}$$

The candidate therefore contributes TP-like evidence at A and FN-like evidence at B, even though both checkpoints detected it at the construction threshold.

Now suppose checkpoint B's confidences are recalibrated without changing their ordering, moving this detection from 0.40 to 0.55. Its detection set and confidence ranking can remain unchanged, so its AP can remain unchanged, but z_{jB}^{det} changes from 0 to 1. The apparent difference between checkpoints A and B then disappears. Thus, for a candidate with $f_j = 1$, assessment-level variation can reflect confidence calibration rather than a difference in which object is detected. A candidate with $f_j < 1$ provides a construction-threshold presence difference across checkpoints, but this difference can also be affected by confidence calibration when a score crosses τ_{cand}. Retained candidates therefore provide comparative prediction evidence rather than verified changes in object detection.

This observation motivates removing invariant candidates from the comparison score. Retained candidates can also cross τ_{det}, and removal discards some assessment-level differences. We therefore evaluate the net effect directly. Table 4 compares exact invariant removal with no removal, matched random removal, soft weighting, and a near-invariant cutoff, while Table 8 compares the same four scoring functions before and after removal.

B Experimental Protocol and Complete Results

We document the implementation, matched comparison protocol, and detector-wise selection errors. These results complement the aggregate RMSE comparison in Figure 2.

B.1 Complete selection pseudocode

Pseudocode 1 specifies the complete label-free selection procedure in implementation order. Detector outputs are cached post-NMS predictions, and no target annotation is accessed. Exact score ties are resolved in favor of the earliest checkpoint.

Pseudocode 1: Code-style pseudocode for TED checkpoint selection.

```
INPUT:
  checkpoints Theta = {theta[1], ..., theta[T]}
  unlabeled target dataset D = {x[1], ..., x[N]}

CONSTANTS:
  tau_cand  = 0.05        # candidate construction threshold
  tau_det   = 0.50        # assessment threshold
  tau_assoc = 0.40        # association IoU threshold
  epsilon   = 1e-9

# 1. Construct the shared candidate set
for t in 1..T:
  for i in 1..N:
    calP[t,i] = post_nms_detect(theta[t], x[i])
    K[i,t] = len(calP[t,i])

calC = []
for i in 1..N:
  for ell in predicted_classes(calP[1..T,i]):
    calV[i,ell] = {(t,k) for t in 1..T, k in 1..K[i,t] if
                   c[i,t,k] == ell and s[i,t,k] >= tau_cand}
    calE[i,ell] = {{(t,k),(t_prime,k_prime)} in calV[i,ell]
                   if (t,k) != (t_prime,k_prime) and
                   IoU(b[i,t,k], b[i,t_prime,k_prime]) >= tau_assoc}
                   # t may equal t_prime when k != k_prime

    for C_j in union_find(calV[i,ell], calE[i,ell]):
      j = calC.append(C_j)
      ell[j] = ell
      for t in 1..T:
        z_cand[j,t] = int(exists k with (t,k) in C_j)
        n_det[j,t] = sum(int(s[i,t,k] >= tau_det)
                         for k with (t,k) in C_j)
        z_det[j,t] = int(n_det[j,t] > 0)

J = len(calC)

# 2. Compute recurrence and the common-mode removal mask
for j in 1..J:
  f[j] = mean(z_cand[j,1..T])
         # relative evidence weight, not a calibrated probability
  a[j] = int(f[j] < 1)

# 3. Compute retained class-wise proxy masses
reindex classes with at least one retained candidate as ell = 1..L
for ell in 1..L:
  for t in 1..T:
    TP[ell,t] = sum(a[j] * f[j] * z_det[j,t]
                    for j in 1..J if ell[j] == ell)
    FN[ell,t] = sum(a[j] * f[j] * (1 - z_det[j,t])
                    for j in 1..J if ell[j] == ell)
    FP[ell,t] = sum(a[j] * ((1 - f[j]) * n_det[j,t] +
                    f[j] * (n_det[j,t] - z_det[j,t]))
                    for j in 1..J if ell[j] == ell)

```

```
# 4. Compute class-wise F1-like scores and the checkpoint score
for t in 1..T:
  for ell in 1..L:
    P[ell,t] = TP[ell,t] / (TP[ell,t] + FP[ell,t] + epsilon)
    R[ell,t] = TP[ell,t] / (TP[ell,t] + FN[ell,t] + epsilon)
    F[ell,t] = 2 * P[ell,t] * R[ell,t] /
               (P[ell,t] + R[ell,t] + epsilon)

  q[t] = mean(F[ell,t] for ell in 1..L)

OUTPUT:
  hat_t = earliest_argmax(q[1..T])
```

B.2 Implementation details

Threshold selection. The main configuration was fixed before target annotations were accessed, with the Faster R-CNN pipeline providing the reference operating convention. Its default test-time score threshold is 0.05, so the construction threshold $\tau_{\text{cand}} = 0.05$ preserves the post-NMS detections supplied to AP evaluation. The assessment threshold $\tau_{\text{det}} = 0.50$ is a conventional deployment operating point for confident detections. It defines a high-confidence presence test, allowing lower-confidence detections to support recurrence without automatically counting as successful detections. Faster R-CNN uses an NMS IoU threshold of 0.50. The association threshold $\tau_{\text{assoc}} = 0.40$ is slightly lower so that detections of the same object remain connected under moderate checkpoint-to-checkpoint localization drift while still requiring substantial spatial overlap. These three values are transferred unchanged to Faster R-CNN, YOLOv5, and Deformable-DETR and are not selected separately by shift, adaptation method, detector, or evaluation metric. Section 4.4 evaluates nearby alternatives after the main configuration has been fixed.

