

# Supplementary Information:

## Cross-view agreement rescoring improves tiled small-object detection without retraining

Hyun-Woo Na<sup>1</sup>, Eunkyung Kim<sup>2,\*</sup>, and Haneol Jang<sup>1,\*</sup>

<sup>1</sup>Department of Computer Engineering, Hanbat National University, Daejeon, Republic of Korea

<sup>2</sup>Hanbat National University, Department of Artificial Intelligence Software, 109 Jiphyeonbuk-ro, Sejong 30139, Republic of Korea

\*Corresponding authors: ekim@hanbat.ac.kr; hejang@hanbat.ac.kr

### ABSTRACT

This document contains the complete analyses summarised in the main text: the full related-work discussion, the oracle decomposition, per-component ablations, the complete merge-rule battery including Soft-NMS and CMA-NMS, the label-tier comparison, portability across thirteen detectors and two capacity ladders, the overlap–gain curve, orthogonality with adaptive slicing, generalisation to DOTA-v1.5 and AI-TOD-v2 (validation and held-out test), open-vocabulary experiments, mechanism and per-class analyses, per-image gain distributions, computational cost, comparisons against published numbers, and training details for the evaluation-subject detectors.

### Contents

|          |                                                                                              |          |
|----------|----------------------------------------------------------------------------------------------|----------|
| <b>1</b> | <b>Extended related work</b>                                                                 | <b>2</b> |
| 1.1      | Related Work                                                                                 | 2        |
| <b>2</b> | <b>Method details and design rationale</b>                                                   | <b>5</b> |
| 2.1      | Method                                                                                       | 5        |
| 2.2      | Notation and view generation                                                                 | 5        |
| 2.3      | The effective view set $C(d)$                                                                | 5        |
| 2.4      | Rescoring                                                                                    | 5        |
| 2.5      | Properties                                                                                   | 5        |
| 2.6      | Experimental setup                                                                           | 5        |
| <b>3</b> | <b>Full experimental analyses</b>                                                            | <b>6</b> |
| 3.1      | Oracle decomposition: where is the bottleneck?                                               | 6        |
| 3.2      | Main result: VisDrone test-dev                                                               | 7        |
| 3.3      | Component ablation                                                                           | 8        |
| 3.4      | Ablating $C(d)$ : an enabling condition                                                      | 9        |
| 3.5      | Hyper-parameter sensitivity                                                                  | 9        |
| 3.6      | What do labels buy? The futility of recalibration                                            | 10       |
| 3.7      | Comparison against all standard merge rules                                                  | 10       |
| 3.8      | Does the gain stack with Soft-NMS?                                                           | 12       |
| 3.9      | Does the gain survive a crop-aware merge? CMA-NMS                                            | 13       |
| 3.10     | Portability: thirteen detectors across three families and two capacity ladders, no re-tuning | 13       |
| 3.11     | The overlap–gain curve                                                                       | 14       |
| 3.12     | Orthogonality: CAR on top of adaptive slicing                                                | 15       |
| 3.13     | Generalisation: DOTA-v1.5 without re-tuning                                                  | 15       |
| 3.14     | Generalisation: AI-TOD-v2, twelve detectors, no re-tuning                                    | 16       |
| 3.15     | Held-out confirmation: AI-TOD-v2 test                                                        | 16       |
| 3.16     | Open-vocabulary detection: where fine-tuning is impossible                                   | 17       |

|      |                                                                            |           |
|------|----------------------------------------------------------------------------|-----------|
| 3.17 | Why it works: discriminability, mechanism, qualitative cases               | 17        |
| 3.18 | Per-class analysis: which bottleneck limits which class                    | 18        |
| 3.19 | How the gain distributes across images                                     | 19        |
| 3.20 | Computational cost                                                         | 20        |
| 3.21 | Overhead across all benchmarks: bounded by detection density               | 20        |
| 3.22 | Relation to published numbers                                              | 20        |
| 4    | <b>Training details and reproduction</b>                                   | <b>21</b> |
| 4.1  | RT-DETR training details (evaluation subjects; not part of the method)     | 21        |
| 4.2  | Faster R-CNN training details (evaluation subject; not part of the method) | 21        |
| 4.3  | Reproduction information                                                   | 21        |
| 4.4  | Robustness across image-size groups                                        | 22        |
| 4.5  | $\sigma$ -conditionality in detail                                         | 22        |
| 4.6  | Mechanism: definitions and per-class detail                                | 22        |
|      | <b>References</b>                                                          | <b>23</b> |

## 1 Extended related work

### 1.1 Related Work

Table 1 summarises the methods closest to this work: the axis on which each intervenes, whether it needs training or extra computation, and what separates it from CAR. Two observations organise the section. First, the crowded axis is *where to crop*; the merging stage that our diagnosis identifies as the bottleneck (Section 3.1) has attracted far less attention. Second, several works touch the same raw signal — detections repeating across overlapping views — but none normalises it by per-box visibility, which Table 8 shows is the difference between working and failing.

**The model-side axis** A large body of tiny-object work improves the *detector itself*: receptive-field-aware label assignment (RFLA<sup>1</sup>) and query-adaptive DETR architectures (DQ-DETR<sup>2</sup>) are representative. These methods retrain or redesign the model and are evaluated end-to-end. They are orthogonal to this paper in the strict sense: CAR operates at inference time on the outputs of *whatever* detector is deployed — including tiny-object-specialised ones — and modifies no weights. The portability experiment (Section 3.10) is the empirical form of this claim.

**Sliced inference and “where to crop”** SAHI<sup>3</sup> raised small-object AP substantially using inference slicing alone and proposed NMS/NMM merging. Follow-up work has largely improved the cropping side: resolution-adaptive slicing (ASAHI<sup>4</sup>) and training-free crop selection driven by decoder cross-attention entropy (ViCrop-Det<sup>5</sup>). A 2026 preprint reuses the ASAHI acronym for a different method<sup>13</sup> that couples resolution-adaptive slice counts with slicing-assisted *fine-tuning* and a Cluster-DIoU-NMS merge; it therefore belongs to the label-using tier and, like its namesake, works the cropping axis rather than the scoring axis addressed here. Our oracle O3 — retaining only the best view per ground-truth object — was effectively identical to the baseline (26.45 vs. 26.40 AP): on a fixed grid at this image scale, *little remains to be gained by choosing views better*, and the residual gain lies in merging. On much larger images this is not so — on DOTA-v1.5 the same oracle is worth +4.67 AP (Section 3.1) — which is precisely why we treat the two directions as composable rather than competing. We verify that composition empirically in Section 3.12, placing CAR on top of an ASAHI-style adaptive grid and showing that the gains accumulate. Their applicability also differs: ViCrop-Det depends on decoder cross-attention and is thus DETR-specific, whereas CAR uses only scores and geometry and is detector-agnostic (Section 3.10).

**Merging: suppression, coordinates, and agreement** Improved suppression rules (Soft-NMS<sup>6</sup> and relatives) refine *which* boxes are removed but preserve the original score ordering — and our diagnosis is precisely that this ordering is the bottleneck. WBF<sup>7</sup> score-weight-averages box coordinates and finally rescales confidence by cluster size,  $C \cdot \min(n, T)/T$ . We separate the two components and provide a counterexample for each: coordinate averaging is consistently harmful in the multi-view small-object setting (Tables 6, 7; localisation noise from different framings is mixed, lowering IoU), and the denominator of the confidence rescaling (total member count) drives the method below the baseline under tiled inference (Table 8). CAR repairs the latter with visibility normalisation. Section 3.7 extends this to a full comparison — Soft-NMS (linear and Gaussian), Weighted-NMS (box voting) and SAHI’s Greedy-NMM — *measured on identical detections*.

**Concurrent work** TA-TM<sup>8</sup> independently proposed a training-free post-processing step for PCB defect detection that adjusts scores using neighbouring-tile agreement — an independent rediscovery that supports the validity of the signal. A head-to-head comparison is precluded by disjoint domains and metrics (PCB-defect mAP@50 in a saturated 72–94 regime versus aerial AP in the 20–45 range); we therefore contrast the functional forms directly, and additionally transplant TA-TM’s rule onto our

**Table 1.** Positioning against the closest related methods. “TF” = training-free at deployment; “+0 FP” = no forward passes beyond the tiled baseline. None of the listed methods normalises cross-view agreement by *per-box visibility* — the component shown to be decisive in Table 8.

| Method                                    | Axis                                       | TF | +0 FP | Key difference from CAR                                                                                                                            |
|-------------------------------------------|--------------------------------------------|----|-------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| RFLA <sup>1</sup><br>(ECCV 2022)          | detector: label assign-<br>ment            | ✗  | —     | retrains the detector; orthogonal — CAR post-<br>processes any detector                                                                            |
| DQ-DETR <sup>2</sup><br>(ECCV 2024)       | detector: architecture                     | ✗  | —     | same axis as above; complementary, not competing                                                                                                   |
| SAHI <sup>3</sup> (ICIP 2022)             | cropping + merge                           | ✓  | ✓     | its merge compares scores across framings on<br>one axis; CAR replaces exactly this stage (default<br>NMM: up to $-6.1$ AP <sub>s</sub> , Tab. 10) |
| ASAHI <sup>4</sup> (Remote<br>Sens. 2023) | cropping: adaptive<br>grid                 | ✓  | ✓     | improves where to crop; composable — CAR adds<br>+1.70 AP on top (Tab. 14)                                                                         |
| ViCrop-Det <sup>5</sup> (2026)            | cropping: attention-<br>guided             | ✓  | ✓     | DETR-specific crop selection; CAR is detector-<br>agnostic (Tab. 12)                                                                               |
| Soft-NMS <sup>6</sup><br>(ICCV 2017)      | merge: suppression                         | ✓  | ✓     | preserves the original score ordering — the very<br>bottleneck (74%, Tab. 4)                                                                       |
| WBF <sup>7</sup> (IVC 2021)               | merge: coords +<br>cluster-size rescale    | ✓  | ✓     | denominator = total members; under tiling this<br>falls below baseline (Tab. 8); coordinate averaging<br>harmful (NR2)                             |
| TA-TM <sup>8</sup> (2026)                 | merge: additive,<br>boundary-only          | ✓  | ✓     | can only <i>raise</i> scores near tile borders; cannot<br>demote interior high-confidence FPs ( $-3.60$ AP<br>transplanted, Tab. 10)               |
| Group Evidence <sup>9</sup><br>(2025)     | merge: candidate re-<br>vival              | ✓  | ✗     | needs extra embedding passes; recall-first trade-off<br>rather than AP                                                                             |
| CRFace <sup>10</sup><br>(CVPR 2021)       | scoring: learned<br>ranker                 | ✗  | ✓     | trains the ranker; we show training buys<br>+0.05 AP <sub>s</sub> at most (NR1)                                                                    |
| PCR <sup>11</sup> (ICCV 2025)             | consistency → AutoE-<br>val                | ✓  | ✓     | same observation (pre-NMS boxes carry signal),<br>used to <i>estimate</i> accuracy, not to correct ranking                                         |
| CCS <sup>12</sup> (2025)                  | consistency → moni-<br>toring              | ✓  | ✓     | deployment-time evaluation, not merging                                                                                                            |
| <b>CAR (ours)</b>                         | merge: visibility-<br>normalised agreement | ✓  | ✓     | the denominator $C(d)$ : count only the views that<br>could have seen the box                                                                      |

detection pool in Section 3.7. The differences lie in scope and definition. TA-TM builds a tile-adjacency graph and applies  $s' = \min(1, s + \lambda A)$  with  $\lambda = 0.2$  *only to detections near tile boundaries* (16 px by default), where  $A$  is the *maximum* confidence among qualifying neighbours. We define, for *every* detection, a per-box visibility set  $C(d)$  and apply multiplicative reweighting by the *ratio*  $\text{agr} = |M|/|C|$ . This distinction is not cosmetic: an additive form with  $A \geq 0$  *can only raise scores, never lower them*. That suffices to recover boundary artefacts, but makes it structurally impossible to demote a high-confidence false positive — such as the 0.91-confidence detection that no view reproduced and that CAR moves from rank 23 to rank 2,402 (Section 3.17). The ranking loss identified by our diagnosis (Section 3.1) is a global phenomenon, not a boundary one; consistently, the gains TA-TM reports are +0.001 to +0.002 mAP@50, which the authors themselves describe as conservative refinement. We add a bottleneck diagnosis, statistical testing on standard aerial benchmarks, and portability across thirteen detectors.

Group Evidence Matters<sup>9</sup> validates low-confidence candidates revived by overlapping tiles in UAV imagery using spatial and semantic DBSCAN gates before reweighting. It shares the absence of retraining, but its semantic gate requires ResNet-18 embeddings and therefore *additional forward passes*, violating our Pareto condition (zero increase in forward passes). Its reported metric also differs: recall 0.685  $\rightarrow$  0.778 against precision 0.801  $\rightarrow$  0.595, a recall-first trade-off on a different axis from the ranking (AP) improvement claimed here.

**Learned rescoring** Rescoring an existing detection set to maximise AP is itself an established idea: Pato et al. train a bidirectional-RNN context model that adjusts the confidences of a detector’s outputs — without touching the detector — and improve AP on COCO<sup>14</sup>. CAR shares the goal (repairing the order of a fixed box set) but not the means: the signal is geometric agreement across views rather than a learned context model, no training data is required, and the method operates inside tiled inference, where the cross-view evidence exists in the first place. Our label-trained control (main text) bounds what learned rescoring adds on top of the agreement signal in this setting: +0.05 AP<sub>s</sub>.

**Other uses of consistency signals** Closest in motivation is PCR<sup>11</sup>, which starts from the same observation we do — the candidate boxes a detector discards at NMS still carry usable information — and aggregates pre-/post-NMS spatial consistency and overlap-confidence reliability into a label-free *estimate of detector accuracy* (AutoEval). CCS<sup>12</sup> similarly monitors deployed detectors through cumulative consensus, and prediction consistency under test-time augmentation has long served uncertainty estimation. All of these use the signal to *evaluate* a detector. We use a visibility-normalised form of the same signal to *re-rank* detections inside the merging stage, where it changes AP directly; and we show that its form must aggregate *whether* a detection reproduces rather than average scores — in Table 7, noisy-OR and averaging-style aggregations are all inferior to weighted maximum.

**Calibration and uncertainty** A parallel literature asks whether detector confidences are *probabilistically* correct rather than correctly *ordered*. Küppers et al. generalise the expected calibration error to detection as D-ECE and fit post-hoc calibrators conditioned on box position and scale while treating the detector as a black box<sup>15</sup>; Kuzucu et al. show that calibration and accuracy must be measured jointly, and that cheap post-hoc calibrators outperform train-time ones<sup>16</sup>. That line targets a different quantity from ours. AP depends only on the ranking, so a calibrator that is a *strictly* monotone function of the score alone cannot change it at all; practical detection calibrators are therefore *conditional* —<sup>15</sup> conditions on box position and scale — and any such conditioning perturbs the global ordering, which is exactly why<sup>16</sup> insist that calibration and accuracy be measured jointly. The question is thus not whether recalibration moves AP but whether it moves it in a useful direction, and we answer that empirically rather than by assertion: a  $(\sigma \times a_{\text{net}})$ -binned isotonic regressor fitted *in sample on true ground truth* — an upper bound on what recalibration can do here — buys only +0.32 AP while *losing* 0.22 AP<sub>s</sub> (Section 3.6). It reorders without improving; CAR instead supplies information the score does not carry. The closest point of contact is IoU-aware calibration<sup>17</sup>, which folds duplicate modelling into the confidence estimate so that NMS can be dropped; it reasons about duplication *within* a single view and needs an IoU-prediction head, whereas our evidence is (dis)agreement *between* views of the same object at different magnifications. Consistency is also the basis of sampling-based uncertainty — MC-dropout detection<sup>18</sup> and deep ensembles<sup>19</sup> — but those require repeated forward passes, which is exactly the budget we refuse to spend: our repeated measurements already exist inside the tiling that SAHI performs anyway.

**Improving score ranking** CRFace<sup>10</sup> showed that a learned confidence ranker recovers much of the benefit of multi-scale inference in face detection — a precedent for our diagnosis that “the gain comes from better ranking, not better boxes”. CRFace, however, *trains* the ranker. We present a training-free counterpart and further show that, in this setting, training does not even help (Section 3.6).

**Open-vocabulary detection** YOLO-World<sup>20</sup> and Grounding DINO<sup>21</sup> detect arbitrary classes from text prompts. Because the test-time class set is unknown during training, fine-tuning is impossible in principle and score calibration is notoriously poor. This is a natural setting for a training-free post-processor, and we verify CAR on two architecturally distinct OVD models (Table 17).

## 2 Method details and design rationale

### 2.1 Method

#### 2.2 Notation and view generation

For an image  $I$  we construct a view set  $V = \{v_0, v_1, \dots\}$ . View  $v_0$  is the full image resized to the detector input size  $N$ ; the remainder form a SAHI grid with slice size  $S$  and overlap ratio  $r$ . Writing  $R_v$  for the region of view  $v$ , its magnification is  $\sigma_v = N / \max(H_{R_v}, W_{R_v})$ . Each view is passed through the *frozen* detector and the resulting boxes are mapped back to original image coordinates. Up to this point the pipeline is identical to SAHI; only the subsequent merging stage is replaced — *the number of forward passes is the same as the baseline*.

#### 2.3 The effective view set $C(d)$

For a detection  $d$  with box  $b$ , score  $s$ , class  $c$  and original size  $a = \sqrt{wh}$ :

$$C(d) = \{u \in V : b \subset R_u, \text{dist}(b, \partial R_u) \geq m \cdot a\}, \quad (1)$$

$$M(d) = \{u \in C(d) : \exists d' \in u, \text{cls}(d') = c, \text{IoU}(d', d) \geq \tau_a\}, \quad (2)$$

$$\text{agr}(d) = |M(d)| / |C(d)|. \quad (3)$$

That is, Eq. (1) collects the views that *could have seen*  $d$ , Eq. (2) those that actually re-detected it, and Eq. (3) takes their ratio; if  $C(d) = \emptyset$  we set  $\text{agr} = 0$ . The intuition is simple: *a view in which the box lay outside the crop cannot count as evidence against it*. (The empirical premise on which this rests — that agreement is discriminative at a fixed score — is verified in Figure 12.) Large objects do not fit inside tiles and therefore have small  $|C(d)|$  — this is not a bias but a faithful representation, because genuinely fewer valid repeated measurements exist for them.

This denominator decides whether the method works. If, following WBF’s confidence rescaling, one divides by the total number of views  $|V|$  — in an ensemble where every member sees everything the two definitions coincide, since  $|C(d)| = |V|$ , but under tiling they do not — the method falls below the baseline (Section 3.4, a gap of 2.51 AP<sub>s</sub>). One exception is applied to the boundary margin: when an edge of a view coincides with an edge of the image itself, no margin is required on that edge, since an object at the image border can never be distant from it in any view. This exception added +0.36 AP<sub>s</sub> on the development split (21.14 → 21.50).