Checkpoint generation and inference weights. We generated the training checkpoints using the ALDI and ALDI++ implementations (Kay et al., 2025). Before domain adaptation, each detector underwent a source-only burn-in using labeled source data only. To normalize training across datasets of different sizes, we used at least 10 source epochs for burn-in and at least 30 target epochs for domain adaptation. Let N_{src} and N_{tgt} be the numbers of source and target training images, B_{burn} the number of source images processed per burn-in iteration, and B_{tgt} the number of target images processed per DAOD iteration. The burn-in budget was

$$I_{\text{burn}} = \left\lceil \frac{10N_{\text{src}}}{B_{\text{burn}}} \right\rceil, \qquad \frac{I_{\text{burn}} B_{\text{burn}}}{N_{\text{src}}} \geq 10, \tag{14}$$

and the DAOD budget was

$$I_{\text{DAOD}} = \left\lceil \frac{30N_{\text{tgt}}}{B_{\text{tgt}}} \right\rceil, \qquad \frac{I_{\text{DAOD}} B_{\text{tgt}}}{N_{\text{tgt}}} \geq 30. \tag{15}$$

All models were trained on four NVIDIA RTX A6000 GPUs with an aggregate global batch size of 48. The source-only burn-in uses $B_{\text{burn}} = 48$. During DAOD training, each iteration contains a balanced source–target batch, so 24 target images contribute to the target-epoch count and $B_{\text{tgt}} = 24$. The source-trained weights initialized the corresponding DAOD run, but the source-only initialization itself was not included among the evaluated checkpoints. We saved one checkpoint at the end of each target epoch, yielding 30 checkpoints per setup. Equivalently, checkpoint k was saved at iteration $\lceil kN_{\text{tgt}}/B_{\text{tgt}} \rceil$ for $k = 1, \ldots, 30$, counted from the start of target-domain adaptation. These positions were fixed before selector evaluation, and all selectors receive the same ordered checkpoint set within a setup. For teacher–student adaptation recipes, checkpoint predictions are generated from the teacher weights used by the native inference procedure. The remaining architecture, optimization, augmentation, and teacher-update settings followed ALDI and ALDI++.

Prediction caching. Each checkpoint is evaluated once on the fixed unlabeled target split. For every target image, we cache the post-NMS bounding boxes, predicted classes, and confidence scores used by the common detector interface. Candidate construction, association, recurrence estimation, and checkpoint scoring operate only on these cached predictions. Target annotations are neither loaded during caching nor exposed to the selector. They are accessed only after a checkpoint has been selected to compute AP and the post-hoc diagnostics reported in the paper.

B.3 Benchmark and comparison protocol

The benchmark consists of 20 Faster R-CNN setups, formed from five domain shifts and four adaptation methods (MT, AT, ALDI++, and CMT), and 20 additional setups formed from the same five shifts, MT and AT, and the two additional detector families (YOLOv5 and Deformable-DETR). This gives 40 distinct training runs in total. Each setup provides the prespecified set of 30 checkpoints described above. A comparison includes only methods available on every setup in the corresponding evaluation block, with identical target images and checkpoint sets supplied to all included methods.

B.4 Selection errors by detector family

Table 6 evaluates 20 additional YOLOv5 and Deformable-DETR setups. TED retains the lowest RMSE under all three metrics, with an AP_{50} error of 0.419 compared with 3.623 for the strongest compatible baseline. The margin persists under AP_{75} and mAP, indicating that cross-checkpoint presence, absence, and multiplicity evidence transfers across detector families with substantially different output structures. The table includes the selectors supported by the common detector interface, while the detector-specific Faster R-CNN comparison is reported in Figure 2.

Table 6: **Label-free checkpoint selection on YOLOv5 and Deformable-DETR.** On the 20 additional setups, TED remains the lowest-error selector under AP_{50}, AP_{75}, and mAP, showing that its advantage is not confined to Faster R-CNN. Only selectors available through every evaluated detector interface are included.

Selector	$\mathrm{RMSE}_{AP50} \downarrow$	$\mathrm{RMSE}_{AP75} \downarrow$	$\mathrm{RMSE}_{mAP} \downarrow$
BNM	12.606	8.697	8.017
CMR-Det-Hard	8.517	4.332	4.742
CCS	8.101	4.236	4.588
AC	7.842	4.240	4.474
ATC	7.821	4.218	4.461
CMR-Det-Soft	7.814	3.954	4.350
ES	6.469	3.633	3.630
PS	6.450	3.628	3.621
IM	5.423	3.101	2.940
MaNo	4.966	2.858	2.879
DIV	4.322	2.096	2.100
SND	3.623	2.733	2.520
TED	**0.419**	**1.150**	**0.632**

C Mechanism, Ablations, and Robustness

The supplementary analyses characterize the setup-level effect of common-mode removal and its interaction with alternative weighting rules and scoring functions. We also evaluate observable coverage, checkpoint budget, candidate association, oracle location, and trajectory direction.

C.1 Setup-level removal outcomes

Figure 4 shows that the aggregate gain from common-mode evidence removal is driven primarily by avoiding large errors rather than by uniformly improving every setup. Removal improves 20 setups, leaves 13 tied, and worsens 7. More importantly, several setups with large pre-removal gaps move close to the oracle after removal, while most regressions remain comparatively small. The many ties indicate that invariant candidates are often not decisive, but the asymmetric tail behavior shows why retaining them can still be costly: shared evidence can overturn the ranking in a small number of consequential cases.

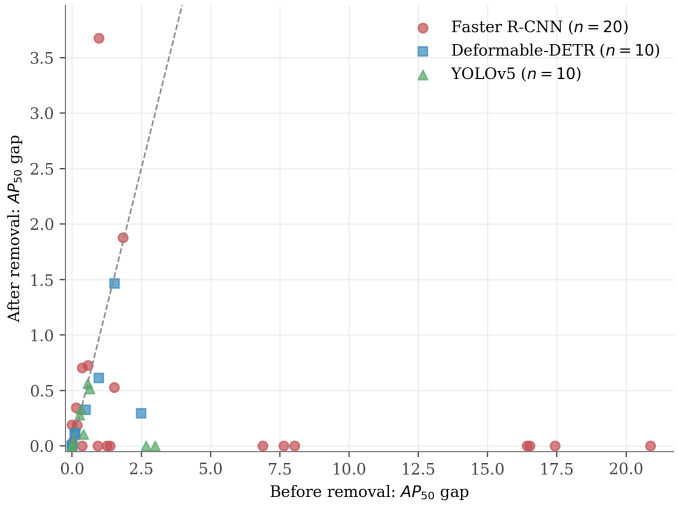

Figure 4: **Per-setup effect of common-mode evidence removal.** Each point is one setup; the horizontal and vertical axes give the AP_{50} oracle gap before and after removal, respectively. Points below the dashed diagonal improve. The largest changes are reductions of high pre-removal gaps, showing that removal mainly suppresses tail failures rather than producing a uniform shift.

C.2 Removal and weighting variants

Control definitions. Table 7 organizes the controls by the hypothesis they test. Random removal asks whether the gain comes from discarding an equal amount of evidence irrespective of identity; the entropy and variance rules ask whether smoothly weighting candidates by presence variation is sufficient; and the near-invariant cutoff asks whether removal should extend beyond exactly shared candidates. Exact removal assigns zero weight to invariant candidates and unit weight to every varying candidate. The entropy and variance rules also assign zero weight at $f_j = 1$, but continuously down-weight varying candidates as their recurrence approaches either extreme. Each rule multiplies all four candidate-level components, including background and duplicate evidence, while retaining the recurrence factor f_j. The binary entropy is

$$H_2(f) = -\frac{f \log f + (1-f)\log(1-f)}{\log 2}, \tag{16}$$

with $0 \log 0 = 0$ and numerical stabilization inside the logarithms in the implementation. The variance rule normalizes Bernoulli variance by its maximum of $1/4$. Both soft rules suppress extreme recurrence values and attain their maximum at $f = 0.5$, but their effective TP-like weights remain $f_j H_2(f_j)$ and $4f_j^2(1-f_j)$, respectively.

Table 7: **Removal and weighting definitions.** The controls separate random pruning from rules that target candidates according to recurrence. All variants retain the recurrence factor f_j; they differ only in the comparison weight a_j.

Variant	Retention weight a_j
No removal	1
Random (count-matched)	$\mathbf{1}\{j \notin R_{\text{count}}\}$
Random (recurrence-mass-matched)	$\mathbf{1}\{j \notin R_{\text{mass}}\}$
Entropy soft rule	$H_2(f_j)$
Variance soft rule	$4f_j(1-f_j)$
Remove $f_j \geq 0.90$	$\mathbf{1}\{f_j < 0.90\}$
Remove $f_j = 1$ (TED)	$\mathbf{1}\{f_j < 1\}$

Matched random removal. Let $\mathcal{I} = \{j : f_j = 1\}$, $N_{\text{inv}} = |\mathcal{I}|$, and $M_{\text{inv}} = \sum_{j \in \mathcal{I}} f_j$. The count-matched control samples R_{count} uniformly without replacement from all candidates, with

exactly N_{inv} candidates removed. The recurrence-mass-matched control randomly permutes all candidates and removes the shortest prefix R_{mass} whose cumulative recurrence reaches or exceeds M_{inv}. Removal is candidate-wise, so the final candidate can cause a small mass overshoot. Since invariant candidates have $f_j = 1$, $M_{\text{inv}} = N_{\text{inv}}$, but matching this mass with lower-recurrence candidates generally requires removing more candidates. For each setup, each random-removal control is evaluated 100 times using distinct random seeds, with the checkpoint predictions and shared candidate set held fixed.

Near-invariant cutoff. With $T_s = 30$, removing candidates with $f_j \geq 0.90$ excludes those observed at 27 or more checkpoints, whereas exact invariant removal excludes only candidates observed at all 30 checkpoints. The broader cutoff therefore suppresses candidates that still contain genuine presence differences, making it a test of whether near-common evidence should also be treated as non-discriminative.

C.3 Scoring functions before and after removal

Table 8 applies the Precision-, Recall-, AP-, and F1-like scoring functions to the same reconstructed evidence before and after invariant-candidate removal. The candidate set, recurrence weights, assessment threshold, and checkpoint set are otherwise unchanged. Removal lowers RMSE_{AP50} for all four scoring functions, with the largest reduction for F1-like scoring (6.105 to 0.743). Its effect under AP_{75} is mixed for the Recall- and AP-like scores, while F1-like scoring improves substantially under all three evaluation metrics. The results show that removal benefits several scoring functions, with its largest gain arising in combination with the F1-like score.

Table 8: **Selection scoring functions before and after invariant-candidate removal on 40 setups.** Entries are oracle-relative RMSE in AP points. Bold marks the lower error within each scoring function and metric.

	No removal ($a_j = 1$)			Remove $f_j = 1$		
Scoring function	RMSE_{AP50}	RMSE_{AP75}	RMSE_{mAP}	RMSE_{AP50}	RMSE_{AP75}	RMSE_{mAP}
Precision-like	9.374	5.823	5.635	**6.674**	**3.923**	**3.775**
Recall-like	1.583	**2.102**	1.449	**1.258**	2.184	**1.391**
AP-like	9.199	**6.366**	5.880	**9.185**	6.372	**5.871**
F1-like	6.105	4.278	3.891	**0.743**	**1.391**	**0.881**

C.4 Controlled common-evidence injection

Table 9 isolates the score sensitivity summarized in Appendix A.2. At injection level γ, we append all-checkpoint candidates whose total recurrence mass is γ times the original total recurrence mass. Each appended candidate has one assessment-level detection at every checkpoint and therefore contributes identical TP-like evidence before removal. Without removal, this shared mass changes the nonlinear scores and eventually alters 15 of 40 selections. With removal, every appended candidate has $a_j = 0$, so the scores and selections remain unchanged when the existing associations are held fixed. Table 4 reports the corresponding effect of naturally occurring invariant candidates.