#### 2.4 Rescoring

$$\hat{s}(d) = s(d) \cdot (1 + \lambda \cdot \text{agr}(d)) \quad (4)$$

All detections from all views are re-ranked by  $\hat{s}$  and class-wise NMS is applied. Two choices of form are justified experimentally: (i) *weighting, not replacement* — replacing the score with a calibrated probability loses 1.37 AP relative to weighting (Table 9); the ranking information carried by the detector score is too valuable to discard; and (ii) *maximum-based aggregation* — noisy-OR combination of cluster scores is inferior to weighted maximum (Table 7). The value of repeated measurement lies in confirming reproduction, not in accumulating probability.

#### 2.5 Properties

- **Zero training, zero labels:** nothing is fitted to any data.
- **Zero additional forward passes:** CPU post-processing on detections SAHI has already computed (+18.7 ms/image, Section 3.20).
- **Detector-agnostic:** only scores and geometry are used, so any detector with the same output schema is supported — the structural reason for the portability of Section 3.10.
- **One hyper-parameter:**  $\lambda$  (the matching threshold  $\tau_a$  is insensitive over a wide range, Section 3.3).

#### 2.6 Experimental setup

**Datasets and splits** Development (DEV) and test (TEST) were separated strictly at the image level (Table 2). All tuning, ablation and analysis were carried out on DEV; TEST was used *exactly once*, to produce the main result, enforced by a marker file that locks the timestamp and configuration. The DOTA and OVD experiments are hyper-parameter transfer checks and involve no tuning on any split.

**Frozen configuration**  $S = 512$ ,  $r = 0.6$ ,  $\lambda = 5.0$ ,  $\tau_a = 0.75$ ,  $m = 0.5$  (margin waived on image edges), final class-wise NMS at IoU 0.6 (a merge threshold, unrelated to the tile overlap ratio  $r$ , and identical for B2 and CAR), pre-merge score filter  $s \geq 0.02$ . These were fixed in a configuration file *before* the TEST run. (*The specific value of  $\lambda$  is uncritical: on DEV, any  $\lambda \in [2, 5]$  moves AP<sub>s</sub> by only 0.12; see the sensitivity analysis in Section 3.3.*)

**Table 2.** Evaluation data.

| Dataset                            | Images | Role                          | Notes                                   |
|------------------------------------|--------|-------------------------------|-----------------------------------------|
| VisDrone2019-DET val <sup>22</sup> | 548    | DEV — all tuning              | 10 classes; 74.5% are $1360 \times 765$ |
| VisDrone2019-DET test-dev          | 1,610  | <b>TEST — used once</b>       | primary endpoint                        |
| DOTA-v1.5 val <sup>23</sup>        | 458    | generalisation (no re-tuning) | native resolution, 16 classes           |

**Table 3.** Where each operating point is used. No result is compared across operating points; all head-to-head comparisons are within a row.

| Operating point                     | Used in                                          | Status                        |
|-------------------------------------|--------------------------------------------------|-------------------------------|
| $S = 640, r = 0.25$ (SAHI default)  | Tables 4, 7, 9; Section 3.1 diagnosis            | pre-freeze exploration        |
| $S = 512, r = 0.60$ (frozen)        | Tables 6, 10, 8, 12, 18, 21, 20; Figs. 7, 12, 13 | frozen before the TEST run    |
| Frozen point, transferred unchanged | Tables 5, 15, 17                                 | transfer check (no re-tuning) |
| Sweep over $S \times r$             | Table 13, Fig. 11                                | controlled comparison         |
| ASAH-style adaptive grid            | Table 14                                         | orthogonality check           |

**Operating points** Two operating points appear in this paper, and every table states which one it uses (Table 3). Ablations predating the final method were run at the SAHI default; everything from the method’s freeze onwards uses the frozen point. Every qualitative conclusion (agreement helps, WBF hurts, learning is useless) holds in the same direction at both.

**Evaluation** Official pycocotools COCOeval<sup>24</sup>. `maxDets` is 500 for VisDrone, 2,000 for DOTA (up to 1,814 GT per image) and 10,000 for the DOTA difficult-inclusive variant (up to 8,609); in each configuration we verified that feeding the ground truth as predictions returns  $AP = 100.00$ . The evaluator ranks and truncates *per image and category*; ranks quoted in this document and in the main text are positions in the full per-image–class list, before that truncation.  $AP_s$  follows the COCO definition (area  $< 32^2$ ) and is never compared across tables with different protocols (see the caveat in Section 3.13).

**Statistics** Image-level paired bootstrap with 10,000 resamples (seed 1234)<sup>25</sup>. The primary endpoint is a single pre-registered metric: *the  $AP_s$  difference between CAR and SAHI on TEST, with a 95% CI lower bound above zero*. Secondary metrics ( $AP$ ,  $AP_m$ ,  $AP_l$ ) use Holm correction<sup>26</sup>. Every row of the portability and generalisation tables also carries a 95% CI from the same procedure (Tables 12, 15).

**Baselines** B0: single-scale full-image inference. B2: SAHI + class-wise NMS (the primary comparison; it uses *the same views and the same detection set* as CAR, so the difference is attributable purely to the merge rule). B4: multi-view + WBF.<sup>1</sup> Section 3.7 adds Soft-NMS (linear, Gaussian), Weighted-NMS and Greedy-NMM (IoU, IOS) under identical conditions. Label-using references (Block B) and OVD (Block C) are reported in separate tables.

### 3 Full experimental analyses

#### 3.1 Oracle decomposition: where is the bottleneck?

Before designing the method, we computed four upper bounds from the detection dump alone on half of DEV (274 images,  $S = 640/r = 0.25$  grid, YOLOv8s).

An auxiliary bound: the **recall ceiling O1** is 83.4% (fraction of GT covered at IoU 0.5 by the union of multi-view boxes; 79.0% small, 92.6% medium, 94.1% large).

Three conclusions determined the design. (i) *Recall is not the bottleneck* — the boxes are already in the pool. (ii) *View selection has little room here* — on this benchmark the “where to crop” line of work has almost nothing left to gain, though we show below that this part of the decomposition is benchmark-dependent. (iii) *Ranking accounts for 74% of the bottleneck* — +14.97 AP is available without changing a single box.

The split is also *metric-stable*, which matters because the headroom is quoted in AP while small objects are the target of the paper: measured in  $AP_s$  the ranking share is likewise 74% (13.47 of 18.17, Table 4). The diagnosis therefore does not depend on which of the two metrics one privileges.

<sup>1</sup>The indices B0–B4 come from the pre-registered design matrix and two are absent here: B1 (image-pyramid test-time augmentation) was specified but never run, and B3 (SAHI’s NMM / Greedy-NMM merge) is reported as part of the merge-rule battery in Section 3.7 rather than as a standalone baseline. The original indices are retained so that the manuscript matches the released code and decision log.

**Table 4.** Oracle decomposition (DEV subset, 274 images).

|                                                  | Condition                | AP / AP <sub>s</sub> | Definition                                                 |
|--------------------------------------------------|--------------------------|----------------------|------------------------------------------------------------|
| B0                                               | single scale             | 21.99 / 11.99        | one full-image pass                                        |
| B2                                               | SAHI + NMS               | 26.40 / 18.78        | primary comparison                                         |
| O3                                               | view oracle              | 26.45 / 18.82        | best view per GT, scores unchanged                         |
| O4                                               | fusion oracle            | 36.68 / 27.29        | one best box per GT                                        |
| O2b                                              | ranking oracle           | 41.38 / 32.25        | <b>B2's box set unchanged</b> , scores $\leftarrow$ GT-IoU |
| O2                                               | score oracle             | 46.75 / 36.95        | scores $\leftarrow$ GT-IoU, then re-merge                  |
| <i>Decomposition of the recoverable headroom</i> |                          |                      |                                                            |
|                                                  | ranking only (O2b – B2)  | +14.97 AP            | <b>74%</b>                                                 |
|                                                  | box selection (O2 – O2b) | +5.37 AP             | 26%                                                        |
|                                                  | view selection (O3 – B2) | +0.05 AP             | $\approx 0\%$                                              |

**Table 5.** The oracle decomposition repeated on a second dataset and detector (DOTA-v1.5 val, 458 images, YOLO11s-OBB, frozen operating point), computed from stored detections only. The ranking share replicates; the view-selection share does not.

| Component                | VisDrone (274 img, YOLOv8s) |               | DOTA-v1.5 (458 img, YOLO11s-OBB) |              |
|--------------------------|-----------------------------|---------------|----------------------------------|--------------|
|                          | $\Delta$ AP                 | share         | $\Delta$ AP                      | share        |
| Ranking only (O2b – B2)  | +14.97                      | <b>74%</b>    | +14.52                           | <b>74.0%</b> |
| Box selection (O2 – O2b) | +5.37                       | 26%           | +5.11                            | 26.0%        |
| Total (O2 – B2)          | +20.35                      | 100%          | +19.63                           | 100%         |
| View selection (O3 – B2) | +0.05                       | $\approx 0\%$ | +4.67                            | <b>23.8%</b> |
| Recall ceiling O1        | 83.4%                       |               | 84.7%                            |              |

**Does the decomposition replicate?** The numbers above come from one dataset, one detector and one operating point. We repeated the identical decomposition on DOTA-v1.5 (458 images, YOLO11s-OBB, frozen operating point) using only the stored detections — no new forward passes (Table 5). Two components reproduce closely and one does not. **The ranking share reproduces almost exactly:** 74.0% in AP against 74% on VisDrone, with box selection likewise at 26.0% against 26%; the recall ceiling is also comparable (84.7% vs. 83.4%). **View selection does not reproduce:** negligible on VisDrone (+0.05 AP), it is worth +4.67 AP on DOTA, i.e. 23.8% of the headroom. DOTA images are up to an order of magnitude larger and yield far more views per image, so which view happens to frame an object well varies much more.

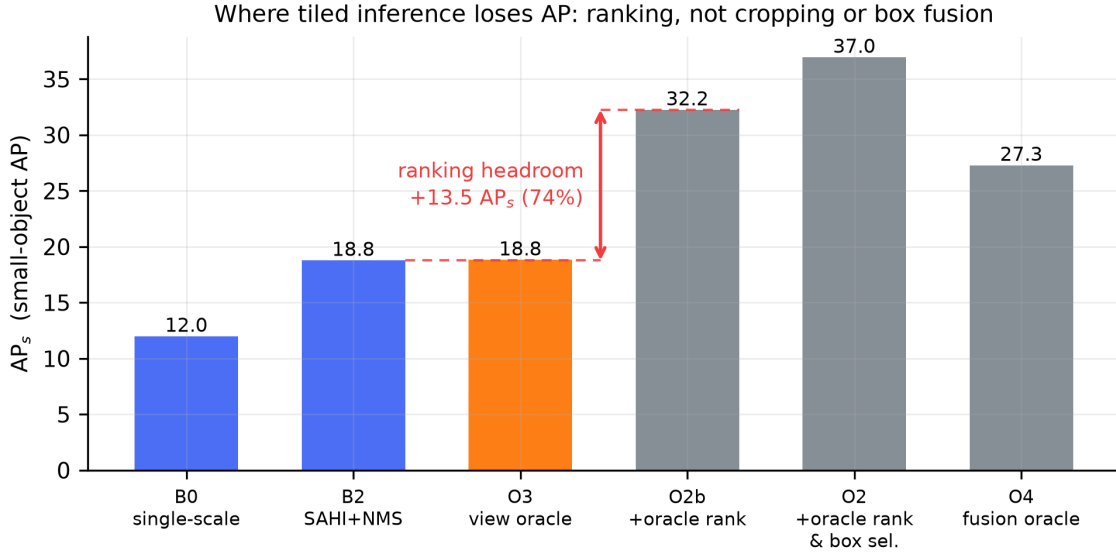
We therefore state the diagnosis in the form the evidence supports: *the dominance of ranking is stable across datasets, whereas the headroom available to view selection is not*. On very large images the “where to crop” line of work retains a real, if smaller, target. This does not alter the design of CAR, which operates on whatever view set it is given — and Section 3.12 shows that the two axes compose rather than compete.

Observation 1 ( $\sigma$ -conditionality) was tested on the same dump: holding GT-IoU quality fixed, the ratio of median scores across  $\sigma$  reaches 1.68 in the lowest-quality bin but is already 1.03–1.04 in the high-quality bins (Appendix 4.5). The conditional misalignment is real but *concentrated in the low-quality region* — and what separates true from false positives there is not a rescaling of the score axis but new information (confirmed by the failure of calibration in Section 3.6). Agreement is that information (Figure 12).

### 3.2 Main result: VisDrone test-dev

Figure 3 shows the effect qualitatively before the aggregate numbers: under a fixed detection budget, tiling recovers objects single-pass inference misses, and CAR then replaces high-ranked background errors with re-detected small objects. Figure 4 repeats the exercise on held-out test-dev scenes, where no selection or tuning decision ever looked.

**Primary endpoint:**  $\Delta$ AP<sub>s</sub> = +1.05, 95% CI [+0.89, +1.23],  $p < 10^{-4}$  (paired bootstrap, 10,000 resamples) — the



**Figure 1.** Oracle decomposition of the AP lost by tiled inference (DEV subset, 274 images, YOLOv8s,  $S = 640/r = 0.25$ ; bars are  $AP_s$ ). Replacing only the scores (O2b, B2’s box set untouched) recovers 74% of the headroom, while the view oracle (O3) is indistinguishable from the baseline. The split is metric-stable: the ranking share is 74% whether measured in AP (14.97 of 20.35) or in  $AP_s$  (13.47 of 18.17). The bottleneck is ranking, not cropping.

**Table 6.** Main result on TEST (VisDrone test-dev, 1,610 images; Block A, training-free). Speed measured on an RTX 3070 Ti.

| Condition         | AP           | AP <sub>50</sub> | AP <sub>75</sub> | AP <sub>s</sub> | AP <sub>m</sub> | AP <sub>l</sub> | ms/img | FPS  |
|-------------------|--------------|------------------|------------------|-----------------|-----------------|-----------------|--------|------|
| B0 single scale   | 16.07        | 27.45            | 16.54            | 7.10            | 25.19           | 36.19           | —      | —    |
| B2 SAHI + NMS     | 20.83        | 36.25            | 20.87            | 13.53           | 28.15           | 34.41           | 131.8  | 7.58 |
| B4 WBF fusion     | 19.49        | 34.59            | 19.78            | 12.62           | 26.16           | 30.84           | —      | —    |
| <b>CAR (ours)</b> | <b>22.53</b> | <b>39.02</b>     | <b>22.71</b>     | <b>14.58</b>    | <b>30.30</b>    | <b>36.42</b>    | 150.5  | 6.64 |

CI lower bound lies well above zero, satisfying the pre-registered criterion. After Holm correction the secondary metrics  $AP + 1.69$  (CI  $[+1.53, +1.85]$ ) and  $AP_m + 2.15$  (CI  $[+1.87, +2.44]$ ) were significant, while  $AP_l + 2.01$  was not ( $p = 0.088$ ; CI  $[-0.26, +5.07]$ ) — consistent with the nature of the method, since large objects have small  $|C(d)|$  and thus a weak agreement signal.

WBF was again worse than SAHI on the independent test split ( $-1.34$  AP,  $-0.91$   $AP_s$ ), reconfirming that the commonly used coordinate averaging is damaging rather than helpful in the multi-view small-object regime.

**The same comparison, extended to every detector** The pre-registered endpoint above concerns one detector. Extending the identical protocol — frozen operating point, no re-tuning — to all 25 detector configurations of the grand grid on test-dev, CAR improves *every* row: 25/25 on AP (median  $+1.51$ ) and 25/25 on  $AP_s$  (median  $+0.93$ ). VisDrone test-dev annotations are public and the split is designated by the benchmark for general evaluation, so this extension involves no protocol violation; the row list was committed before execution.

**Reading val and test-dev side by side** The absolute gains on test-dev are about  $0.6\times$  their val counterparts (median ratio 0.61 for AP, 0.58 for  $AP_s$ ). This is not tuning optimism evaporating: the *relative* gains nearly coincide ( $+11.2\%$  of the tiled baseline on val vs.  $+9.4\%$  on test-dev for AP;  $+9.9\%$  vs.  $+8.6\%$  for  $AP_s$ ). Test-dev is simply a harder split — the same detectors score 5–7 AP lower before CAR is applied — so the same proportional improvement is worth fewer absolute points. Where the baseline finds less, a re-ranking has less to re-rank.

### 3.3 Component ablation

At the SAHI default operating point ( $S = 640$ ,  $r = 0.25$ , DEV 548 images) we isolated each component of the merge rule.



no detector training · no labels · no extra forward passes — the same  $\lambda$  transfers to 12 detectors in 3 families, 2 datasets and 2 open-vocabulary models

**Figure 2.** CAR overview, on a real DEV image with real detections. **(1)** The views are exactly SAHI’s — one full image and 18 overlapping tiles here — and the frozen detector is run on them once, producing 5,592 boxes. Nothing is added to this stage. **(2)** One detection  $d$ , as the individual views see it. Four views contain  $d$  with the required margin and therefore form  $C(d)$ ; three of them re-detect it, giving  $\text{agr}(d) = 3/4$ . The greyed view is discarded from the denominator because  $d$  touches its crop edge — counting it would penalise  $d$  for a view that never had a fair look. **(3)** The score is multiplied by  $1 + \lambda \text{agr}$  and the usual class-wise NMS follows. A correct detection at score 0.10 that every view re-found climbs 1,485 ranks; a vendor’s cart read as a truck at score 0.89 that no view reproduced falls 477.

Reading the table: (i) agreement weighting alone contributes +1.10 and is the only substantive component; (ii) WBF coordinate averaging is  $-0.86$  on its own and, even combined with agreement, remains worse than agreement alone ( $+0.23 < +1.10$ ) — *leaving the coordinates untouched is best (NR2)*; (iii) noisy-OR justifies the choice in Eq. (4) that confirmation of reproduction, not probability accumulation, is what matters (NR3).

*Hyper-parameter sensitivity* (all on DEV, frozen operating point):  $\text{AP}_s$  is 22.08–22.20 for  $\lambda \in [2, 5]$  (a span of 0.12), a wide plateau; even  $\lambda = 0.5$  yields +1.07 (measured in the single-parameter sweep, at  $\tau_a = 0.5$ ).  $\tau_a$  gives 21.99–22.20 over  $[0.6, 0.75]$  and the margin  $m$  gives 21.49–21.50 over  $[0.25, 1.0]$ , i.e. effectively insensitive. The one sensitive choice is the denominator of Section 3.4.

### 3.4 Ablating $C(d)$ : an enabling condition

At the frozen operating point we varied only the denominator of the agreement.

Replacing the denominator with the total view count pushes the method *below* the baseline (19.69 vs. 20.07  $\text{AP}_s$ ): the moment the silence of views that could not have seen the object is counted as disagreement, noise buries the signal. The gap to the proposed definition is **+2.51  $\text{AP}_s$**  (95% CI  $[+2.08, +2.76]$ ,  $p < 10^{-4}$ ; in AP, +3.52 with CI  $[+3.09, +3.78]$ ), measured with the same paired bootstrap used for the primary endpoint. This  $|M|/|V|$  variant is structurally the same as transplanting WBF’s cluster-size rescaling (denominator = total member count) into tiled inference — *an agreement normalisation that works for ensembles is harmful under tiling, and visibility normalisation  $C(d)$  reverses it*. Excluding the detection’s own view is also slightly worse; the definition as stated is best.

### 3.5 Hyper-parameter sensitivity

All sensitivity checks use the development split only (VisDrone validation, YOLOv8s), varying one knob at a time around the frozen setting ( $\lambda=5$ ,  $\tau_a=0.75$ ,  $m=0.5$ ); Supplementary Fig. 5 shows the curves. The gain saturates immediately in  $\lambda$ : relative to no rescoring ( $\lambda=0$ , AP 25.85),  $\lambda=1$  already recovers 89% of the full gain and every  $\lambda \in [2, 10]$  lies within 0.15 AP of the

**Table 7.** Component ablation of the merge rule. VisDrone DEV (548 images), YOLOv8s, pre-freeze operating point ( $S = 640$ ,  $r = 0.25$ ); identical detections throughout, only the merge rule changes.

| Variant                                                            | AP           | AP <sub>s</sub> | $\Delta$ AP vs. B2 |
|--------------------------------------------------------------------|--------------|-----------------|--------------------|
| B2 SAHI + NMS                                                      | 25.76        | 18.65           | —                  |
| <b>agreement weighting + NMS (= CAR, <math>\lambda = 2</math>)</b> | <b>26.86</b> | <b>19.47</b>    | +1.10              |
| WBF coordinate averaging (B4)                                      | 24.90        | 17.86           | −0.86              |
| WBF + agreement weighting ( $\lambda = 2$ )                        | 25.99        | 18.68           | +0.23              |
| noisy-OR aggregation                                               | 23.97        | 17.36           | −1.79              |

**Table 8.** Effective view set ablation. VisDrone DEV (548 images), YOLOv8s, frozen operating point; identical detections, only the denominator of the agreement changes. The gap between the proposed definition and the ensemble-style one carries a paired-bootstrap 95% CI (10,000 resamples, seed 1234).

| Variant                                              | AP           | AP <sub>s</sub> |
|------------------------------------------------------|--------------|-----------------|
| B2 SAHI + NMS                                        | 25.85        | 20.07           |
| agr = $ M / C $ ( <b>ours</b> )                      | <b>28.74</b> | <b>22.20</b>    |
| agr = $ M / V $ (divide by all views)                | 25.22        | <b>19.69</b>    |
| agr = $ M / C $ , excluding the detection’s own view | 28.31        | 21.86           |

best value (28.62–28.77) — the rescoring acts through rank, so the exact multiplier barely matters. The matching threshold and the eligibility margin are similarly benign:  $\tau_a \in [0.6, 0.8]$  spans 0.33 AP and  $m \in [0.25, 0.75]$  spans 0.13 AP. No knob was ever re-tuned per dataset, per detector or per merge rule.

### 3.6 What do labels buy? The futility of recalibration

If the bottleneck is ranking, the natural approach is to *calibrate* the score: learn a layer mapping scores to hit probability, conditioned on magnification and size bins. We tested this systematically across three levels of label access and two model classes; all were rejected.

The GBDT is a small model ( $\leq 100$  trees) over 13 features: score, magnification, size, geometry and agreement statistics. We observe:

1. **True labels open a margin of +0.87 AP<sub>s</sub>** (19.52 vs. 18.65), of which label-free agreement weighting already recovers **94%** (+0.82). What the learned layer adds *on top of* agreement weighting is +0.24 AP / **+0.05 AP<sub>s</sub>**.
2. Training the same layer on pseudo-labels to stay label-free (agreement thresholds, with leakage controlled by a view split) is in fact *harmful*, at −0.14 AP<sub>s</sub>.
3. Replacing the score with the calibrated probability costs 1.37 AP relative to weighting — the loss of ranking information dominates the calibration gain.
4. Both remain below 5% of the margin opened by the oracle (+17.22 AP<sub>s</sub>): the residual headroom lies outside the feature set of score, geometry and agreement. This is a problem of *information*, not of labels.

**The same holds under a different model class** To answer the objection that “GBDT was the wrong choice”, we repeated the question with  $(\sigma \times a_{\text{net}})$ -binned isotonic regression (twelve bins, one regressor each, fitted in sample) at the frozen operating point. Even fitted on true ground truth it reaches only 26.17 AP / 19.85 AP<sub>s</sub> against B2’s 25.85 / 20.07 — a gain of +0.32 AP but a *loss of* −0.22 AP<sub>s</sub> on the small-object metric — and when combined with agreement (28.61 / 21.81) it remains worse than agreement alone (28.74 / 22.20). We conclude that the  $\sigma$ -conditional misalignment of Section 3.1 is real but *is not a signal recoverable by learnable rescaling*; what opens the bottleneck is the additional information carried by agreement (NR1). Figure 6 summarises the three label tiers against the oracle ceiling.

### 3.7 Comparison against all standard merge rules

To answer the objection that “NMS is merely a weak baseline”, we replaced the merge rule alone on *an identical detection set* and compared against every standard rule (frozen operating point, DEV 548 images, 663,268 detections). The Greedy-NMM rows are produced by calling the SAHI library’s own `GreedyNMMPostprocess` directly on our detection pool, so they are

**Table 9.** Label tiers. VisDrone DEV (548 images), YOLOv8s, pre-freeze operating point ( $S = 640$ ,  $r = 0.25$ ); the rescoring layer and its features are identical across rows and only label access changes. Blocks are kept separate: A is training-free, B uses labels.

| Block                    | Condition                                       | AP    | AP <sub>s</sub> |
|--------------------------|-------------------------------------------------|-------|-----------------|
| —                        | B2 SAHI + NMS                                   | 25.76 | 18.65           |
| <b>A (training-free)</b> | <b>CAR (agreement only, zero labels)</b>        | 26.86 | <b>19.47</b>    |
| A (training-free)        | GBDT rescoring + agreement pseudo-labels        | 25.49 | 18.51           |
| B (labels used)          | GBDT rescoring + <i>true labels</i> (weighting) | 27.10 | 19.52           |
| B (labels used)          | GBDT rescoring + true labels (replacement)      | 25.73 | 18.06           |
| Ceiling                  | score oracle (GT-IoU substituted)               | 44.27 | 35.87           |

not subject to any reimplement error.<sup>2</sup> We additionally include the rule of the concurrent TA-TM<sup>8</sup> — additive, applied only to detections within 16 px of a tile boundary,  $\lambda = 0.2$ , with  $A$  the maximum confidence among matching neighbours — transplanted onto the same detections.

Two findings emerge (Figure 7). First, **CAR leads the strongest alternative (Soft-NMS Gaussian) by +1.22 AP<sub>s</sub> / +1.69 AP** — rules that refine suppression while preserving the original score ordering (the Soft-NMS family) cannot open the ranking bottleneck of Section 3.1. Second, **SAHI’s default merge, Greedy-NMM, is 4.2–6.1 AP<sub>s</sub> worse than NMS** on dense small objects, because merging a cluster into its envelope swallows adjacent instances into one large box (NR4). The grand grid of Section 4.6 shows this is not a quirk of one detector: across all 59 (dataset, detector) configurations the NMS-to-Greedy-NMM gap has median 7.6 AP and reaches 16.4 (DOTA, dense oriented scenes) and 16.1 (AI-TOD with RT-DETR). The mechanism follows from the matching metric. SAHI’s slicing path defaults to intersection-over-smaller, under which any box fully contained in a larger one matches at IOS = 1 regardless of relative size; each merge replaces the group by its envelope, which is larger still and therefore swallows further neighbours on the next greedy step. The denser the scene, the longer the cascade — which is exactly why the collapse is worst on the densest benchmarks. Our reimplement is, if anything, generous: calling SAHI’s own `GreedyNMMPostprocess` on the same detections scores *lower* than our version at the same settings (16.54 vs. 17.89 AP; IOS 0.5), so the reported gaps are conservative. For practitioners applying SAHI to crowded scenes this is itself an actionable finding: prefer NMS over the default, and CAR over NMS. Section 3.8 asks whether the strongest of these rules can be combined with CAR rather than replaced by it.

**Geometry-aware suppression does not open the bottleneck either** ASahi replaces standard NMS with Cluster-DIoU-NMS, which subtracts a centre-distance penalty  $\rho^2/c^2$  from the IoU before thresholding<sup>4,27</sup>; the *cluster* part is a parallelisation of the same rule and returns identical boxes<sup>28</sup>, so Diou-NMS is a faithful stand-in. On dense small objects it lands at 19.99 AP<sub>s</sub>, marginally *below* plain NMS: for objects a few pixels across the competing centres are close,  $\rho^2/c^2 \rightarrow 0$ , and the rule degenerates to NMS while what remains of the penalty spares duplicates more often than it spares genuine neighbours. Applying it after CAR changes nothing (22.20  $\rightarrow$  22.17). Together with Soft-NMS (Section 3.8) and Greedy-NMM this completes a pattern across three mechanically distinct families: decayed suppression *absorbs* part of the effect, envelope merging is *unable* to express it, and geometric suppression is simply *unrelated* to it. **No rule that preserves the original score ordering escapes the neighbourhood of the NMS baseline**; the ranking loss of Section 3.1 is reachable only by changing the order itself.

**The additive, boundary-limited form does not transfer** Transplanting TA-TM’s rule onto the same detections costs 3.60 AP (1.61 AP<sub>s</sub>) relative to NMS. We do not read this as a defect of that method, which targets a different regime — PCB boundary artefacts, where its own baseline sits at 72–94 mAP@50 — but as a measurement of what the functional form can and cannot do here. Only 20.4% of our detections lie within 16 px of a tile boundary, so the rule never sees four fifths of the pool; and because it is additive with  $A \geq 0$  it can only *raise* scores, which promotes boundary false positives while leaving untouched the high-confidence interior errors that dominate the ranking loss (Section 3.1). This is the empirical counterpart of the functional-form argument in Section 1.1.

<sup>2</sup>We had first written an independent reimplement of the rule. It agrees with the library to within 0.19 AP (0.15 AP<sub>s</sub>); the small residual comes from the library re-validating the overlap against the *growing* merged box, whereas the reimplement merges a matched group into its envelope in one step. All numbers reported here are the library’s.

**Table 10.** Merge-rule battery. Identical detections; only the merge rule changes (DEV, frozen operating point). “SAHI default” means the settings that `get_sliced_prediction` actually applies — GREEDYNMM with match metric IOS at threshold 0.5, merging a matched pair into its union box and keeping the higher score, class-aware. This differs from the library’s class-level default (IOU), so we report both metrics and the comparison does not depend on which default is meant. Our reimplementation applies the rule to all views at once rather than view by view, as stated in the text.

| Merge rule                     | AP           | AP <sub>s</sub> | Note                          |
|--------------------------------|--------------|-----------------|-------------------------------|
| Greedy-NMM (IOS 0.5)           | 17.92        | 13.99           | <b>SAHI default merge</b>     |
| Greedy-NMM (IoU 0.5)           | 21.40        | 15.83           |                               |
| TA-TM rule <sup>8</sup>        | 22.25        | 18.46           | additive, boundary-limited    |
| NMS (B2)                       | 25.85        | 20.07           | SAHI common practice baseline |
| Soft-NMS linear <sup>6</sup>   | 26.58        | 20.54           |                               |
| Weighted-NMS (box voting)      | 26.19        | 20.57           | coordinate voting only        |
| Soft-NMS Gaussian <sup>6</sup> | 27.05        | 20.98           | <b>strongest alternative</b>  |
| DIoU-NMS <sup>27</sup>         | 25.72        | 19.99           | = ASahi’s Cluster-DIoU-NMS    |
| + CAR                          | 28.74        | 22.17           | no interaction                |
| <b>CAR (ours)</b>              | <b>28.74</b> | <b>22.20</b>    | $\lambda = 5.0$               |

**Is WBF harmful only at one setting?** Our claim that coordinate fusion hurts small objects (NR2) rested on a single configuration, so we swept it: cluster IoU  $\in \{0.30, 0.45, 0.55, 0.65, 0.75\} \times$  score rule  $\in \{\text{max, average, WBF’s cluster-size rescaling}\} \times \{\text{fuse coordinates, keep the top-scoring box}\}$ , thirty settings on identical detections. Twenty-nine fall below the NMS baseline; the single exception (+0.25 AP<sub>s</sub>) is cluster IoU 0.75 with max scoring *and fused coordinates* — the setting at which clusters are so tight that almost nothing is actually merged. The sweep separates the two halves of the method, and the coordinate half turns out to be dose-dependent: averaging coordinates rather than keeping the top-scoring box costs 2.59 AP<sub>s</sub> at cluster IoU 0.30, 0.95 at 0.55, and *gains* 0.35 at 0.75 (mean +0.62 over the fifteen matched pairs). Loosening the clusters is what makes fusion harmful, which is the same statement as the exception above. Averaging *scores* is uniformly worse (−2.16 mean versus −0.94 for max). WBF’s own cluster-size rescaling is the worst of the three (−2.27 mean), which is the point of contact with our design: it divides by a nominal number of models, whereas in tiled inference the cluster size is not a count of independent opinions but a function of *where the object happens to sit relative to the tile grid*. That is exactly the quantity our denominator  $|C(d)|$  normalises away (Section 2.1) — the same raw material — −2.27 AP<sub>s</sub> averaged over its ten settings against +2.13 at our frozen operating point — separated by which denominator is used.

**Practical guidance for SAHI users on dense small objects** (i) Replace the default Greedy-NMM merge with plain NMS (+4.2 to +6.1 AP<sub>s</sub>, Table 10). (ii) Replace NMS with CAR (+1.18 to +2.13 AP<sub>s</sub> across all eight operating points of Table 13; CPU-only, no extra forward passes). (iii) Do not increase tile overlap under NMS — it costs computation and slightly *reduces* AP<sub>s</sub> — whereas under CAR the same overlap buys accuracy (Figure 11). (iv) Avoid WBF coordinate averaging for small objects (−0.43 to −1.91 AP<sub>s</sub> across detector families, Section 3.10).

### 3.8 Does the gain stack with Soft-NMS?

Table 10 ranks Soft-NMS as the strongest standard alternative, and it acts at a different point of the pipeline than CAR does: CAR rewrites scores *before* merging, Soft-NMS softens suppression *during* it. The natural question is whether the two compose. We ran the full  $2 \times 3$  grid — raw score  $s$  versus CAR score  $\hat{s}$ , under NMS, Soft-NMS linear and Soft-NMS gaussian, with the Soft-NMS hyper-parameters exactly as in Table 10, not re-tuned for the combined conditions (Figure 8a).

Read column-wise, CAR contributes under every suppression rule (+2.13, +1.88, +1.26 AP<sub>s</sub>). Read row-wise, the converse fails: most of what Soft-NMS adds on raw scores vanishes once CAR has been applied. The aggressive gaussian variant — the stronger of the two on raw scores, +0.91 AP<sub>s</sub> — retains +0.05 (5%); the threshold-gated linear variant retains +0.22 of its +0.47 (47%), and that remainder is real (paired bootstrap, 10,000 resamples: +0.22 AP<sub>s</sub> [+0.18, +0.27], +0.32 AP

**Table 11.** CAR on top of CMA-NMS<sup>29</sup>. Fraction of detector×view pairings where CMA-NMS+CAR > CMA-NMS (AP); per benchmark, both view generators pooled.

| Benchmark         | Improved / total | %   |
|-------------------|------------------|-----|
| VisDrone-val      | 50/50            | 100 |
| VisDrone test-dev | 50/50            | 100 |
| DOTA-v1.5-val     | 20/20            | 100 |
| AI-TOD-v2-val     | 43/49            | 88  |
| AI-TOD-v2-test    | 47/50            | 94  |
| Total             | 210/219          | 96  |

$[+0.28, +0.37]$ , both  $p < 10^{-4}$ ).

The recovery sets say why (Figure 8b). Counting the ground-truth objects each method newly brings into the top- $K_c$  region relative to SAHI: Soft-NMS gaussian recovers 1,094, CAR recovers 2,069 — and **77.8% of the former (851) are objects CAR recovers anyway**. Only 243 objects (0.6% of ground truth) are exclusive to Soft-NMS, which caps what any combination could add; the actual gaussian combination recovers fewer objects than the union of the two (2,003 vs. 2,312) while losing more against the SAHI baseline ( $-784$  vs.  $-606$ ). The two mechanisms are therefore not orthogonal but largely redundant: both repair the same failure — duplicate boxes of one object mishandled at the merge — one by demoting mis-localised duplicates *before* suppression (CAR), the other by decaying rather than deleting *during* it (Soft-NMS). Once the ranking is fixed upstream, a global re-ordering like the gaussian decay has nothing left to do. The linear decay keeps a small local niche: its exclusive recoveries are smaller than average (median 18.0 px) and concentrated in the densest pedestrian regions, exactly where a hard threshold over-suppresses.

This is the merge-stage counterpart of Section 3.12: at the *view-generation* stage CAR composes with adaptive slicing; at the *merge* stage it largely subsumes softened suppression. We do not adopt the  $+0.22$ . The operating point, including plain NMS, was frozen before the single TEST run (Section 2.6), and changing the merge rule afterwards would detach the reported numbers from the reported method. Practitioners free of that constraint can take the linear variant as a small, cheap add-on.

### 3.9 Does the gain survive a crop-aware merge? CMA-NMS

The strongest version of the orthogonality question uses the most recent merge rule we are aware of: Crop Margin Aware NMS (CMA-NMS), the merging stage of DAHI<sup>29</sup>. Unlike the rules of Sections 3.7–3.8, CMA-NMS is *crop-aware*: a detection failing a margin test (its box not fully inside its crop shrunk by  $\theta=10$  px) is treated as potentially truncated, and against detections from *other* crops the suppression test switches from  $\text{IoU} > 0.65$  to the stricter intersection-over-smaller  $\text{IoS} > 0.9$ . We re-implemented the rule from Algorithm 2 of the original paper with its published defaults and verified the direction of its reported gains before wiring it into our battery.

CMA-NMS is a good merge rule on its own: on our smoke configuration it recovers  $+0.9$  AP over plain NMS. The question is whether CAR still adds gain *on top*. Table 11 says yes, everywhere: across 219 detector×view pairings on five benchmarks, rescoring before CMA-NMS improves AP in 210 (96%), with every VisDrone and DOTA pairing improving. Two further observations matter. First, the combination is not redundant: CMA+CAR exceeds *both* of its parents. Second, CAR levels the playing field between merge rules — the median AP of CMA+CAR (20.64) and NMS+CAR (20.63) are indistinguishable, although CMA-NMS and NMS differ without rescoring. This is exactly the signature predicted by the ranking account: once the scores are repaired, the choice of suppression geometry matters far less.