Table 9: **Controlled common-mode evidence injection.** "Mass" is the injected recurrence mass as a percentage of the original total recurrence mass. "Changes" counts selections that differ from the zero-injection result.

	No removal		TED removal	
Mass	$\max \lvert\Delta q\rvert$	Changes	$\max \lvert\Delta q\rvert$	Changes
0%	0.000	0	0.000	0
10%	0.092	2	0.000	0
25%	0.187	11	0.000	0
50%	0.287	14	0.000	0
100%	0.396	15	0.000	0

C.5 REACHABILITY OF THE CHECKPOINT SET

Table 10 quantifies the portion of target objects detected by at least one checkpoint and therefore available to trajectory reconstruction. Overall, the checkpoint populations expose 72.9% of all GT objects, with substantial variation across setups. Faster R-CNN reaches roughly 91% on S2C and less than half on V2C, while Deformable-DETR reaches nearly 90% on V2C. Reachability therefore depends on the detector–domain trajectory. It records whether an object is detected anywhere in the trajectory, whereas oracle AP additionally reflects localization and checkpoint-specific detection quality. Objects missed by every checkpoint remain outside the observable candidate set.

Table 10: **Reachable GT support.** A GT object is reachable if at least one evaluated checkpoint detects it. Coverage measures the observable scope of cross-checkpoint reconstruction, not checkpoint quality: it varies substantially across detector–shift pairs and need not track oracle AP_{50}.

Shift	Detector	DAOD	GT	Reachable	Unseen	Coverage (%)	Oracle AP_{50}
C2F	Faster R-CNN	AT	30,351	24,124	6,227	79.5	0.5978
C2F	Faster R-CNN	MT	30,351	24,127	6,224	79.5	0.5950
C2F	Faster R-CNN	ALDI++	30,351	25,688	4,663	84.6	0.6450
C2F	Faster R-CNN	CMT	30,351	24,263	6,088	79.9	0.5971
C2F	Deformable-DETR	AT	30,351	22,630	7,721	74.6	0.4430
C2F	Deformable-DETR	MT	30,351	22,626	7,725	74.5	0.4435
C2F	YOLOv5	AT	30,351	25,487	4,864	84.0	0.6318
C2F	YOLOv5	MT	30,351	25,653	4,698	84.5	0.6361
S2C	Faster R-CNN	AT	4,652	4,229	423	90.9	0.7672
S2C	Faster R-CNN	MT	4,652	4,219	433	90.7	0.7669
S2C	Faster R-CNN	ALDI++	4,652	4,319	333	92.8	0.7707
S2C	Faster R-CNN	CMT	4,652	4,220	432	90.7	0.7653
S2C	Deformable-DETR	AT	4,652	3,452	1,200	74.2	0.5634
S2C	Deformable-DETR	MT	4,652	3,483	1,169	74.9	0.5652
S2C	YOLOv5	AT	4,652	3,985	667	85.7	0.7390
S2C	YOLOv5	MT	4,652	3,961	691	85.1	0.7360
K2C	Faster R-CNN	AT	4,652	4,048	604	87.0	0.7085
K2C	Faster R-CNN	MT	4,652	4,047	605	87.0	0.7093
K2C	Faster R-CNN	ALDI++	4,652	4,106	546	88.3	0.7098
K2C	Faster R-CNN	CMT	4,652	4,049	603	87.0	0.7073
K2C	Deformable-DETR	AT	4,652	3,012	1,640	64.7	0.4648
K2C	Deformable-DETR	MT	4,652	3,021	1,631	64.9	0.4571
K2C	YOLOv5	AT	4,652	3,025	1,627	65.0	0.5601
K2C	YOLOv5	MT	4,652	3,038	1,614	65.3	0.5641
V2C	Faster R-CNN	AT	1,526	755	771	49.5	0.3029
V2C	Faster R-CNN	MT	1,526	743	783	48.7	0.3019
V2C	Faster R-CNN	ALDI++	1,526	896	630	58.7	0.3544
V2C	Faster R-CNN	CMT	1,526	756	770	49.5	0.3058
V2C	Deformable-DETR	AT	1,526	1,372	154	89.9	0.4099
V2C	Deformable-DETR	MT	1,526	1,352	174	88.6	0.4125
V2C	YOLOv5	AT	1,526	875	651	57.3	0.3520
V2C	YOLOv5	MT	1,526	855	671	56.0	0.3538
X2D	Faster R-CNN	AT	30,593	20,443	10,150	66.8	0.4534
X2D	Faster R-CNN	MT	30,593	20,409	10,184	66.7	0.4482
X2D	Faster R-CNN	ALDI++	30,593	22,049	8,544	72.1	0.4523
X2D	Faster R-CNN	CMT	30,593	20,418	10,175	66.7	0.4463
X2D	Deformable-DETR	AT	30,593	17,095	13,498	55.9	0.2235
X2D	Deformable-DETR	MT	30,593	17,159	13,434	56.1	0.2233
X2D	YOLOv5	AT	30,593	19,212	11,381	62.8	0.4085
X2D	YOLOv5	MT	30,593	19,167	11,426	62.7	0.4130
Setup average (40 setups)						73.6	–
All setups combined			574,192	418,368	155,824	72.9	–

C.6 CHECKPOINT BUDGET

Table 11 evaluates candidate trajectories constructed from evenly spaced checkpoint subsets. Sixteen checkpoints leave all three RMSE values nearly unchanged, and eight checkpoints preserve the AP_{50} result with modest degradation under the stricter metrics. With four checkpoints, AP_{50} RMSE increases from 0.743 to 2.063, indicating that sparse observations weaken recurrence and absence evidence. The Top-3 rate decreases earlier than RMSE because a selected checkpoint can move

outside the oracle's top three while remaining close in AP. The oracle remains defined over the full trajectory, so each subset changes both the available trajectory evidence and the selectable checkpoints.