### 3.10 Portability: thirteen detectors across three families and two capacity ladders, no re-tuning

Keeping  $\lambda$  and  $\tau_d$  exactly as chosen on YOLOv8s, we varied only the detector (DEV). Table 12 is organised as a before/after comparison: for each detector we report the tiled baseline (B2) and the same detections after CAR, grouped by family so that the *capacity ladder* within a family can be read directly. Two families are swept end to end (YOLOv8 and YOLO11, nano through extra-large); the DETR and two-stage families anchor the architectural axis. Each  $\Delta$  carries a 95% CI from the same paired bootstrap (10,000 resamples).

**All thirteen improve,  $\Delta\text{AP}_s$  from  $+1.43$  to  $+2.35$  (mean  $+1.85$ ), with every per-row CI lower bound  $\geq +1.12$**  (Figure 9); in AP the mean is  $+2.71$  with every lower bound  $\geq +1.98$ . Three axes are covered simultaneously: *capacity* (2.6M to 68.2M, a factor of 26), *generation* (YOLOv8 and YOLO11 swept in parallel) and *family* (one-stage CNN, DETR set prediction, two-stage proposal refinement).

Two properties are visible only because the ladders are complete. First, *the gain does not decay as the detector strengthens*: within YOLO11 it rises monotonically with capacity ( $+1.58 \rightarrow +2.07$   $\text{AP}_s$  from n to x) even though the baseline itself climbs

**Table 12.** Detector portability, as a before/after comparison. Hyper-parameters fixed at the YOLOv8s values; *no per-detector re-tuning*. Families are grouped, and within a family models are ordered by capacity, so that the effect of scaling the detector can be read down each block. Every row improves and every  $\Delta AP_s$  interval excludes zero.

| Family                        | Model          | Params (M) | AP    |              |                      | AP <sub>s</sub> |              |                      |
|-------------------------------|----------------|------------|-------|--------------|----------------------|-----------------|--------------|----------------------|
|                               |                |            | B2    | CAR          | $\Delta$ [95% CI]    | B2              | CAR          | $\Delta$ [95% CI]    |
| YOLOv8<br>one-stage CNN       | YOLOv8n        | 3.2        | 22.20 | <b>24.69</b> | +2.49 [+2.14, +2.78] | 16.84           | <b>18.27</b> | +1.43 [+1.12, +1.75] |
|                               | YOLOv8s        | 11.2       | 25.85 | <b>28.74</b> | +2.89 [+2.58, +3.20] | 20.07           | <b>22.20</b> | +2.13 [+1.78, +2.43] |
|                               | YOLOv8m        | 25.9       | 29.18 | <b>31.59</b> | +2.41 [+2.16, +2.68] | 23.17           | <b>24.93</b> | +1.76 [+1.46, +2.08] |
|                               | YOLOv8l        | 43.7       | 29.53 | <b>32.54</b> | +3.01 [+2.61, +3.34] | 24.08           | <b>25.92</b> | +1.84 [+1.53, +2.18] |
|                               | YOLOv8x        | 68.2       | 29.83 | <b>32.90</b> | +3.07 [+2.69, +3.36] | 23.70           | <b>26.05</b> | +2.35 [+1.96, +2.71] |
| YOLO11<br>one-stage CNN       | YOLO11n        | 2.6        | 22.69 | <b>25.37</b> | +2.68 [+2.30, +2.95] | 17.48           | <b>19.06</b> | +1.58 [+1.33, +1.87] |
|                               | YOLO11s        | 9.4        | 27.56 | <b>29.81</b> | +2.25 [+1.98, +2.50] | 21.55           | <b>23.07</b> | +1.52 [+1.28, +1.81] |
|                               | YOLO11m        | 20.1       | 30.59 | <b>33.18</b> | +2.59 [+2.34, +2.83] | 23.98           | <b>25.86</b> | +1.88 [+1.61, +2.19] |
|                               | YOLO11l        | 25.3       | 31.37 | <b>34.07</b> | +2.70 [+2.39, +2.98] | 25.02           | <b>27.02</b> | +2.00 [+1.71, +2.38] |
|                               | YOLO11x        | 56.9       | 32.37 | <b>34.84</b> | +2.47 [+2.19, +2.70] | 26.11           | <b>28.18</b> | +2.07 [+1.76, +2.34] |
| RT-DETR<br>DETR set pred.     | RT-DETR-l      | 32.8       | 28.82 | <b>31.81</b> | +3.00 [+2.67, +3.29] | 22.88           | <b>24.68</b> | +1.79 [+1.45, +2.13] |
|                               | RT-DETR-x      | 67.3       | 29.63 | <b>32.77</b> | +3.13 [+2.76, +3.44] | 23.82           | <b>25.57</b> | +1.75 [+1.49, +2.04] |
| <b>Faster R-CNN</b> two-stage | <b>R50-FPN</b> | 41.4       | 25.39 | <b>27.95</b> | +2.56 [+2.29, +2.82] | 18.36           | <b>20.33</b> | +1.97 [+1.69, +2.29] |

from 17.5 to 26.1 AP<sub>s</sub>. Second, *the two generations track each other at matched capacity* — nano +1.43 vs. +1.58, large +1.84 vs. +2.00 — although YOLO11 is the stronger baseline throughout. Both observations point the same way: what CAR repairs is not a deficiency of a particular model but a deficiency of *the tiled-inference procedure itself*, which every detector inherits equally. The harmfulness of WBF likewise crosses all three families:  $-1.91$  AP<sub>s</sub> on RT-DETR and  $-0.43$  on Faster R-CNN.

*Note (self-trained subjects).* No public VisDrone checkpoints exist for RT-DETR-l or Faster R-CNN, so we trained them ourselves (Appendices 4.1, 4.2; the three community DETR checkpoints we verified all scored AP  $\approx 2$  and were unusable). Training used the train split only, and for Faster R-CNN we *eliminated model selection entirely* by taking the last checkpoint of a fixed 12-epoch schedule (val untouched). CAR does not modify either set of weights. *Note (RT-DETR candidate budget).* DETR-style models emit exactly `num_queries` = 300 candidates per view (2,840,400 detections / 9,468 views = 300.0). This is an architectural constant, and since B2 and CAR share the same candidate set it introduces no bias into  $\Delta AP_s$ . *Note (checkpoint provenance).* Within each YOLO ladder all five checkpoints come from a single publisher trained under one recipe (YOLOv8: mshamrai; YOLO11: erbayat), so capacity varies while the training protocol is held fixed; SHA-256 hashes are listed in the released manifest. *Note (exclusion).* The YOLOv5s community checkpoint uses a legacy serialisation format that could not be loaded, and was excluded.

**Tiling is not free for the DETR family — and CAR makes it safe** The before/after framing hides a case where the *baseline itself* is the problem. In two of the 59 grand-grid configurations — and only these two — tiled inference lands *below* the single-pass detector: RT-DETR-l and RT-DETR-x on VisDrone (in-domain), where overall AP drops from 29.25  $\rightarrow$  28.82 and 30.55  $\rightarrow$  29.63 respectively even as AP<sub>s</sub> improves (20.13  $\rightarrow$  22.88, 21.93  $\rightarrow$  23.82). Tiling recovers small objects but floods the pool with cross-tile duplicates and boundary fragments whose set-prediction confidences are not comparable across views; the extra false positives outrank enough true detections to give back more than the small-object gain. CAR repairs the ordering rather than the boxes, and that is enough to flip the sign: 31.81 and 32.77 AP — above single-pass by +2.56 and +2.22 — while pushing AP<sub>s</sub> further still (24.68, 25.57). Figure 10 shows the same effect qualitatively. Crucially, the phenomenon *replicates on the held-out split*: on VisDrone test-dev (1,610 images never used for any tuning decision) the same two configurations again land below single-pass under tiling (23.06  $\rightarrow$  22.90 and 24.10  $\rightarrow$  23.70) and again finish above it with CAR (25.06, 25.83 — +2.00 and +1.73 over single-pass). What was an observation on the development split is therefore a reproducible property of the architecture, not an artefact of image selection. The practical reading: for NMS-free architectures, cross-view agreement is not merely an enhancement to tiled inference but the condition under which tiling is worthwhile at all.

### 3.11 The overlap–gain curve

Using the frozen values, we measured eight operating points over slice size  $\times$  overlap ratio. *In each row the two methods use the same view set and the same detections* — only the merge rule differs.

Three things are visible at once (Figure 11). (1) **CAR dominates SAHI at all eight operating points** — it is always higher at equal compute. (2) **At fixed  $S$ , increasing overlap makes SAHI worse** (18.65  $\rightarrow$  18.44; 20.38  $\rightarrow$  20.07). NMS discards the

**Table 13.** Operating-point sweep (DEV). “agr levels” is the number of distinct agreement values, i.e. the resolution of the signal.

| $S$        | $r$         | views/img | rel. FLOPs | agr levels | B2 AP <sub>s</sub> | CAR AP <sub>s</sub> | $\Delta$ AP <sub>s</sub> | $\Delta$ AP |
|------------|-------------|-----------|------------|------------|--------------------|---------------------|--------------------------|-------------|
| 640        | 0.25        | 6.2       | 1.00       | 9          | 18.65              | 19.82               | +1.18                    | +1.60       |
| 640        | 0.40        | 6.4       | 1.04       | 21         | 18.61              | 19.81               | +1.20                    | +1.57       |
| 640        | 0.50        | 7.9       | 1.28       | 17         | 18.44              | 19.85               | +1.41                    | +1.84       |
| 640        | 0.60        | 8.2       | 1.33       | 21         | 18.49              | 19.90               | +1.41                    | +1.86       |
| 512        | 0.25        | 8.8       | 1.42       | 17         | 20.23              | 21.43               | +1.20                    | +1.92       |
| 512        | 0.40        | 8.9       | 1.44       | 17         | 20.38              | 21.66               | +1.27                    | +2.17       |
| 512        | 0.50        | 10.7      | 1.74       | 21         | 20.10              | 21.67               | +1.57                    | +2.34       |
| <b>512</b> | <b>0.60</b> | 17.3      | 2.79       | <b>39</b>  | 20.07              | <b>22.20</b>        | +2.13                    | +2.89       |

**Table 14.** Adaptive slicing + CAR (DEV).

| View generation                     | Merge      | views/img | AP           | AP <sub>s</sub> |
|-------------------------------------|------------|-----------|--------------|-----------------|
| ASAHI-style adaptive grid (reimpl.) | NMS        | 7.1       | 26.76        | 20.20           |
| ASAHI-style adaptive grid (reimpl.) | <b>CAR</b> | 7.1       | <b>28.46</b> | <b>21.40</b>    |

duplicates that the extra overlap creates and gains only extra false positives; for a SAHI user, additional overlap is wasted computation. (3) **CAR’s gain grows with overlap** — as the number of distinct agreement levels rises from 9 to 39,  $\Delta$ AP<sub>s</sub> grows from +1.18 to +2.13. This is the quantification of Observation 2: the same redundancy is waste to NMS and evidence to CAR.

### 3.12 Orthogonality: CAR on top of adaptive slicing

Our claim is that the relationship to the “where to crop” line of work is composition, not competition (Section 1.1; O3  $\approx$  0 on VisDrone-scale images, Section 3.1). To demonstrate this, we reimplemented the principle of ASahi<sup>4</sup> — a grid whose tile count adapts to resolution (6 or 12 tiles) — replaced the view generator *entirely*, and applied CAR on top.  $\lambda$  and  $\tau_a$  remain at the frozen values.

$\Delta + 1.70$  AP /  $+1.20$  AP<sub>s</sub>. Two side observations support the orthogonality claim. First, with only 7.1 views the adaptive grid reaches a *higher* baseline (26.76 AP) than the fixed SAHI grid at 17.3 views (25.85 AP, last row of Table 13) — improving the cropping is genuinely a valid axis. Second, on top of that improved cropping CAR still adds a comparable margin — improving the merging is a *separate* valid axis. The two do not overlap; they accumulate.

### 3.13 Generalisation: DOTA-v1.5 without re-tuning

We changed the dataset (drone to satellite nadir), the detector (YOLO11-OBb trained on DOTA-v1.0, with oriented boxes converted to axis-aligned envelopes) and the class system (16 classes), keeping  $\lambda$  and  $\tau_a$  at the VisDrone values. The standard DOTA-v1.5 protocol excludes 35,557 objects marked *difficult*, whose median  $\sqrt{\text{area}}$  is 13.4 px — i.e. the standard evaluation does not count the smallest objects. We therefore report both protocols, changing *only the evaluation convention* on identical detections.

**Oriented-box convention** Following DOTA’s horizontal-bounding-box protocol, every oriented annotation and every oriented detection is reduced to the axis-aligned envelope of its four corners,  $[\min_i x_i, \min_i y_i, \max_i x_i, \max_i y_i]$ ; no minimum-area rectangle fitting and no angle information is used downstream. Every geometric operation in CAR is therefore axis-aligned:  $C(d)$  tests containment of that envelope in the view rectangle with margin  $ma_{\text{orig}}$  ( $m = 0.5$ , waived on sides coinciding with the image border);  $\text{agr}(d)$  counts a view  $u \in C(d)$  as agreeing when it holds at least one same-class detection whose axis-aligned IoU with  $d$  is at least  $\tau_a = 0.75$ , each view contributing at most once and the view that produced  $d$  agreeing by construction; and the merge is class-wise NMS at IoU 0.6. The only rotated-geometry operation anywhere in the pipeline is the detector’s own per-view suppression, performed by Ultralytics before the envelope is taken.

This convention is applied identically to every condition, so the paired comparison is unaffected — but it is not neutral in absolute terms. On DOTA-v1.5 val the envelope has a median  $1.76\times$  the area of the oriented box over all 69,565 annotations (ship  $2.06\times$ , large-vehicle  $2.12\times$ , harbor  $1.79\times$ ; median oriented aspect ratio 2.28; under the difficult-excluding standard

**Table 15.** DOTA-v1.5 val under both evaluation protocols.

| Protocol (eval. GT / median $\sqrt{\text{area}}$ ) | Detector    | B2 AP | CAR AP       | $\Delta\text{AP}$ [95% CI] |
|----------------------------------------------------|-------------|-------|--------------|----------------------------|
| standard (34,008 / 33.4 px)                        | YOLO11s-obb | 43.36 | <b>46.12</b> | +2.76 [+1.95, +3.75]       |
| standard (34,008 / 33.4 px)                        | YOLO11m-obb | 44.93 | <b>47.40</b> | +2.47 [+1.72, +3.45]       |
| <b>difficult-inclusive</b> (69,565 / 19.9 px)      | YOLO11s-obb | 39.76 | <b>42.83</b> | +3.07 [+2.35, +3.91]       |
| <b>difficult-inclusive</b> (69,565 / 19.9 px)      | YOLO11m-obb | 41.25 | <b>44.27</b> | +3.02 [+2.33, +3.86]       |

protocol these become  $1.70\times$  and  $2.42$ ), so at a fixed threshold axis-aligned IoU is more permissive than oriented IoU for elongated instances, and the size bucket of a box is assigned from envelope rather than oriented area. We therefore read DOTA as  $\Delta\text{AP}$  under a fixed convention and do not compare its absolute values with oriented-box literature.

The gain is if anything *larger* than on VisDrone and obtained without any re-tuning — the strongest evidence for hyper-parameter transfer — and **all four rows have CI lower bounds above +1.7**. Furthermore, as the task becomes harder the baseline drops (mean  $44.1 \rightarrow 40.5$ ) while **the gain grows** (mean  $+2.62 \rightarrow +3.04$ ), in the same direction for both detectors.

*Caveat: DOTA claims are made on  $\Delta\text{AP}$  only.* The “small” ( $< 32^2$ ) bucket of  $\text{AP}_s$  denotes different populations under the two protocols (the difficult-inclusive setting admits a large mass of essentially undetectable 13 px objects), and with only 458 images the small-subgroup intervals are wide and unstable (e.g. standard/11s:  $\Delta\text{AP}_s + 2.29 [+0.64, +4.76]$ ). Difficult-inclusive numbers must also not be compared against standard-protocol figures in the literature.

### 3.14 Generalisation: AI-TOD-v2, twelve detectors, no re-tuning

DOTA changes the sensor and the class system but keeps the images very large. AI-TOD-v2<sup>30,31</sup> changes the opposite axis: its images are  $800 \times 800$  and its objects have a median  $\sqrt{\text{area}}$  of 11.8 px, roughly half of VisDrone’s. It is the standard benchmark for tiny-object detection, and it is the setting in which our own scaling argument predicts the *least* headroom, because the number of views a tile grid can produce falls with image size.

**Why the geometry survives** That prediction turns out to depend entirely on the operating point. At SAHI’s defaults ( $S = 640$ ,  $r = 0.25$ ) an  $800 \times 800$  image yields five views; at our frozen point ( $S = 512$ ,  $r = 0.6$ ) the stride is 205 px and it yields ten. Ground truth measured through the same  $C(d)$  code gives a median  $|C(d)|$  of 4 on AI-TOD-v2 val — identical to VisDrone — with 84.7% of boxes seen by three or more views. Small objects in fact help here: the margin  $m_{\text{orig}}$  scales with object size, so a 12 px object clears the tile boundary test that a 50 px object would fail.

**Reconstruction of the benchmark** AI-TOD does not distribute its images; they must be synthesised from the xView training set, and the publicly downloadable subset omits that portion (66.7% of val images, 38.5% of annotations, with the vehicle class dropping from 85.6% to 63.6% of instances). We obtained xView and rebuilt the full benchmark without the official toolkit, whose dependencies no longer build: the tiling convention is recoverable from the file names (`{source}__{x}_{y}.png` is an  $800 \times 800$  crop at  $(x, y)$ , zero-padded at the borders), which we confirmed against the non-xView tiles whose sources we already hold — normalised correlation 0.998 at scale 1.0 at the position the file name specifies — and then verified on coordinates, comparing our tiles’ annotations against the xView labels translated by  $(x, y)$  (median  $L_1$  distance 1.0 px; v2 is a re-annotation of v1, so exact agreement is not expected). The result matches the official counts exactly: 2,804 val images with 75,082 boxes and 11,214 train images with 301,494 boxes.

**Detectors and evaluation** No public in-domain checkpoint exists, so we trained twelve — the same YOLO and RT-DETR ladders as Table 12 — with the recipe of Appendix 4.1 unchanged, varying only the dataset. CAR is then applied with  $\lambda$ ,  $\tau_a$ ,  $S$  and  $r$  at their VisDrone values. AI-TOD defines its own size buckets, and they are *not* COCO’s:  $\text{AP}_{\text{vt}}$  is  $[0, 8^2)$ ,  $\text{AP}_t$  is  $[8^2, 16^2)$  and its  $\text{AP}_s$  is  $[16^2, 32^2)$  where COCO’s is everything below  $32^2$ . We therefore headline  $\text{AP}_{\text{vt}}$  and  $\text{AP}_t$  and never place these columns beside the  $\text{AP}_s$  of Tables 6 and 12. Evaluation uses the area ranges and  $\text{maxDets} = 1500$  of the official `aitodpycocotools`; we verified that our wrapper reproduces that implementation exactly, on an oracle input and on a perturbed one, across all five buckets.

### 3.15 Held-out confirmation: AI-TOD-v2 test

All AI-TOD numbers so far were computed on the validation split, which — like every development split in this paper — was visible during method freezing even though no hyper-parameter was tuned on it. The AI-TOD-v2 test split (14,018 images, 376,121 annotated instances, roughly five times the size of the validation split) provides a stronger check: it was never touched

**Table 16. AI-TOD-v2, twelve detectors, no re-tuning.** Frozen operating point transferred unchanged from VisDrone. AI-TOD’s own size buckets<sup>31</sup> differ from COCO’s:  $AP_{vt}$  is  $[0, 8^2)$  and  $AP_t$  is  $[8^2, 16^2)$ , so these columns are *not* comparable with the  $AP_s$  of Tables 6 and 12. Evaluated with the official `aitodpycocotools` area ranges and `maxDets=1500`.

| Detector  | $AP_{vt}$ |              | $AP_t$ |              | AP    |              |
|-----------|-----------|--------------|--------|--------------|-------|--------------|
|           | B2        | CAR          | B2     | CAR          | B2    | CAR          |
| YOLOv8n   | 10.03     | <b>9.91</b>  | 19.33  | <b>19.96</b> | 20.79 | <b>21.44</b> |
| YOLOv8s   | 11.27     | <b>11.89</b> | 23.78  | <b>24.28</b> | 25.10 | <b>25.76</b> |
| YOLOv8m   | 13.18     | <b>13.82</b> | 26.13  | <b>26.84</b> | 27.64 | <b>28.24</b> |
| YOLOv8l   | 13.79     | <b>14.60</b> | 26.28  | <b>26.95</b> | 27.95 | <b>28.72</b> |
| YOLOv8x   | 12.55     | <b>12.89</b> | 27.50  | <b>28.47</b> | 28.86 | <b>29.66</b> |
| YOLO11n   | 9.16      | <b>9.96</b>  | 19.59  | <b>20.57</b> | 20.48 | <b>21.42</b> |
| YOLO11s   | 10.74     | <b>12.45</b> | 23.68  | <b>24.28</b> | 25.10 | <b>25.71</b> |
| YOLO11m   | 13.38     | <b>14.47</b> | 26.11  | <b>27.10</b> | 27.23 | <b>28.17</b> |
| YOLO11l   | 11.21     | <b>12.61</b> | 26.24  | <b>26.89</b> | 27.96 | <b>28.61</b> |
| YOLO11x   | 13.13     | <b>13.99</b> | 28.94  | <b>29.72</b> | 29.92 | <b>30.79</b> |
| RT-DETR-l | 6.76      | <b>7.51</b>  | 21.56  | <b>22.45</b> | 23.26 | <b>23.96</b> |
| RT-DETR-x | 8.74      | <b>9.58</b>  | 22.38  | <b>23.25</b> | 24.51 | <b>25.40</b> |

**Table 17.** Open-vocabulary detection on VisDrone val (Block C).

| Detector                          | Design                         | SAHI AP | SAHI $AP_s$ | +CAR AP | +CAR $AP_s$ | $\Delta AP_s$ (rel.) |
|-----------------------------------|--------------------------------|---------|-------------|---------|-------------|----------------------|
| YOLO-World-S <sup>20</sup>        | CNN + text embedding alignment | 9.00    | 4.52        | 11.03   | <b>5.85</b> | +1.33 (+29%)         |
| Grounding DINO-tiny <sup>21</sup> | DETR + cross-modal fusion      | 9.25    | 4.45        | 10.53   | <b>5.48</b> | +1.03 (+23%)         |

until the final evaluation reported here, and every number in this subsection comes from a single run with the globally frozen setting.

The result is a clean replication. All 25 detector configurations improve in AP when CAR is added to SAHI+NMS (25/25; validation: 24/24). The median relative gain is +7.2% of baseline AP against +5.6% on validation, and +10.5% against +9.2% in the very-tiny band  $AP_{vt}$  — the held-out gains are, if anything, marginally *larger*, so nothing in the validation numbers was an artefact of split choice. This mirrors the VisDrone val→test-dev pattern of Section 3.2: absolute deltas shrink with the harder split, relative gains are preserved. The in-domain RT-DETR pair also behaves as on validation (l: 22.47 → 23.10, x: 21.89 → 22.43 AP).

### 3.16 Open-vocabulary detection: where fine-tuning is impossible

Class names were supplied purely as text prompts (no weight modification), and  $\lambda$ ,  $\tau_a$  again retained their VisDrone YOLO values. Since OVD baselines are far below in-domain trained models — as expected — we report them in a separate block and make no cross-block comparison of absolute values.

The method works consistently on two OVD models of fundamentally different design — this is not the coincidence of a single model. This block substantiates the scope declaration of this paper: *in a setting where the test-time class set is unknown at training time, fine-tuning cannot enter in principle*. OVD scores are notoriously miscalibrated with respect to prompt and object size, whereas agreement is indifferent to the score scale and alters only the ordering, which is why it applies naturally here.

**Scope limit: what the method cannot do** Applying the same YOLO-World to DOTA (satellite nadir) yields a baseline of 0.63 AP, which does not constitute a working detector at all (even a class-agnostic evaluation gives only 1.30 — localisation itself collapses). An OVD trained on ground-level natural images does not know aerial-specific categories. **CAR improves the ranking of ability a detector already has; it does not create ability that is absent.** Stating this failure case delimits the claim.

### 3.17 Why it works: discriminability, mechanism, qualitative cases

Figure 12 tests the premise of the method directly: fixing the detector score by quantile and measuring precision by agreement within each quantile, **detections with  $\text{agr} \geq 0.5$  are roughly twice as precise in every score quantile** (84% vs. 42% in the top quantile). Agreement therefore carries information the score does not — an axis orthogonal to the score, which is why reweighting can genuinely improve the ordering.

**Table 18.** Mechanism, quantified (DEV). Definitions in Appendix 4.6.

| Quantity                                                | before ( $s$ ) | after ( $\hat{s}$ ) |
|---------------------------------------------------------|----------------|---------------------|
| precision of the top- $K_c$ region (all classes pooled) | 89.6%          | <b>91.0%</b>        |
| false positives pushed out of the top region            | —              | <b>1,246</b>        |
| true positives pulled into the top region               | —              | <b>4,484</b>        |
| classes improved in post-merge AP                       | —              | <b>10 / 10</b>      |

**Table 19.** What agreement demotes (DEV, YOLOv8s). All detections with  $s > 0.35$ ,  $\text{agr} \leq 0.2$ ,  $|C(d)| \geq 2$ , assigned to exactly one category in the listed priority order. “Better box exists” means another detection in the pre-merge pool covers the same ground-truth object at a strictly higher IoU.

| What the demoted detection actually is                                    | count      | share       |
|---------------------------------------------------------------------------|------------|-------------|
| inside an ignore region (absorbed by the evaluator)                       | 402        | 2.4%        |
| background: no ground truth within IoU 0.1                                | 902        | 5.4%        |
| covers an object of a <i>different</i> class at $\text{IoU} \geq 0.5$     | 1,609      | 9.5%        |
| same class, loose box ( $\text{IoU} 0.3\text{--}0.5$ )                    | 4,398      | 26.1%       |
| same class, $\text{IoU} \geq 0.5$ , but a better box exists               | 5,068      | 30.1%       |
| grazes an object, matches nothing at $\text{IoU} 0.5$                     | 4,173      | 24.8%       |
| <b>same class, <math>\text{IoU} \geq 0.5</math>, no better box exists</b> | <b>306</b> | <b>1.8%</b> |

We also quantified what the reweighting actually moves (Table 18). Comparing, in the pre-merge detection pool, the top- $K_c$  region (where  $K_c$  is the number of ground-truth objects of class  $c$ ; an R-precision construction) before and after rescoring:

Post-merge, per-class AP improves for *all ten classes*, with the largest gains on the larger vehicle categories (bus +8.49, car +6.00, truck +5.12, van +4.02) — consistent with the interpretation in Section 2.1 that a larger number of valid repeated measurements  $|C(d)|$  yields a stronger agreement signal. Section 3.18 takes this apart class by class and shows that the size of the gain is governed by how well agreement separates true from false detections within each class.

**What exactly does agreement demote?** Reporting that low-agreement detections are “false positives” would be imprecise, so we audited every one of them. On DEV there are 16,858 detections with score  $> 0.35$ ,  $\text{agr} \leq 0.2$  and  $|C(d)| \geq 2$  — the population the rescoring pushes down. Assigning each to exactly one category (Table 19) shows that they are of three broad kinds: outright background errors, detections that overlap something but match no ground-truth box at  $\text{IoU} 0.5$ , and *redundant* boxes for objects that another view has already localised better. Only 306 detections — 1.82% — are the uniquely best box for their object, i.e. the only cases where demotion could cost a detection at all. This is the mechanical reason the reweighting raises AP instead of trading recall for precision: it suppresses a population that is 98.2% redundant or invalid.

Figure 13 shows twelve cases, three per row. Rescued true positives (row 1) are correct low-confidence detections re-found by every view that could see them: one at score 0.10 with  $\text{agr} = 1.0$  is lifted from rank 4,592 to 3,107 and thereby recovered. Row 2 shows unambiguous background errors — a street vendor’s cart read as a truck, a striped traffic barrel as a tricycle, a traffic-light head as a car — all at high confidence and all reproduced by no view. Row 3 shows the redundancy case: the demoted box is on a real object, but a better-localised box (dashed) survives, so the object is not lost; the third panel is the rare failure mode in which the demoted box is the only good one. Row 4 shows what the method cannot reach at all — the class confusions of Section 3.18, where the box is right and the label is wrong. Crucially, none of this is a tile-boundary phenomenon; it occurs across the whole image, which is the substantive difference from approaches that treat boundary detections only (TA-TM<sup>8</sup>).

### 3.18 Per-class analysis: which bottleneck limits which class

Per-class AP improves for *all ten classes* (Table 20), but the spread is wide — from +0.58 on awning-tricycle to +8.49 on bus — and three classes remain below AP = 20. A reviewer is entitled to ask why those classes are low and whether rescoring is the right instrument for them. We answer by repeating the recall-versus-ranking decomposition of Section 3.1 along the class axis.

For each class we compute two ceilings of the pre-merge box pool on DEV, both at  $\text{IoU} \geq 0.5$  and both independent of any score:

- $R_{\text{agn}}$  — the fraction of ground-truth instances covered by *some* detection box, regardless of predicted class. This is the purely geometric retrieval ceiling.

**Table 20.** Per-class decomposition on DEV (YOLOv8s, frozen operating point), ordered by achieved AP. *size* is the median  $\sqrt{wh}$  of ground-truth boxes in pixels. *miss* =  $100 - R_{\text{agn}}$  (never boxed), *conf* =  $R_{\text{agn}} - R_{\text{cls}}$  (boxed, wrong class), both as a percentage of ground-truth instances; *head* =  $\text{AP}_{\text{O2}} - \text{AP}_{\text{CAR}}$  is the remaining ranking headroom on the same pool; *sep* is the agreement separation, the median agr of pool-stage true positives minus that of false positives. Only *head* is addressable by rescoring.

| Class           | $n_{\text{GT}}$ | size | B2 AP | CAR AP | $\Delta\text{AP}$ | miss | conf        | head | sep         |
|-----------------|-----------------|------|-------|--------|-------------------|------|-------------|------|-------------|
| bicycle         | 1,287           | 20.8 | 9.57  | 10.31  | +0.73             | 16.2 | 20.0        | 24.6 | 0.30        |
| awning-tricycle | 532             | 29.3 | 9.94  | 10.51  | +0.58             | 11.1 | <b>38.9</b> | 22.8 | 0.37        |
| people          | 5,125           | 16.7 | 17.03 | 17.79  | +0.76             | 13.0 | 13.5        | 20.9 | 0.29        |
| tricycle        | 1,045           | 28.4 | 18.14 | 18.97  | +0.83             | 9.3  | 20.7        | 26.7 | 0.37        |
| truck           | 750             | 38.6 | 24.00 | 29.11  | +5.12             | 8.4  | 20.8        | 22.1 | 0.47        |
| motor           | 4,886           | 20.0 | 25.47 | 26.77  | +1.30             | 5.6  | 8.0         | 23.5 | 0.40        |
| pedestrian      | 8,844           | 17.2 | 29.61 | 30.71  | +1.10             | 8.2  | 3.5         | 20.9 | 0.35        |
| van             | 1,975           | 33.6 | 31.52 | 35.54  | +4.02             | 2.5  | 29.3        | 17.8 | 0.67        |
| bus             | 251             | 35.2 | 38.33 | 46.81  | +8.49             | 4.0  | 14.8        | 15.3 | <b>0.75</b> |
| car             | 14,064          | 32.5 | 54.85 | 60.84  | +6.00             | 2.7  | 1.0         | 13.4 | 0.67        |

- $R_{\text{cls}}$  — the fraction covered by a detection of the *correct* class. The shortfall  $R_{\text{agn}} - R_{\text{cls}}$  is therefore attributable to classification, not to localisation.

This splits each class’s loss into three parts that fail for different reasons:  $100 - R_{\text{agn}}$  instances are never boxed at any view;  $R_{\text{agn}} - R_{\text{cls}}$  are boxed but labelled as another class; and what remains is a *ranking* shortfall, which we size as  $\text{AP}_{\text{O2}} - \text{AP}_{\text{CAR}}$  using the score oracle of Section 3.1 on the identical box pool. **Only the third is reachable by rescoring.** The first two are properties of the detector and of the class taxonomy, and no reordering of a fixed pool can recover them. The three quantities are diagnostic magnitudes on different scales (percentage of instances for the first two, AP for the third); they are not an additive budget and we do not treat them as one.

**The three weakest classes fail for three different reasons** *Awning-tricycle* has an unremarkable retrieval ceiling ( $R_{\text{agn}} = 88.9\%$ ) but loses 38.9 points to misclassification — the largest such loss in the dataset — principally to tricycle (13.3% of its instances) and to car (10.3%). Its problem is fine-grained category discrimination, not detection. *Bicycle* is the only class with a genuinely poor retrieval ceiling (83.8%, 16.2 points never boxed) and a 20.0-point classification loss, mostly to motor (9.6%) and pedestrian (7.7%): a thin, frequently occluded rider is both hard to localise and visually close to two other categories. *People* loses 13.5 points to confusion, of which 9.8% of all its instances go to pedestrian — a distinction (standing versus other posture) that is partly annotator-defined and does not correspond to a stable visual boundary at these object sizes. In all three cases the dominant loss is outside what a rescoring adapter can address, which is the honest reading of why the gains there are small.<sup>3</sup>

**Where CAR helps is predicted by the signal, not by the class** The last column of Table 20 measures how well agreement separates real objects from errors within a class. It tracks the realised gain closely (Figure 14b; Spearman  $\rho = 0.85$ ,  $p = 0.002$ ,  $n = 10$ ; Pearson  $r = 0.91$ ). The four classes with the largest gains — bus, car, truck, van — are precisely those with the highest separation (0.47–0.75); the four with the smallest gains have the lowest (0.29–0.37). This is the per-class counterpart of Figure 12 and it makes the scope of the method explicit: large, rigid objects that span several tiles are re-detected consistently and are therefore re-rankable, whereas small, thin or taxonomically ambiguous objects produce detections that flicker across views whether or not they are real. Where the signal is weak, CAR correctly does little — no class is harmed — but it also cannot manufacture evidence that the view ensemble did not provide. Section 3.19 completes this picture along the image axis.

### 3.19 How the gain distributes across images

An average can hide a lottery, so we scored every DEV image separately under B2 and CAR (YOLOv8s, frozen operating point). Of the 524 images on which  $\text{AP}_s$  is defined, 408 **improve** (77.9%), 17 **are unchanged** and 99 **degrade**; the median per-image change is +1.02  $\text{AP}_s$  and the 10th–90th percentile range is  $[-0.74, +4.64]$  (Figure 15a). Nor is the gain carried by a few outliers: the top decile of improving images contributes 35.5% of the total positive gain, the top quartile 61.0%.

The degradations have a specific and, on inspection, reassuring anatomy. The single worst image ( $-27.8 \text{ AP}_s$ ) is a nearly empty scene with seven annotated objects, of which CAR places five rather than six in the top region — one flipped detection

<sup>3</sup>Attribution of a confused instance to a competing class uses the class of the highest-IoU covering detection. This is a diagnostic attribution, not an exclusive assignment; an instance may be overlapped by detections of several classes.

**Table 21.** Computational cost per image.

| Stage                       | SAHI            | CAR                    |
|-----------------------------|-----------------|------------------------|
| forward passes (shared)     | 112.8 ms        | 112.8 ms               |
| agreement computation (CPU) | —               | +18.7 ms               |
| NMS                         | 19.1 ms         | 19.0 ms                |
| <b>total / FPS</b>          | 131.8 ms / 7.58 | 150.5 ms / <b>6.64</b> |

moves per-image  $AP_s$  by double digits. Binning all images by their number of *small* ground-truth objects (the actual support of  $AP_s$ ) makes the pattern monotone (Figure 15b): with 0–2 small objects no image improves and the standard deviation of the per-image change is 9.31; with more than 100, 88.9% improve and the deviation collapses to 1.69. On the 444 images (85%) with at least 16 small objects — the regime tiled inference exists for — 84.0% improve, the median gain is +1.19, and the worst single image loses 6.28. What remains in the sparse bins is not systematic damage but the quantisation of AP on a handful of instances.

Consistently with this reading, the per-image gain correlates with how much there is to re-rank — the number of annotated objects (Spearman  $\rho = +0.26$ ,  $p < 10^{-8}$ ) and the number of views ( $\rho = +0.17$ ) — and not at all with the image’s mean  $|C(d)|$  ( $\rho = -0.02$ ,  $p = 0.59$ ). Rescoring needs material to reorder, not more witnesses per box.

### 3.20 Computational cost

Table 21 reports the per-image cost measured on an RTX 3070 Ti at the frozen operating point (15.1 views/image).

Because the number of forward passes is identical, the overhead is +14.2% of *total* latency (150.5 vs. 131.8 ms), incurred entirely in CPU post-processing; the GPU cost is unchanged. A robustness analysis across image-size groups is given in Appendix 4.4 — the gain is flat at +1.5 to +2.1 across three size groups, and the controlled analysis of what drives the gain is the overlap sweep of Section 3.11.

### 3.21 Overhead across all benchmarks: bounded by detection density

The single-configuration timing above raises an obvious question: how large can the post-processing overhead become in the worst case? We timed the three CPU stages of CAR (agreement computation, rescoring, final NMS) on 26 benchmark×detector configurations — 120 sampled images each, median of three repeats, on a 32-core CPU host; the GPU forward cost is identical with and without CAR by construction and is therefore excluded.