Table 11: **Checkpoint-budget sensitivity.** Candidate sets are rebuilt from evenly spaced checkpoint subsets, while the oracle remains defined over the full trajectory. Sixteen checkpoints preserve the full-budget error, whereas four are insufficient to maintain reliable trajectory evidence.

T	$\mathrm{RMSE}_{AP50}\downarrow$	$\mathrm{RMSE}_{AP75}\downarrow$	$\mathrm{RMSE}_{mAP}\downarrow$	Top-1	Top-3
Full	0.743	1.391	0.881	0.45	0.75
16	0.774	1.392	0.900	0.42	0.45
8	0.775	1.454	0.966	0.42	0.45
4	2.063	2.069	1.632	0.35	0.40

C.7 Candidate association

Table 12 shows that all-pair and cross-checkpoint-only union–find association produce nearly identical selection errors and GT structure. Temporal one-to-one matching eliminates same-checkpoint multiplicity and reduces multi-GT merges, but increases the number of candidates per detected GT from approximately 1.00 to 1.35 and raises selection RMSE. The one-to-one constraint fragments object trajectories across time, weakening the recurrence and absence patterns used for checkpoint comparison. These results favor temporal continuity over locally purer pairwise associations.

Table 12: **Candidate-association rules at $\tau_{\mathrm{assoc}} = 0.40$.** Union–find variants preserve object continuity and yield similar selection errors. Temporal one-to-one matching reduces local merge artifacts but fragments detected objects, illustrating that cleaner pairwise matching need not produce better checkpoint-level evidence.

Association	$\mathrm{RMSE}_{AP50}\downarrow$	$\mathrm{RMSE}_{AP75}\downarrow$	$\mathrm{RMSE}_{mAP}\downarrow$	Same-checkpoint multiplicity	Multi-GT fraction	Candidates per detected GT
A: all-pair union–find	0.743	1.391	0.881	0.397	0.166	1.001
B: cross-checkpoint-only	0.750	1.391	0.881	0.380	0.165	1.001
C: temporal one-to-one	1.610	1.833	1.435	0.000	0.034	1.351

C.8 Oracle-location sensitivity

Table 13 stratifies selection error by the position of the AP_{50} oracle. Twenty-nine trajectories have an oracle within the first four checkpoints, five in the middle, and six in the late group. TED has the lowest RMSE in all three groups and on the oracle-index-≥ 5 subset. Its margin over DIV is small in the middle group and larger for late oracles and the combined non-early subset.

Table 13: AP_{50} **selection error stratified by oracle location.**

Oracle position	Setups	TED	DIV	SND
Early: 0–3	29	**0.395**	3.573	<u>2.977</u>
Middle: 4–9	5	**1.834**	<u>1.854</u>	2.293
Late: ≥ 10	6	**0.346**	2.848	<u>2.665</u>
Oracle index ≥ 5	11	**1.263**	<u>2.447</u>	2.503
All	40	**0.743**	3.302	<u>2.855</u>

C.9 Rising versus falling trajectories

The six rising trajectories and 34 falling trajectories have similar mean score–AP_{50} correlations of 0.562 and 0.572, with selection RMSEs of 0.340 and 0.793 AP points. The score therefore follows target performance in both temporal directions without a fixed preference for earlier or later checkpoints. This directional split complements the oracle-location analysis by summarizing the overall trajectory trend rather than the position of a single maximum.

C.10 TARGET-PERFORMANCE TRAJECTORIES AND CHECKPOINT SELECTIONS

Figures 5 and 6 cover all 40 setups in two 20-setup grids.

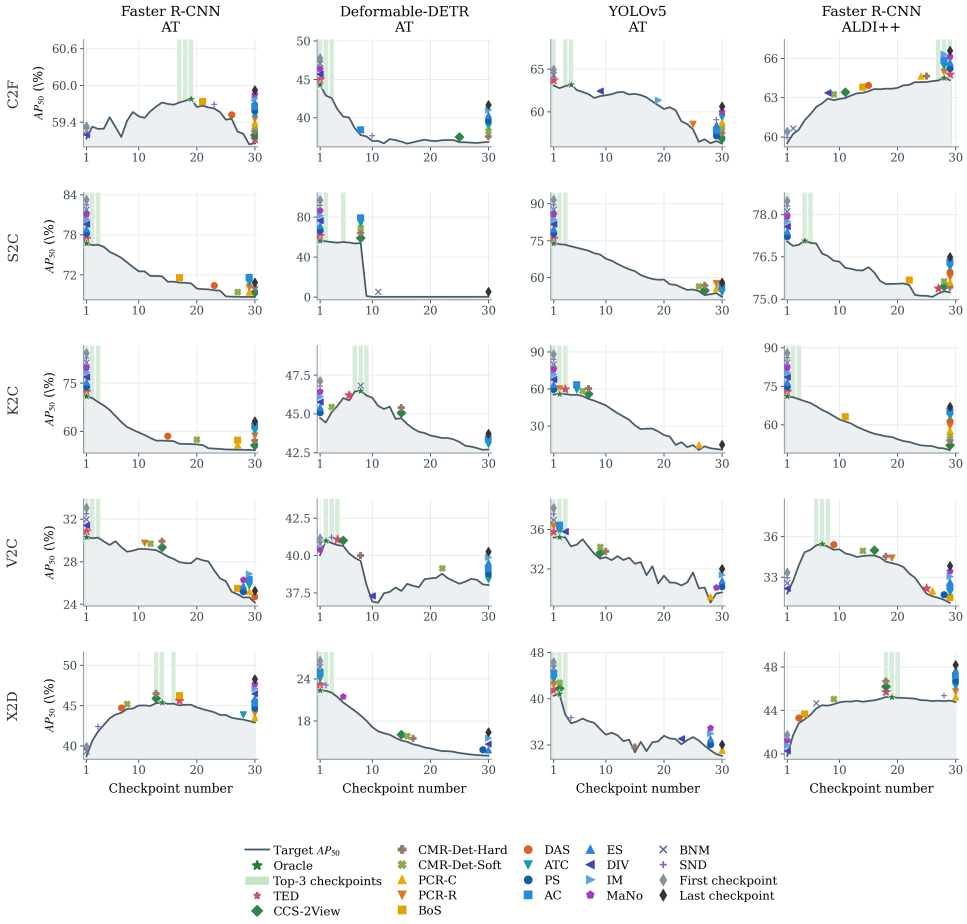

Figure 5: **Target AP_{50} and fixed selections: AT and ALDI++ (20 setups).** Rows denote shifts. The first three columns show AT with Faster R-CNN, Deformable-DETR, and YOLOv5; the fourth shows ALDI++ with Faster R-CNN.

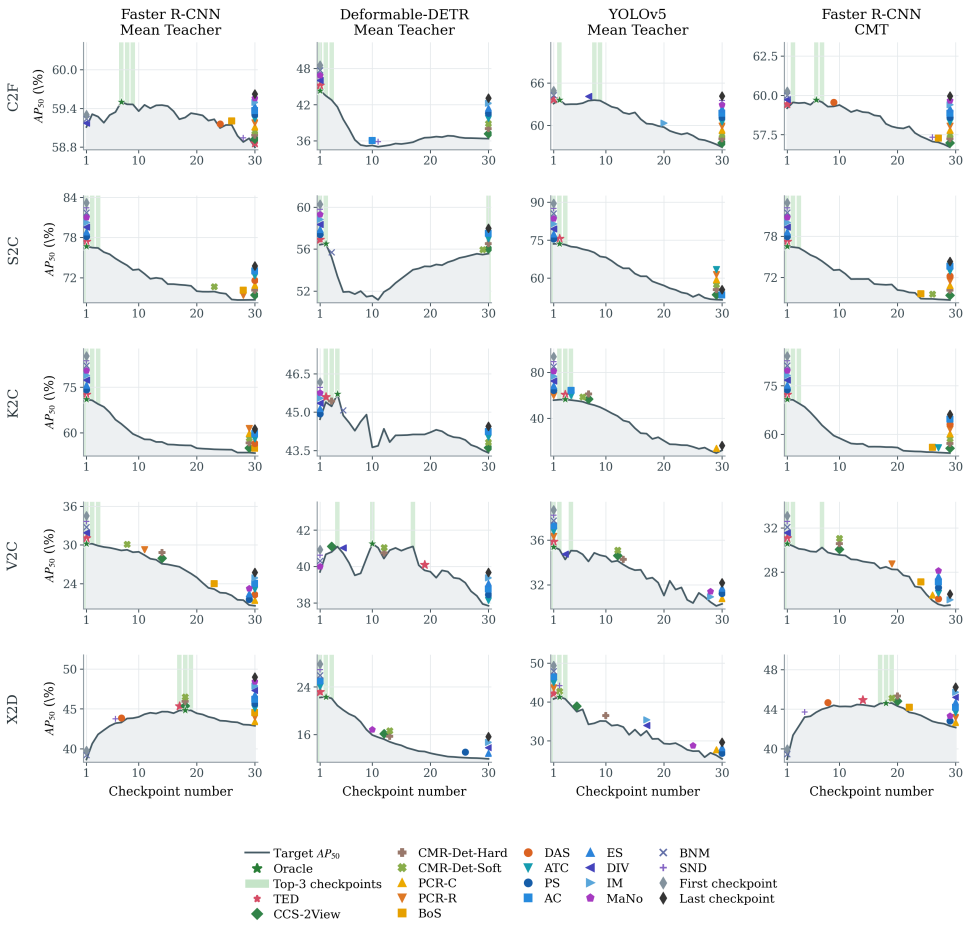

Figure 6: **Target AP_{50} and fixed selections: Mean Teacher and CMT (20 setups).** Rows denote shifts. The first three columns show Mean Teacher with Faster R-CNN, Deformable-DETR, and YOLOv5; the fourth shows CMT with Faster R-CNN.

C.11 CHECKPOINT-WISE ERROR RECONSTRUCTION

Figures 7–10 compare GT TP, FP, and FN trajectories with TED before common-mode evidence removal on all 40 setups. Each curve is independently min–max normalized across checkpoints.