The median overhead is +10.0 ms per image and the minimum +1.5 ms. The overhead tracks detection density rather than image size or detector family: the worst case is DOTA-v1.5 with YOLOv8s (1,131 detections per image after the score floor), at +139.5 ms per image. Two facts keep this bounded in practice. First, dense-output configurations are exactly the ones whose tiled GPU forward pass is also largest (DOTA images produce an order of magnitude more views than VisDrone), so the *relative* overhead grows far more slowly than the absolute one. Second, the agreement stage is embarrassingly parallel across images; our measurements are single-worker per image. For the headline configuration (VisDrone, YOLOv8s), the total cost of CAR remains +14.2% of end-to-end latency as reported above, and no configuration in the sweep changes the number of forward passes.

### 3.22 Relation to published numbers

A head-to-head comparison with related methods *does not exist*: detectors, datasets, splits and metrics all differ, and even where public implementations exist they are tailored to their own domains. To give the reader a sense of scale, we nonetheless collect the one quasi-comparable axis — the  $\Delta$  mAP@50 each paper reports against *its own* tiling baseline. Table 22 is *context*, not a *claim of results*.

**Confounders stated first** (i) TA-TM’s baselines sit at 72–94 mAP@50, a saturated regime where the remaining headroom is structurally small — no claim of superiority can be drawn from this comparison. (ii) Our baseline (36.3) is lower than ASAHI’s (55.1), because our detector is a general-purpose YOLOv8s rather than a VisDrone-specialised transformer head, and *the objection that a lower baseline yields a larger  $\Delta$  is legitimate*.

Our evidence against that objection is Table 12: scaling the detector 21-fold from YOLOv8n (B2 16.84  $AP_s$ ) to YOLOv8x (23.70) raises the baseline substantially yet the gain does not shrink (+1.43 → +2.35). *Our gain is therefore not a function of a weak baseline*. We nevertheless discourage direct comparison of the rows above; every performance claim in this paper rests on the controlled, identical-detection comparisons of Sections 3.2–3.17.

**Table 22.** Reported gains over each paper’s own tiling baseline. Rows are not directly comparable; see the confounders discussed in the text.

| Method                  | Data                     | Detector    | own baseline (mAP@50) | $\Delta$ mAP@50 | Training            |
|-------------------------|--------------------------|-------------|-----------------------|-----------------|---------------------|
| TA-TM <sup>8</sup>      | PCB-Defect               | YOLO12n     | 72.0                  | +0.2            | none                |
| TA-TM <sup>8</sup>      | HRIPCB                   | YOLO12n     | 93.5                  | +0.1            | none                |
| ASAHI <sup>4</sup>      | VisDrone val             | TPH-YOLOv5  | 55.1                  | +1.7            | <b>SAF required</b> |
| ViCrop-Det <sup>5</sup> | VisDrone/DOTA/COCO       | DETR family | —                     | +1 to +3        | none (DETR-only)    |
| <b>CAR (ours)</b>       | <b>VisDrone test-dev</b> | YOLOv8s     | 36.3                  | +2.8            | <b>none</b>         |
| <b>CAR (ours)</b>       | <b>DOTA-v1.5 val</b>     | YOLO11s-obb | 66.3                  | +4.8            | <b>none</b>         |

**Orthogonality, measured** ASAHI and ViCrop-Det change *where to crop*; CAR changes *what to trust*. They are therefore objects of composition rather than competition, and Section 3.12 showed +1.70 AP accumulating when CAR is placed on an ASAHI-style adaptive grid. The DOTA replication makes this point sharper: there the cropping axis has real headroom (23.8% of the total, Table 5), and CAR nevertheless improves on that dataset by +2.47 to +3.07 AP (Table 15). The two axes are therefore not substitutes — where cropping has room, merging still has its own.

## 4 Training details and reproduction

### 4.1 RT-DETR training details (evaluation subjects; not part of the method)

No public VisDrone DETR checkpoint exists — all three community checkpoints we verified scored AP  $\approx 2$  and were unusable — so we trained the two members of this family ourselves. Starting from the COCO-pretrained rtdetr-l and rtdetr-x<sup>32</sup>, **with an identical recipe for both**: VisDrone train (6,471 images), an 80-epoch schedule with patience 20, `imgsz` 640, batch 16, seed 1234, and `mosaic` = 0.0 (so that tile-like augmentation cannot contaminate the SAHI comparison). RT-DETR-l early-stopped at epoch 62 with val mAP50 47.4 / mAP50-95 28.1; RT-DETR-x at epoch 58 with 49.6 / 29.6 — both in the band of publicly reported figures, and ordered as capacity predicts. Training touched val only for the early-stopping decision and never touched test-dev in any form. CAR does not modify these weights. Weight SHA-256 values and the full recipes are released with the code.

Two honest caveats. The family ladder stops at two members because *Ultralytics distributes COCO-pretrained weights only for rtdetr-l and rtdetr-x*; the ResNet variants have no pretrained weights, and training them from scratch on 6,471 images would not produce a meaningful evaluation subject. And the two models were trained on different hardware (rtdetr-l on an NVIDIA A40, rtdetr-x on an L40S) with the same library version pinned; this affects wall-clock time, not the recipe.

### 4.2 Faster R-CNN training details (evaluation subject; not part of the method)

The two-stage detector likewise has no public VisDrone checkpoint, so we trained one. Starting from torchvision’s COCO-pretrained Faster R-CNN (ResNet-50 FPN) implementation<sup>33</sup>: VisDrone train (6,471 images), a **fixed 12-epoch standard**  $1\times$  **schedule** (SGD, lr 0.01, milestones 8/11, batch 8, `min_size` 800, seed 1234), with horizontal flipping as the only augmentation. We raised the guard one level above Appendix 4.1: **by taking the last checkpoint and thereby removing model selection altogether**, val plays no role in training whatsoever. Training took 77 minutes on an A40 and ended at a training loss of 0.72. Its SAHI baseline of 25.39 AP falls in the same band as the public YOLOv8s on the same data (25.85), evidence of a normally converged detector. VisDrone’s ignore regions were excluded from the targets because torchvision has no ignore semantics; we note that this leaves those regions to be learned as background. The weight SHA-256 and recipe are released with the code.

### 4.3 Reproduction information

- **Code**: the method itself is roughly 30 lines; separate modules implement  $C(d)$ /agreement and the bootstrap, and one runner script exists per experiment.
- **Frozen configuration**: fixed in a configuration file before the TEST run and unchanged thereafter.
- **Weights**: a manifest records the SHA-256, provenance and training-recipe audit of every checkpoint. Detector implementations are Ultralytics<sup>34</sup> and torchvision; evaluation is pycocotools<sup>24</sup>.
- **Single-use TEST**: a marker file locks the execution time, configuration and results. The test-dev numbers in this paper come from that single run.

- **Statistics:** image-level paired bootstrap, 10,000 resamples, seed 1234, Holm correction on secondary metrics. Implemented by re-aggregating pycocotools `evalImgs` without re-evaluating detections. The per-row CIs of Tables 12 and 15 follow the same procedure.
- **Decision log:** every design change, rejection and exception is recorded with its justification.

#### 4.4 Robustness across image-size groups

We checked whether the gain is stable across the three resolution groups that VisDrone val itself contains. Size groups are an *uncontrolled* comparison — flight altitude and scene content vary together with size — so they are not used to support any claim in the main text (correlation between view count and  $\Delta AP_s$  is  $+0.17$ ; the controlled analysis is Section 3.11). Table 23 reports the breakdown.

**Table 23.** Results by image-size group (DEV).

| Size               | Images | views/img | B2 $AP_s$ | CAR $AP_s$ | $\Delta AP_s$ |
|--------------------|--------|-----------|-----------|------------|---------------|
| $960 \times 540$   | 121    | 9         | 21.38     | 22.89      | +1.51         |
| $1360 \times 765$  | 408    | 19        | 19.68     | 21.82      | +2.13         |
| $1920 \times 1080$ | 19     | 33        | 18.55     | 20.23      | +1.68         |

The gain is flat at  $+1.5$  to  $+2.1$ . A side observation: SAHI’s  $AP_s$  is in fact *lower* on larger images ( $21.38 \rightarrow 18.55$ ) — the larger the image, the more duplicates there are to merge and the worse the ranking contamination becomes — while CAR’s gain is maintained.

#### 4.5 $\sigma$ -conditionality in detail

Holding the GT-IoU quantile fixed, we computed the ratio of maximum to minimum median score across magnification  $\sigma$  (Table 24).

**Table 24.** Score ratio across  $\sigma$  within GT-IoU quantiles.

| GT-IoU quantile | score ratio across $\sigma$ |
|-----------------|-----------------------------|
| $(0.50, 0.59]$  | <b>1.68</b>                 |
| $(0.59, 0.68]$  | 1.58                        |
| $(0.68, 0.78]$  | 1.15                        |
| $(0.78, 0.87]$  | 1.03                        |
| $(0.87, 1.00]$  | 1.04                        |

For detections of identical quality, the score differs by more than  $1.6\times$  in the low-quality region depending on the magnification at which they were observed, whereas the high-quality region is already aligned. This asymmetry is consistent with the result of Section 3.6: there is little to gain from a global rescaling of the score axis, and what is needed is additional information that separates true from false positives in the low-quality region.

#### 4.6 Mechanism: definitions and per-class detail

**Definition of the top region (Table 18)** In the pre-merge detection pool, a detection  $d$  is called a *true positive* if it overlaps a ground-truth object of the same class at  $\text{IoU} \geq 0.5$  (this is not an exclusive assignment — we note explicitly that this is a pool-stage definition in which duplicates exist). For class  $c$ , let  $K_c$  be the number of ground-truth objects of that class, and take the top  $K_c$  detections by score as the *top region* (the same construction as R-precision). Comparing the top region before ( $s$ ) and after ( $\hat{s}$ , Eq. (4)) rescoring gives the change in precision, the number of false positives pushed out and the number of true positives pulled in.

**Per-class detail** Post-merge per-class AP and its decomposition into retrieval, classification and ranking components are given in the main text (Table 20, Figure 14). The four largest gains (bus, car, truck, van) are all vehicle categories large enough to span several tiles — that is, categories with large  $|C(d)|$ . This does not contradict the non-significant  $AP_l$  on TEST (Section 3.2): in VisDrone most instances of these vehicle categories fall in the COCO small/medium buckets, and the large bucket is itself sparsely populated.

**Table 25. Merge rule  $\times$  detector, all 192 cells measured.** AP<sub>s</sub> on DEV (VisDrone val, 548 images) at the frozen operating point ( $S = 512$ ,  $r = 0.6$ ,  $\lambda = 5.0$ ), no per-cell tuning. Columns 1–4 are merge rules on raw scores; columns 5–8 apply CAR first and then the same rule. CAR improves 94 of the 96 pairs; the two exceptions (in-domain YOLOv8l  $-0.03$ , stock-COCO RT-DETR-x  $-0.02$ ) are both in the +NMM column, which is the informative negative: once the envelope merge has destroyed the box, fixing the score cannot recover it (median  $+0.22$  versus  $+1.76$  for NMS,  $+1.54$  for Soft-NMS linear and  $+0.88$  for Soft-NMS Gaussian — the last already largely absorbed, cf. Section 3.8). Stock-COCO rows use a 6-class mapping and are diagnostic only — read signs and relative sizes, not absolute values.

| Detector                                       | NMM   | NMS   | S-lin | S-gau | +NMM         | +NMS         | +lin         | +gau         | $\Delta(\text{NMS})$ |
|------------------------------------------------|-------|-------|-------|-------|--------------|--------------|--------------|--------------|----------------------|
| <i>In-domain (VisDrone-trained)</i>            |       |       |       |       |              |              |              |              |                      |
| YOLOv8n                                        | 10.65 | 16.84 | 17.19 | 17.59 | <b>10.81</b> | <b>18.27</b> | <b>18.41</b> | <b>18.38</b> | +1.43                |
| YOLOv8s                                        | 13.96 | 20.07 | 20.54 | 20.98 | <b>14.41</b> | <b>22.20</b> | <b>22.42</b> | <b>22.25</b> | +2.13                |
| YOLOv8m                                        | 16.63 | 23.17 | 23.78 | 24.12 | <b>16.75</b> | <b>24.93</b> | <b>25.19</b> | <b>25.00</b> | +1.76                |
| YOLOv8l                                        | 18.27 | 24.08 | 24.77 | 25.22 | <b>18.24</b> | <b>25.92</b> | <b>26.17</b> | <b>25.97</b> | +1.84                |
| YOLOv8x                                        | 18.52 | 23.70 | 24.41 | 24.95 | <b>18.84</b> | <b>26.05</b> | <b>26.33</b> | <b>26.14</b> | +2.35                |
| YOLO11n                                        | 10.39 | 17.48 | 17.97 | 18.33 | <b>10.66</b> | <b>19.06</b> | <b>19.28</b> | <b>19.17</b> | +1.58                |
| YOLO11s                                        | 14.44 | 21.55 | 22.12 | 22.47 | <b>14.54</b> | <b>23.07</b> | <b>23.35</b> | <b>23.21</b> | +1.52                |
| YOLO11m                                        | 17.13 | 23.98 | 24.56 | 25.01 | <b>17.32</b> | <b>25.86</b> | <b>26.11</b> | <b>25.92</b> | +1.88                |
| YOLO11l                                        | 19.13 | 25.02 | 25.69 | 26.26 | <b>19.34</b> | <b>27.02</b> | <b>27.29</b> | <b>27.10</b> | +2.00                |
| YOLO11x                                        | 20.15 | 26.11 | 26.77 | 27.32 | <b>20.40</b> | <b>28.18</b> | <b>28.50</b> | <b>28.28</b> | +2.07                |
| RT-DETR-l                                      | 15.48 | 22.88 | 23.20 | 23.94 | <b>15.51</b> | <b>24.68</b> | <b>24.74</b> | <b>24.72</b> | +1.80                |
| RT-DETR-x                                      | 16.41 | 23.82 | 24.10 | 24.74 | <b>16.42</b> | <b>25.57</b> | <b>25.68</b> | <b>25.63</b> | +1.75                |
| <i>No training at all (stock COCO weights)</i> |       |       |       |       |              |              |              |              |                      |
| YOLOv8n                                        | 2.41  | 4.41  | 4.54  | 4.68  | <b>2.44</b>  | <b>4.90</b>  | <b>4.98</b>  | <b>4.96</b>  | +0.49                |
| YOLOv8s                                        | 4.08  | 5.76  | 5.94  | 6.11  | <b>4.30</b>  | <b>6.91</b>  | <b>7.01</b>  | <b>6.95</b>  | +1.15                |
| YOLOv8m                                        | 5.52  | 6.39  | 6.61  | 6.92  | <b>5.92</b>  | <b>8.16</b>  | <b>8.26</b>  | <b>8.18</b>  | +1.77                |
| YOLOv8l                                        | 6.26  | 6.79  | 7.02  | 7.27  | <b>6.70</b>  | <b>8.54</b>  | <b>8.66</b>  | <b>8.54</b>  | +1.75                |
| YOLOv8x                                        | 6.68  | 6.73  | 6.93  | 7.23  | <b>7.32</b>  | <b>8.95</b>  | <b>9.06</b>  | <b>8.94</b>  | +2.22                |
| YOLO11n                                        | 2.58  | 4.14  | 4.27  | 4.51  | <b>2.72</b>  | <b>5.17</b>  | <b>5.27</b>  | <b>5.26</b>  | +1.03                |
| YOLO11s                                        | 4.05  | 5.38  | 5.56  | 5.80  | <b>4.29</b>  | <b>6.81</b>  | <b>6.93</b>  | <b>6.87</b>  | +1.43                |
| YOLO11m                                        | 5.95  | 6.26  | 6.46  | 6.70  | <b>6.34</b>  | <b>8.18</b>  | <b>8.28</b>  | <b>8.18</b>  | +1.92                |
| YOLO11l                                        | 6.32  | 6.61  | 6.82  | 7.13  | <b>6.84</b>  | <b>8.55</b>  | <b>8.65</b>  | <b>8.54</b>  | +1.94                |
| YOLO11x                                        | 6.84  | 6.75  | 6.96  | 7.15  | <b>7.33</b>  | <b>8.64</b>  | <b>8.74</b>  | <b>8.62</b>  | +1.89                |
| RT-DETR-l                                      | 5.39  | 7.95  | 8.07  | 8.25  | <b>5.47</b>  | <b>8.40</b>  | <b>8.42</b>  | <b>8.45</b>  | +0.45                |
| RT-DETR-x                                      | 6.08  | 8.23  | 8.35  | 8.58  | <b>6.06</b>  | <b>8.59</b>  | <b>8.62</b>  | <b>8.65</b>  | +0.36                |

### Full merge-rule $\times$ detector grid

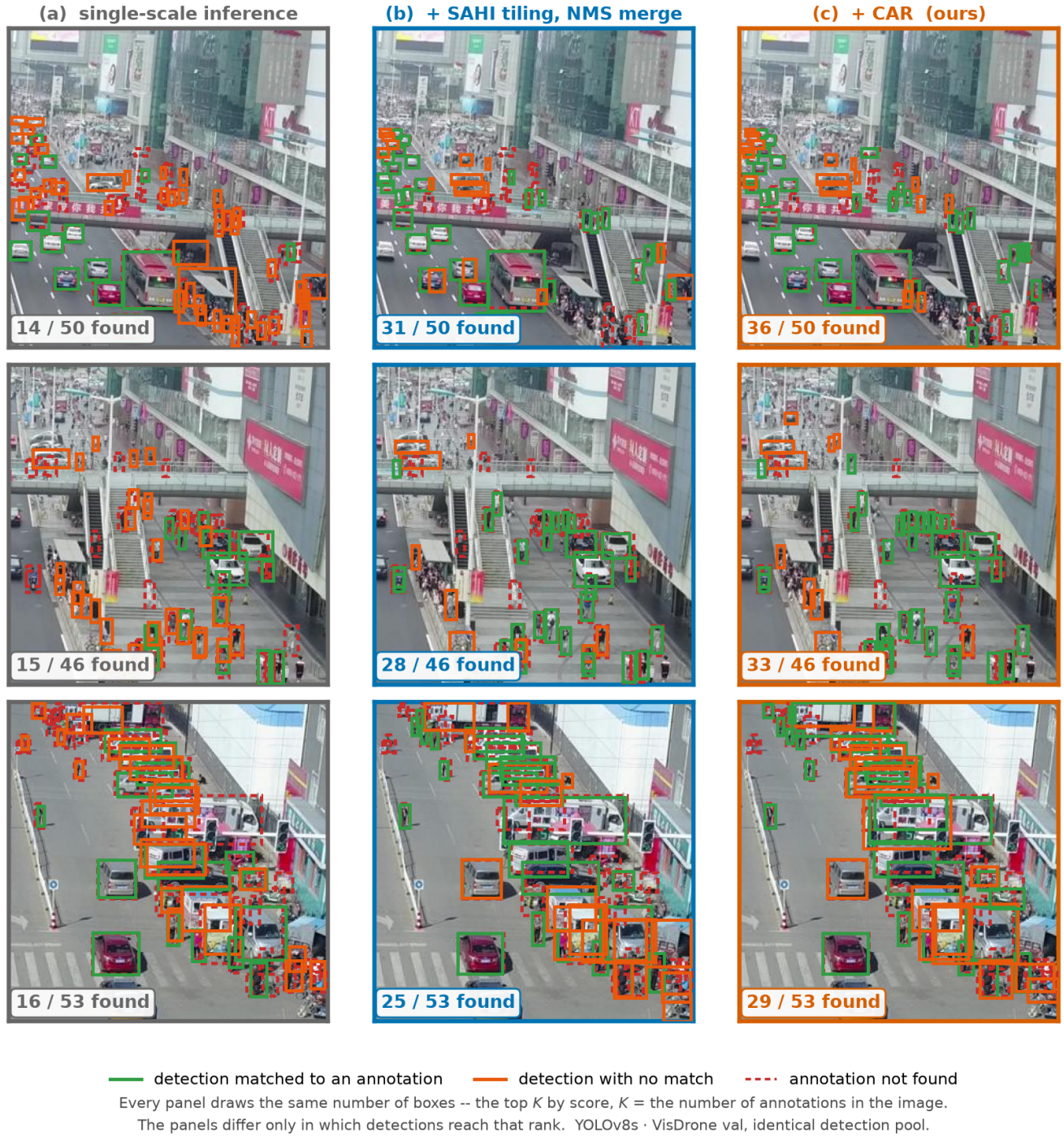
Table 25 crosses the eight merge conditions with twelve architectures under two weight regimes — in-domain and stock COCO — twenty-four configurations in all, so that the two axes examined separately in Table 10 (merge rules, one detector) and Table 12 (detectors, one merge rule) can be read together. The architecture set is the YOLO and RT-DETR ladders of Table 12; Faster R-CNN, the thirteenth detector there, is omitted because no stock-COCO counterpart of it exists under the same class mapping.