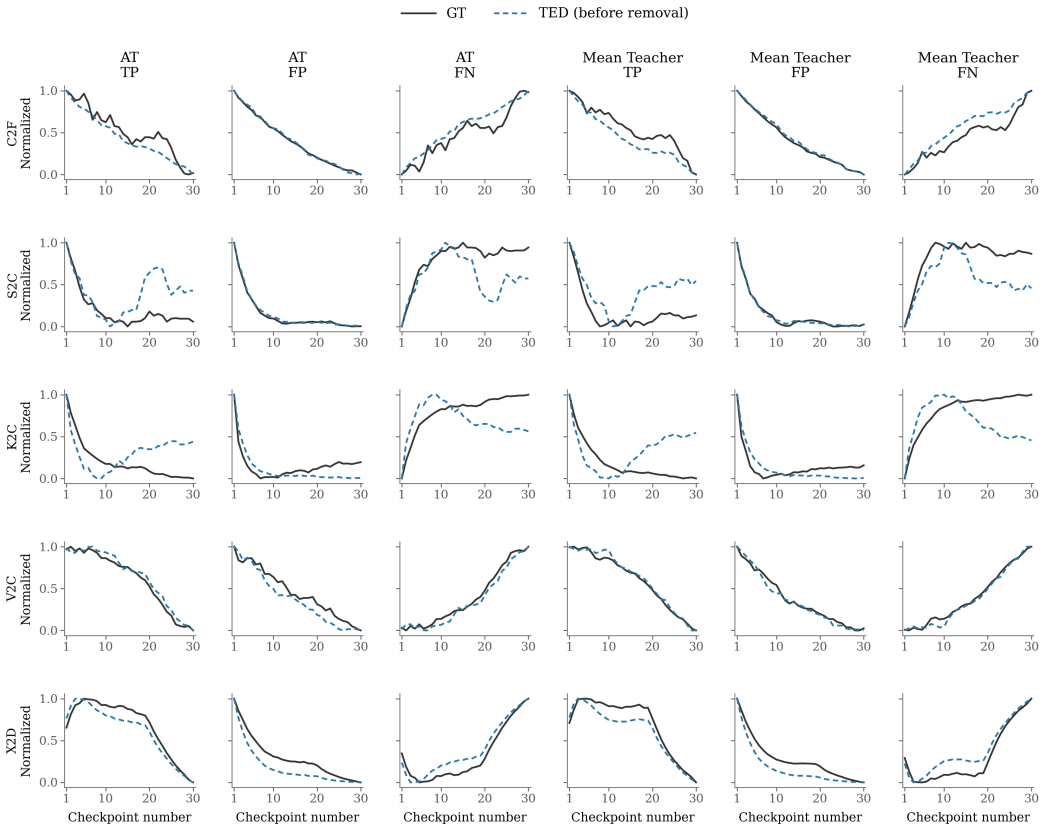

Figure 7: **Faster R-CNN reconstruction: AT and Mean Teacher (10 setups).** Rows denote shifts; the left and right three-column groups show AT and Mean Teacher, respectively. Gray shows GT and dashed blue shows TED before common-mode evidence removal.

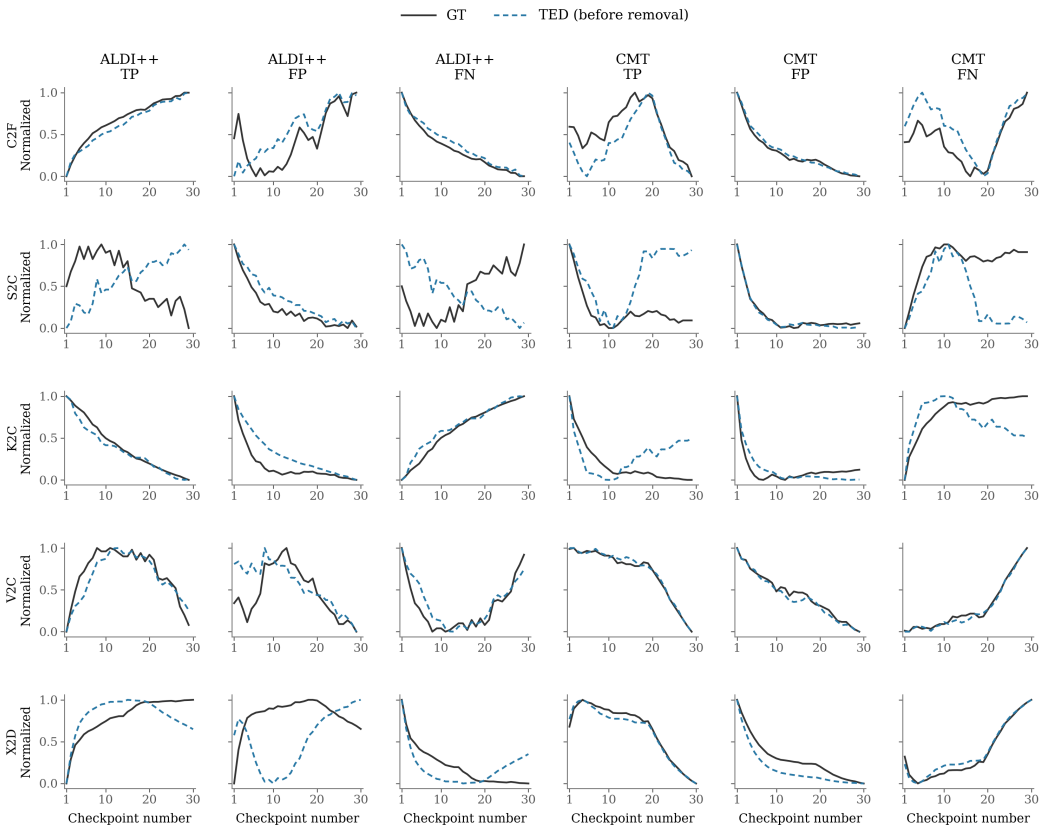

Figure 8: **Faster R-CNN reconstruction: ALDI++ and CMT (10 setups).** Rows denote shifts; the left and right three-column groups show ALDI++ and CMT, respectively. Gray shows GT and dashed blue shows TED before common-mode evidence removal.