## References

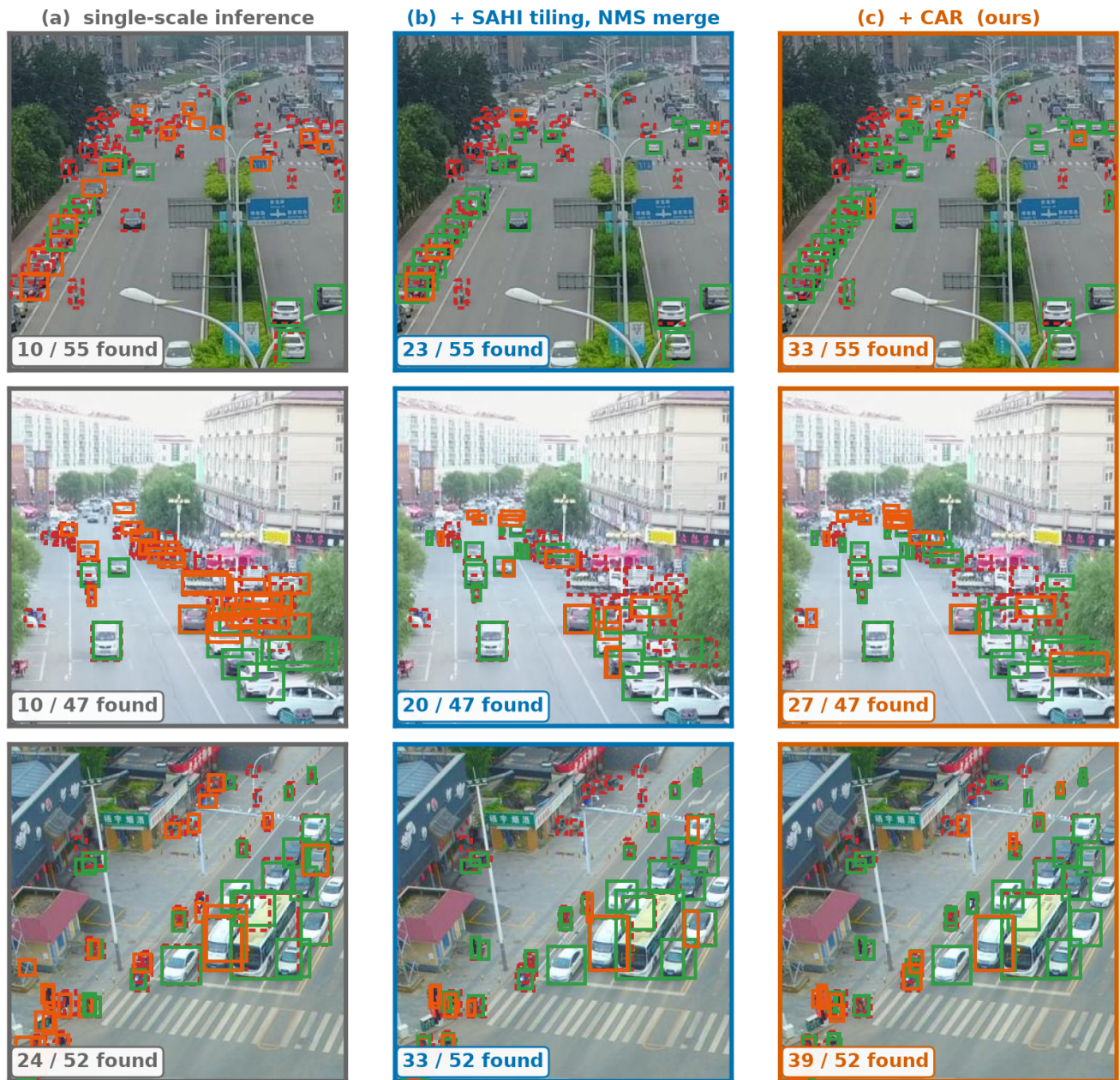
1. Xu, C. *et al.* RFLA: Gaussian receptive field based label assignment for tiny object detection. In *European Conference on Computer Vision (ECCV)*, 526–543 (2022).
2. Huang, Y.-X., Liu, H.-I., Shuai, H.-H. & Cheng, W.-H. DQ-DETR: DETR with dynamic query for tiny object detection. In *European Conference on Computer Vision (ECCV)*, 290–305 (2024).
3. Akyon, F. C., Onur Altinuc, S. & Temizel, A. Slicing aided hyper inference and fine-tuning for small object detection. In *IEEE International Conference on Image Processing (ICIP)*, 966–970 (2022).
4. Zhang, H., Hao, C., Song, W., Jiang, B. & Li, B. Adaptive slicing-aided hyper inference for small object detection in high-resolution remote sensing images. *Remote. Sens.* **15**, 1249 (2023).
5. Wang, H., Li, H., Chen, W. & Zhang, X. ViCrop-Det: Spatial attention entropy guided cropping for training-free small-object detection. *arXiv preprint arXiv:2604.26806* (2026).

6. Bodla, N., Singh, B., Chellappa, R. & Davis, L. S. Soft-NMS – improving object detection with one line of code. In *IEEE International Conference on Computer Vision (ICCV)*, 5561–5569 (2017).
7. Solovyev, R., Wang, W. & Gabruseva, T. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image Vis. Comput.* **107**, 104117 (2021).
8. Shalmani, M. A., Boroujeni, A. R., Amini, A. & Yuan, J. S. From full boards to tiny defects: Scale-aware tile inference with topology-aware merging for high-resolution PCB defect detection. *arXiv preprint arXiv:2605.24726* (2026).
9. Xiao, Y. Group evidence matters: Tiling-based semantic gating for dense object detection. *arXiv preprint arXiv:2509.10779* (2025).
10. Vedapant, N. & Wang, B. CRFace: Confidence ranker for model-agnostic face detection refinement. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1674–1684 (2021).
11. Yoo, S., Kwon, H., Hwang, J.-W. & Lee, K. Automated model evaluation for object detection via prediction consistency and reliability. In *IEEE/CVF International Conference on Computer Vision (ICCV)* (2025).
12. Manoharan, A., Yin, X., Helm, D. & Cheng, C.-H. Cumulative consensus score: Label-free and model-agnostic evaluation of object detectors in deployment. *arXiv preprint arXiv:2509.12871* (2025).
13. Moretti, F., Jin, Y. & Mario, G. Adaptive slicing-assisted hyper inference for enhanced small object detection in high-resolution imagery (2026). ArXiv:2604.19233, [2604.19233](https://arxiv.org/abs/2604.19233).
14. Pato, L. V., Negrinho, R. & Aguiar, P. M. Q. Seeing without looking: Contextual rescoring of object detections for AP maximization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020).
15. Küppers, F., Kronenberger, J., Shantia, A. & Haselhoff, A. Multivariate confidence calibration for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1322–1330 (2020). URL <https://arxiv.org/abs/2004.13546>.
16. Kuzucu, S., Oksuz, K., Sadeghi, J. & Dokania, P. K. On calibration of object detectors: Pitfalls, evaluation and baselines. In *Computer Vision – ECCV 2024*, vol. 15079 of *Lecture Notes in Computer Science*, 185–204 (Springer, 2024). DOI 10.1007/978-3-031-72664-4\_11.
17. Gilg, J., Teepe, T., Herzog, F., Wolters, P. & Rigoll, G. Do we still need non-maximum suppression? accurate confidence estimates and implicit duplication modeling with iou-aware calibration. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 4838–4847 (2024). URL <https://arxiv.org/abs/2309.03110>.
18. Miller, D., Nicholson, L., Dayoub, F. & Sünderhauf, N. Dropout sampling for robust object detection in open-set conditions. In *IEEE International Conference on Robotics and Automation (ICRA)* (2018). URL <https://arxiv.org/abs/1710.06677>.
19. Lakshminarayanan, B., Pritzel, A. & Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30 (NIPS)* (2017). URL <https://arxiv.org/abs/1612.01474>.
20. Cheng, T. *et al.* YOLO-World: Real-time open-vocabulary object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16901–16911 (2024).
21. Liu, S. *et al.* Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision (ECCV)*, 38–55 (2024).
22. Zhu, P. *et al.* Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis Mach. Intell.* **44**, 7380–7399 (2021).
23. Xia, G.-S. *et al.* DOTA: A large-scale dataset for object detection in aerial images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3974–3983 (2018).
24. Lin, T.-Y. *et al.* Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 740–755 (2014).
25. Efron, B. & Tibshirani, R. J. *An Introduction to the Bootstrap* (Chapman & Hall, 1993).
26. Holm, S. A simple sequentially rejective multiple test procedure. *Scand. J. Stat.* **6**, 65–70 (1979).
27. Zheng, Z. *et al.* Distance-IoU loss: Faster and better learning for bounding box regression. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 34, 12993–13000 (2020). DOI 10.1609/aaai.v34i07.6999.
28. Zheng, Z. *et al.* Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybern.* **52**, 8574–8586 (2022). DOI 10.1109/TCYB.2021.3095305.

29. Suárez-Ramírez, J., Santana-Cedr s, D. & Monz n, N. DAHI: a fast and efficient density aided hyper inference technique for large scene object detection. *Pattern Recognit.* **171**, 112228 (2025). DOI 10.1016/j.patcog.2025.112228.
30. Xu, C. *et al.* Detecting tiny objects in aerial images: A normalized wasserstein distance and a new benchmark. *ISPRS J. Photogramm. Remote. Sens.* **190**, 79–93 (2022). URL <https://arxiv.org/abs/2206.13996>.
31. Wang, J., Yang, W., Guo, H., Zhang, R. & Xia, G.-S. Tiny object detection in aerial images. In *25th International Conference on Pattern Recognition (ICPR)*, 3791–3798 (2020). DOI 10.1109/ICPR48806.2021.9413340.
32. Zhao, Y. *et al.* DETRs beat YOLOs on real-time object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16965–16974 (2024).
33. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 91–99 (2015).
34. Jocher, G., Chaurasia, A. & Qiu, J. Ultralytics YOLOv8 and YOLO11. <https://github.com/ultralytics/ultralytics> (2023).

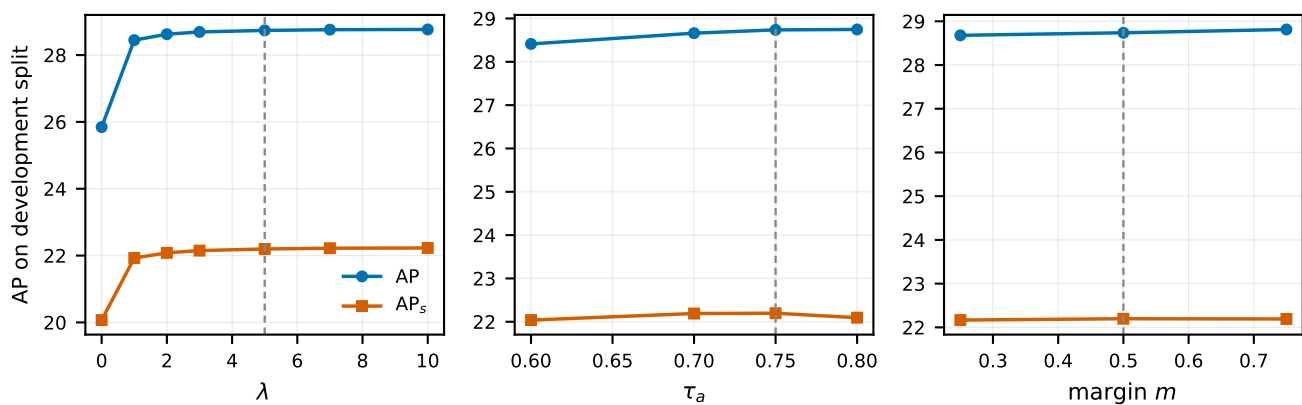


**Figure 3.** What re-ranking looks like. Each row is one VisDrone val scene; the three panels draw *the same number of boxes* — the top  $K$  by score, with  $K$  equal to the number of annotations in the image — so the panels differ only in *which* detections reach that rank, which is precisely the quantity AP measures. (CAR renormalises scores, so a fixed confidence threshold would draw different box counts per condition and conflate the method’s effect with normalisation.) Green: detection matched to an annotation; orange: unmatched detection; dashed red: annotation not found. Counts in each badge are within the displayed crop. The scenes were selected by monotone improvement (a) < (b) < (c) inside the crop; this is not a rare event — 315 of the 548 val images (57%) improve monotonically across the three conditions.

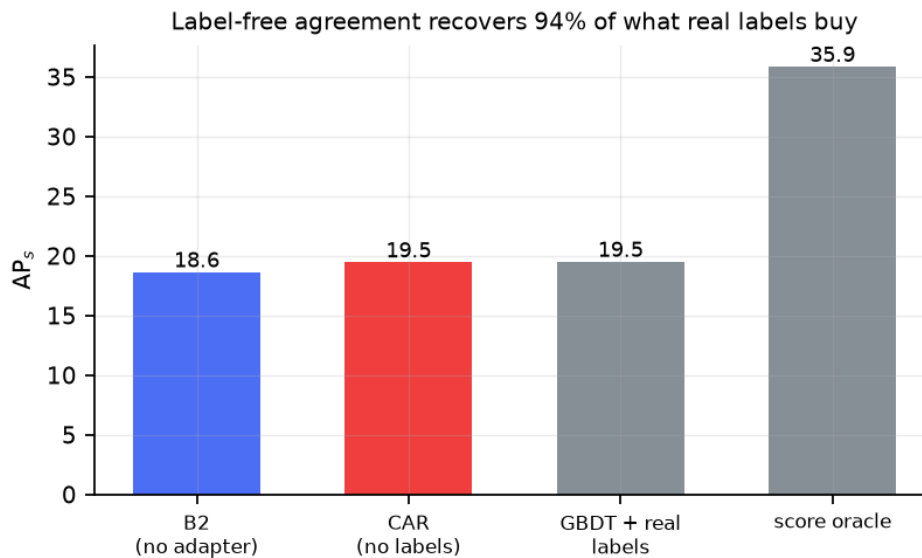


— detection matched to an annotation    — detection with no match    - - - annotation not found  
 Every panel draws the same number of boxes -- the top  $K$  by score,  $K$  = the number of annotations in the image.  
 The panels differ only in which detections reach that rank. yolov8s-visdrone, visdrone-test-dev, identical detection pool.

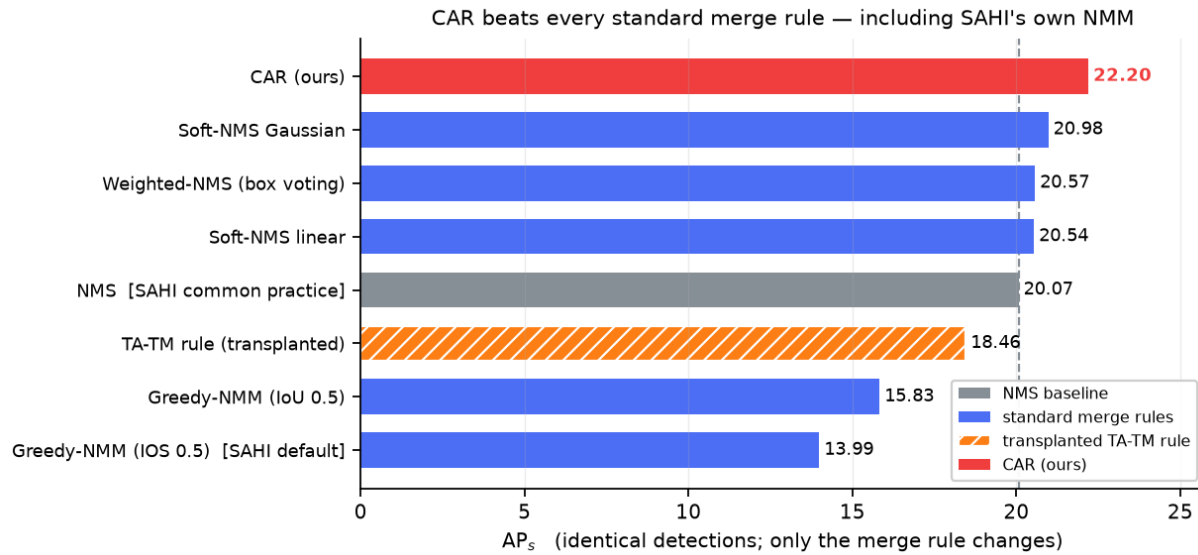
**Figure 4.** The same fixed-budget protocol as Figure 3, on the held-out test split (VisDrone test-dev, YOLOv8s). The progression single-pass → tiling → CAR carries over to scenes that played no role in any development decision.