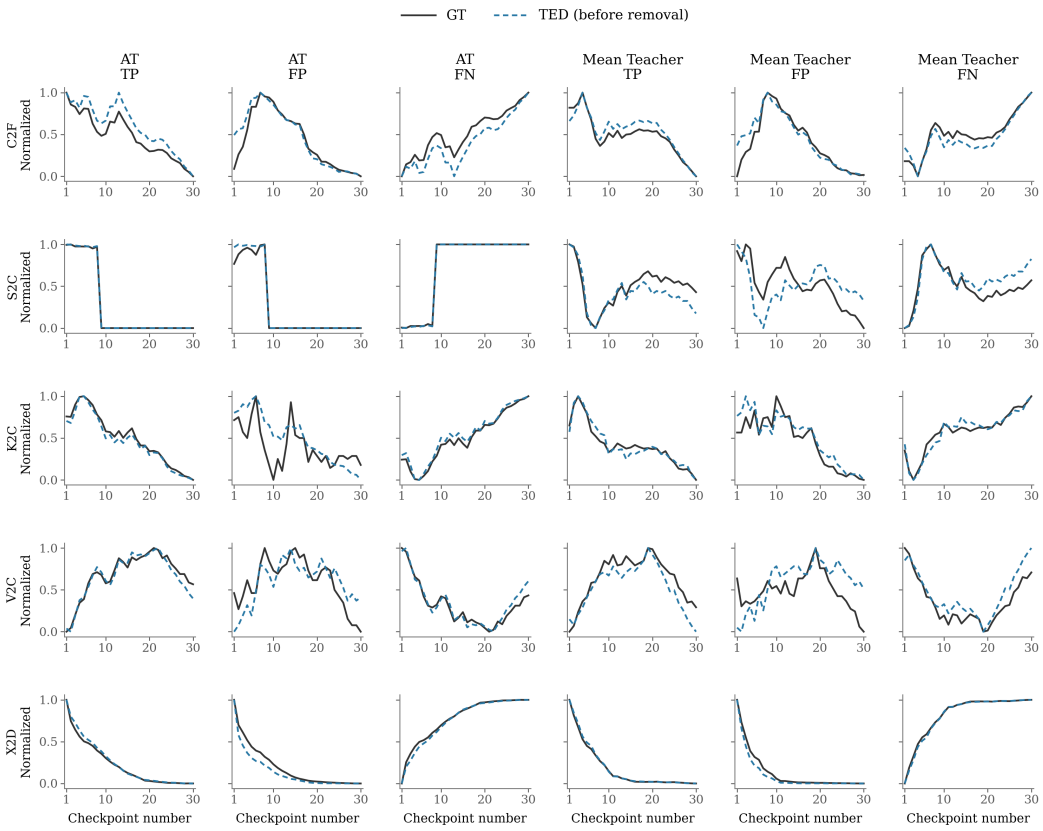

Figure 9: **Deformable-DETR reconstruction: AT and Mean Teacher (10 setups).** Rows denote shifts; the left and right three-column groups show AT and Mean Teacher, respectively. Gray shows GT and dashed blue shows TED before common-mode evidence removal.

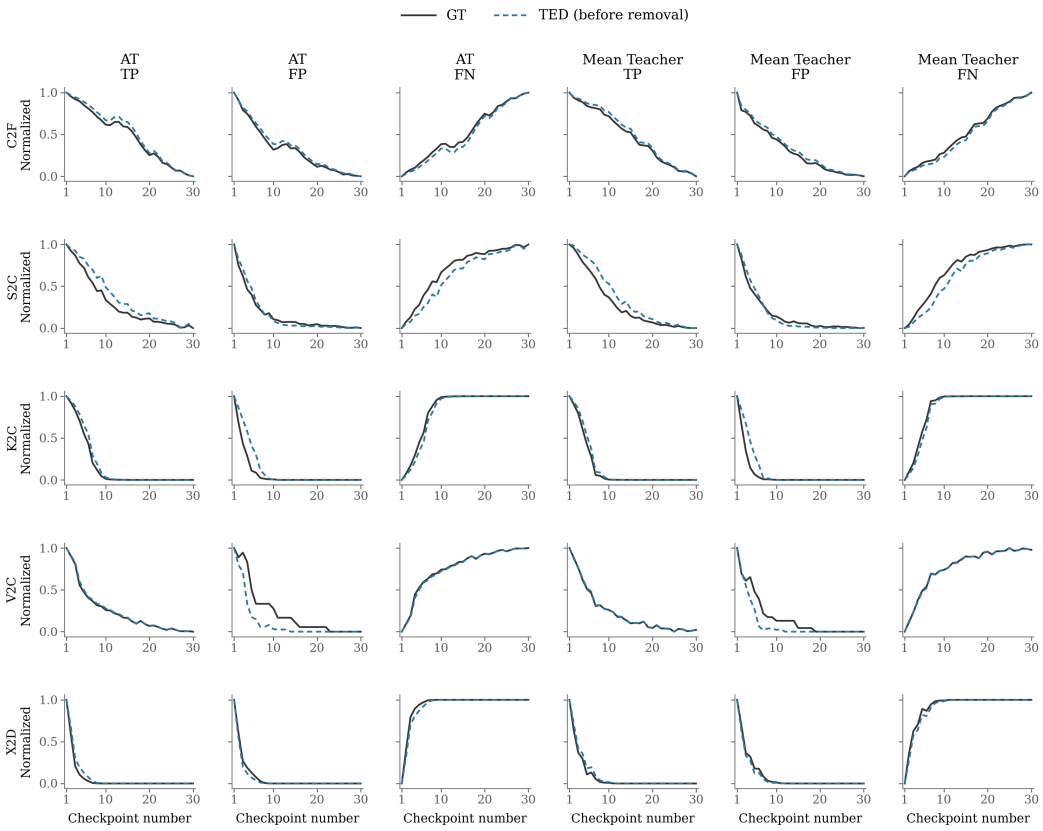

Figure 10: **YOLOv5 reconstruction: AT and Mean Teacher (10 setups).** Rows denote shifts; the left and right three-column groups show AT and Mean Teacher, respectively. Gray shows GT and dashed blue shows TED before common-mode evidence removal.