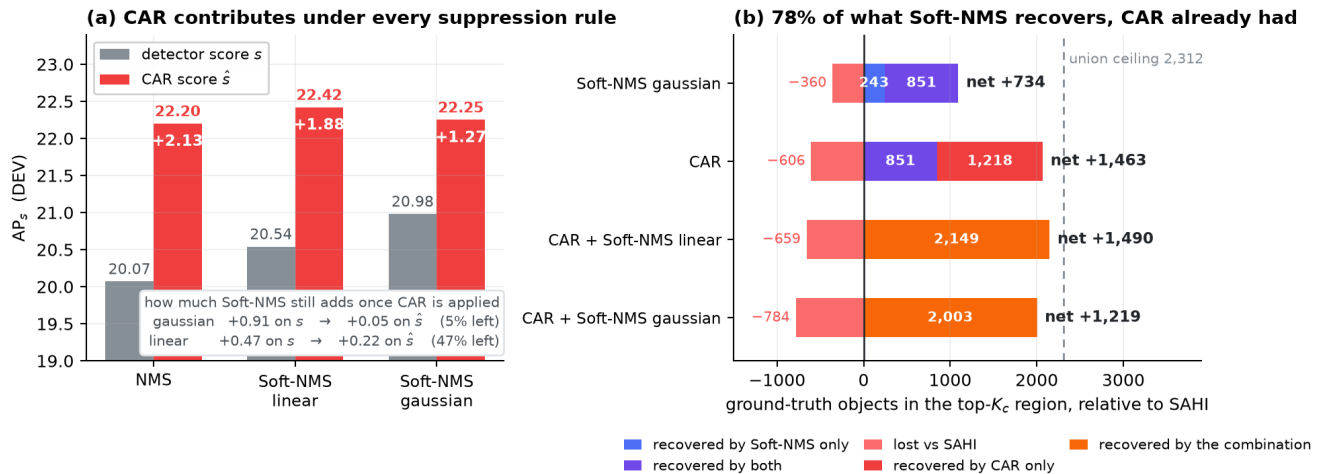
**Figure 5.** Hyper-parameter sensitivity on the development split (VisDrone validation, YOLOv8s), varying one parameter at a time around the frozen setting (dashed vertical lines). Left: the rescoring strength  $\lambda$  saturates at  $\lambda \geq 1$  — the mechanism acts through ranking, so the exact multiplier barely matters. Centre: matching threshold  $\tau_a$ . Right: eligibility margin  $m$ .



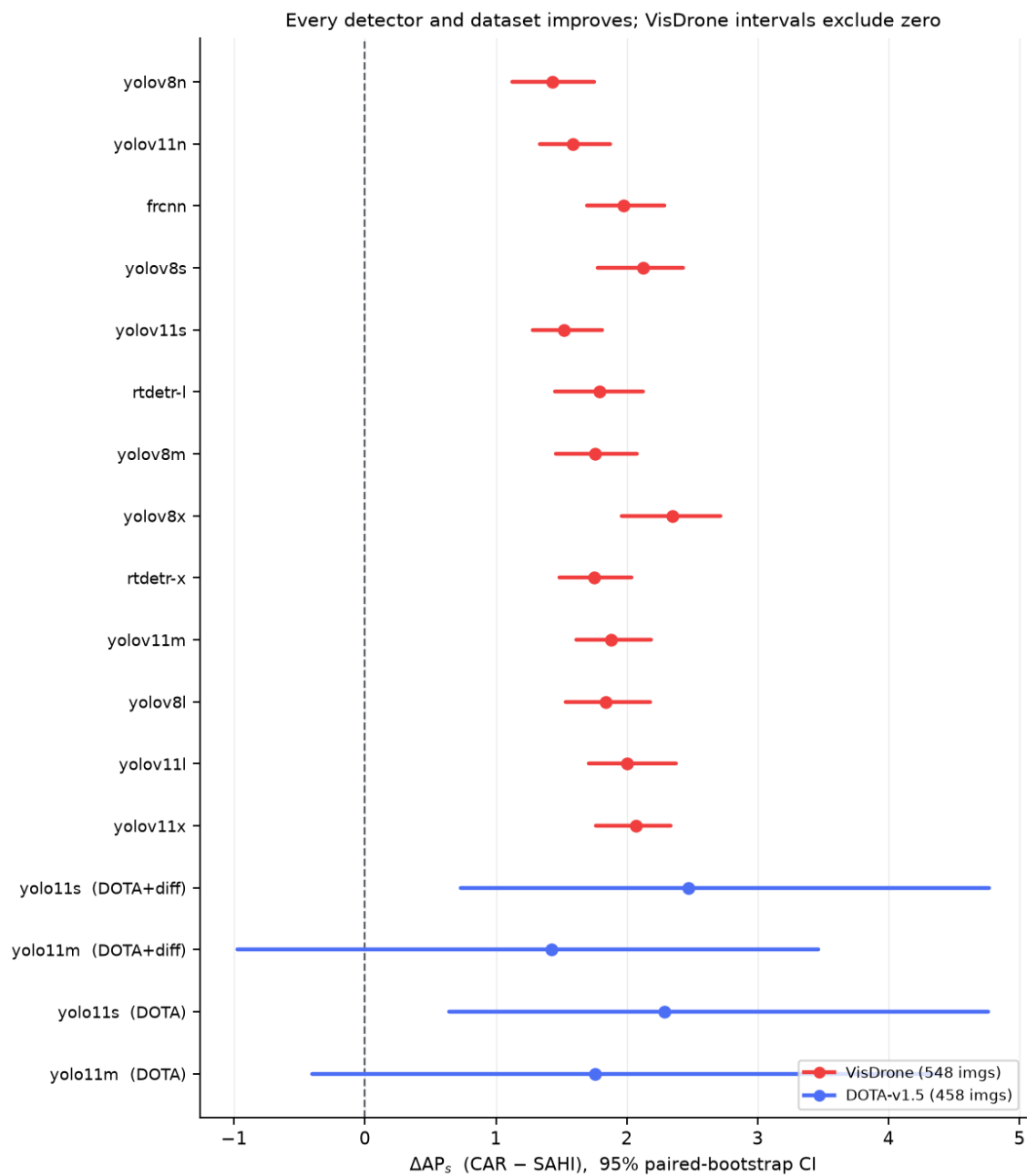
**Figure 6.** Label tiers. Label-free agreement recovers most of what true labels buy, and both remain far from the score oracle.



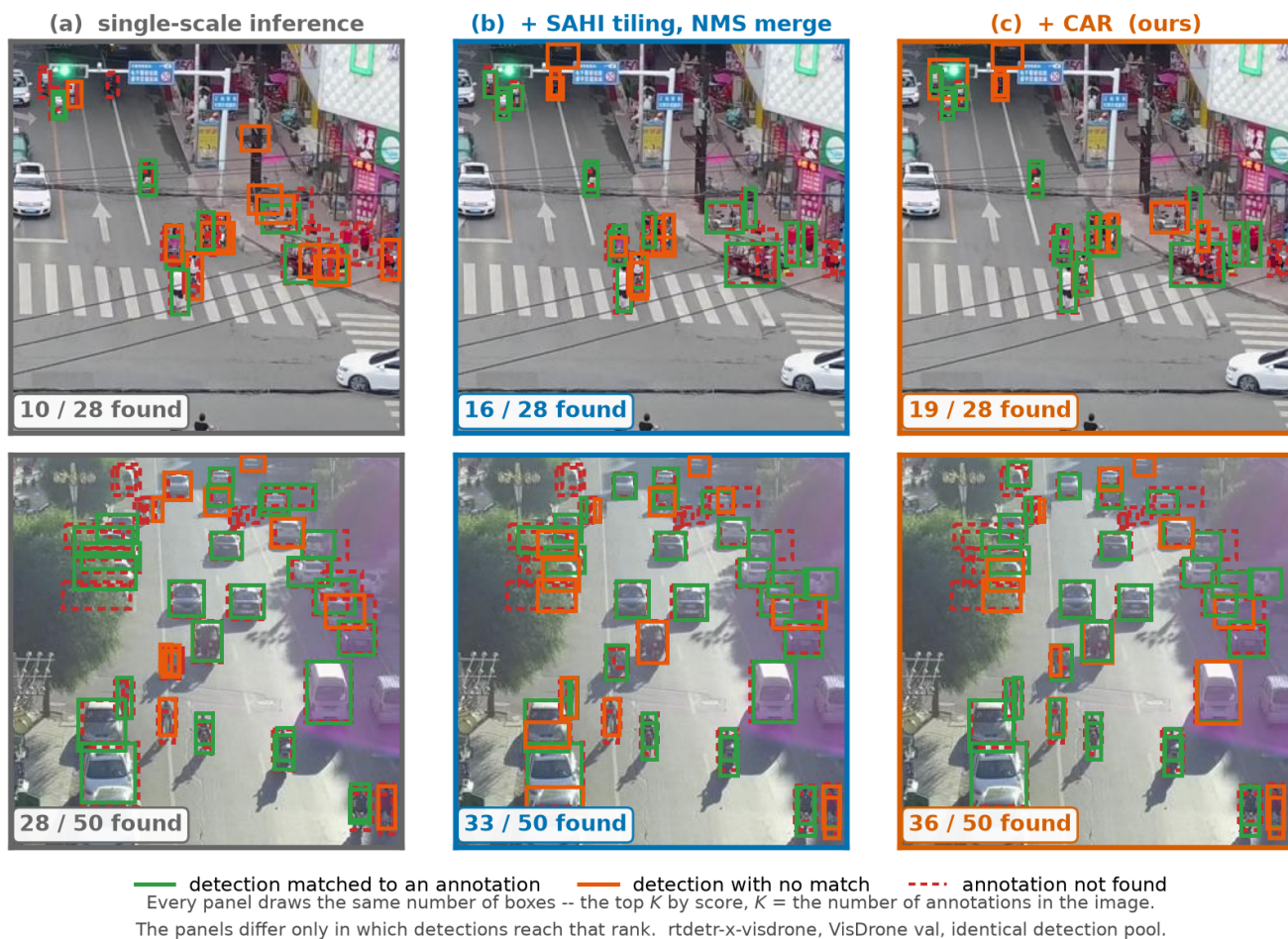
**Figure 7.** Merge-rule battery on identical detections, including the transplanted TA-TM rule (hatched, not a standard merge rule). Rules that preserve the original score ordering cluster tightly around the NMS baseline (dashed); SAHI’s default Greedy-NMM falls far below it.



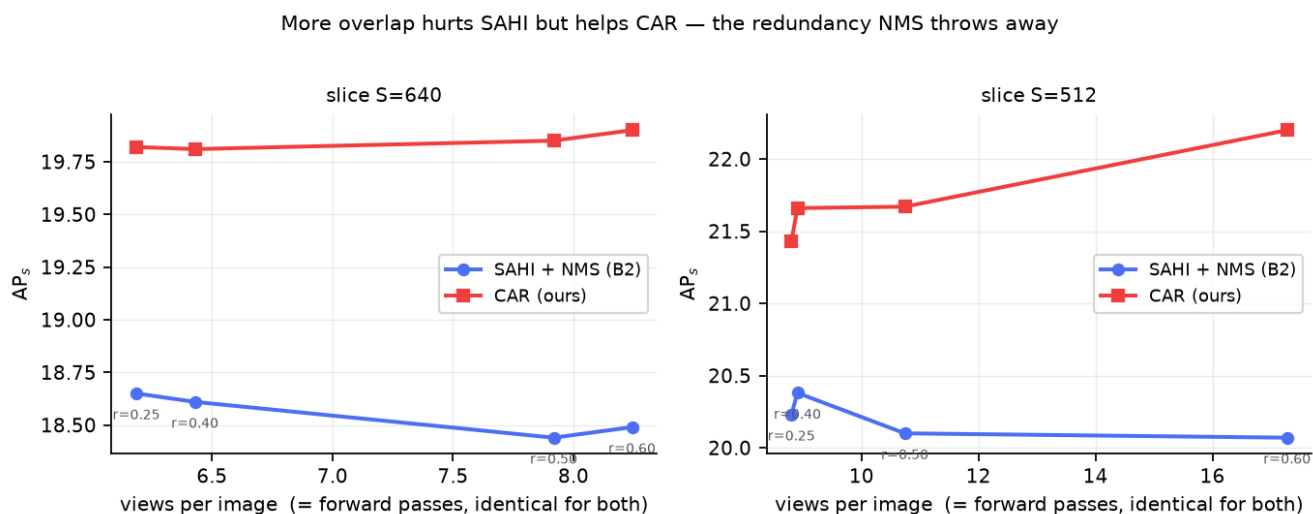
**Figure 8.** CAR and Soft-NMS mostly repair the same failure. (a) The  $2 \times 3$  grid on DEV: raw score  $s$  versus CAR score  $\hat{s}$  under three suppression rules. CAR contributes under every rule; what Soft-NMS adds on  $s$  largely vanishes on  $\hat{s}$  (gaussian keeps 5%, linear 47%). (b) Ground-truth objects newly recovered into the top- $K_c$  region relative to SAHI, with losses drawn to the left of zero. 78% of the gaussian variant’s recoveries are shared with CAR; the combination stays below the union ceiling while losing more.



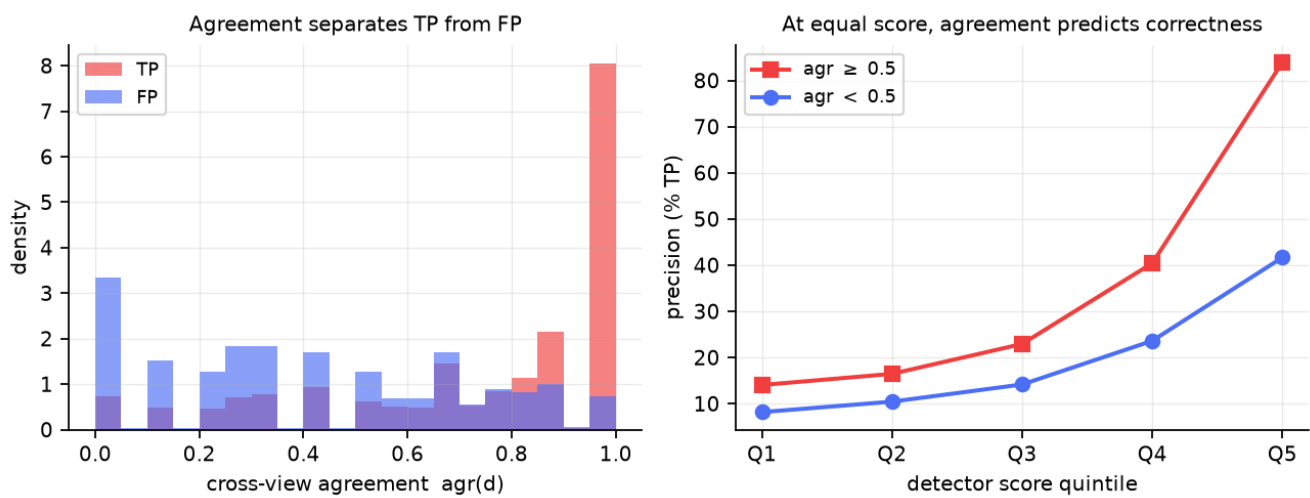
**Figure 9.** Paired-bootstrap 95% confidence intervals for  $\Delta AP_s$  across all detectors and datasets. All VisDrone intervals exclude zero; the wider DOTA intervals reflect its 458-image sample (see Section 3.13).



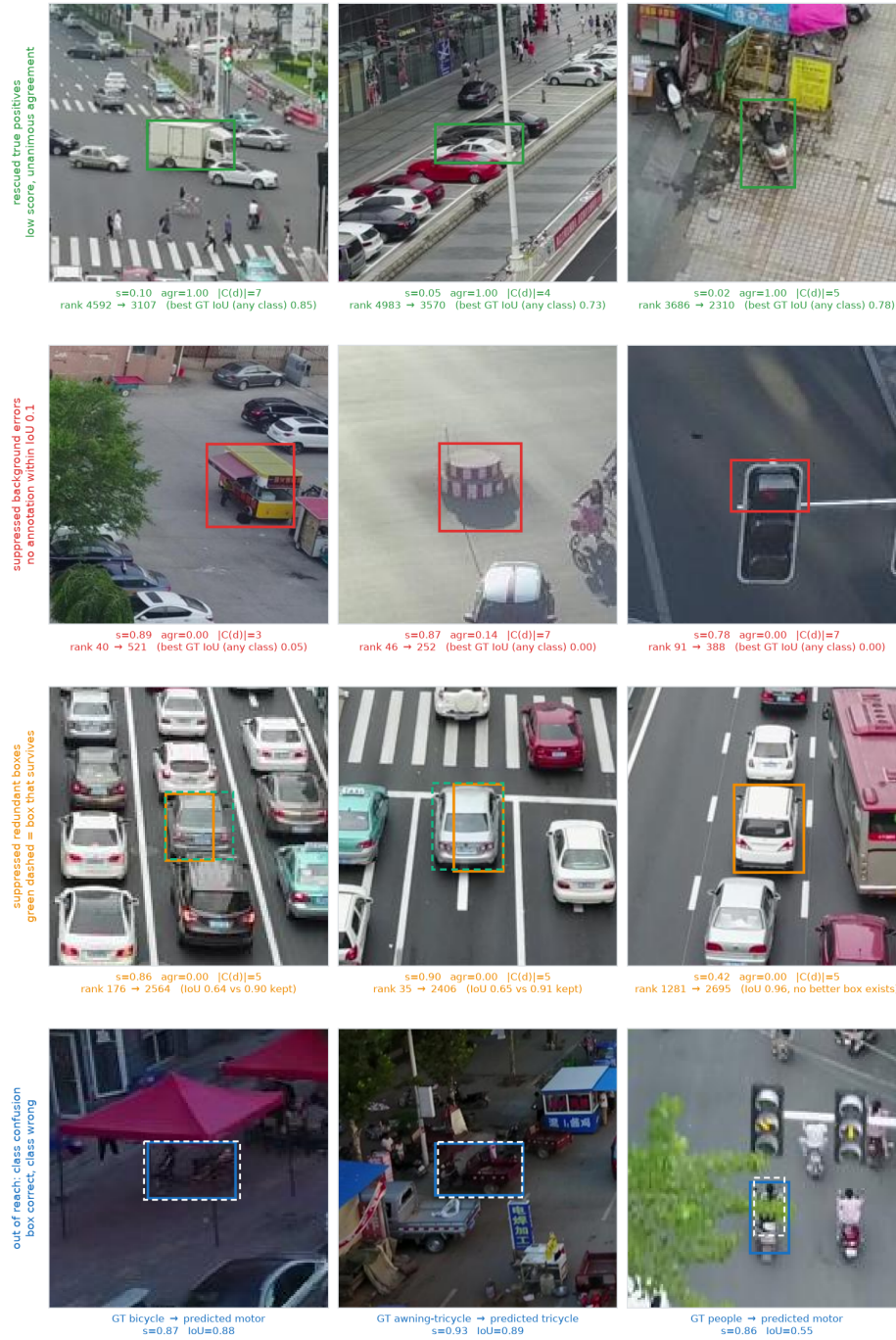
**Figure 10.** The RT-DETR rescue, qualitatively (RT-DETR-x, VisDrone val; same fixed-budget protocol as Figure 3). Tiling (b) recovers small objects that the single pass (a) misses but promotes cross-tile duplicates and boundary fragments into the top ranks; CAR (c) demotes the unreproduced boxes and keeps the recovered objects, ending above the single pass on both overall AP and  $AP_s$ .



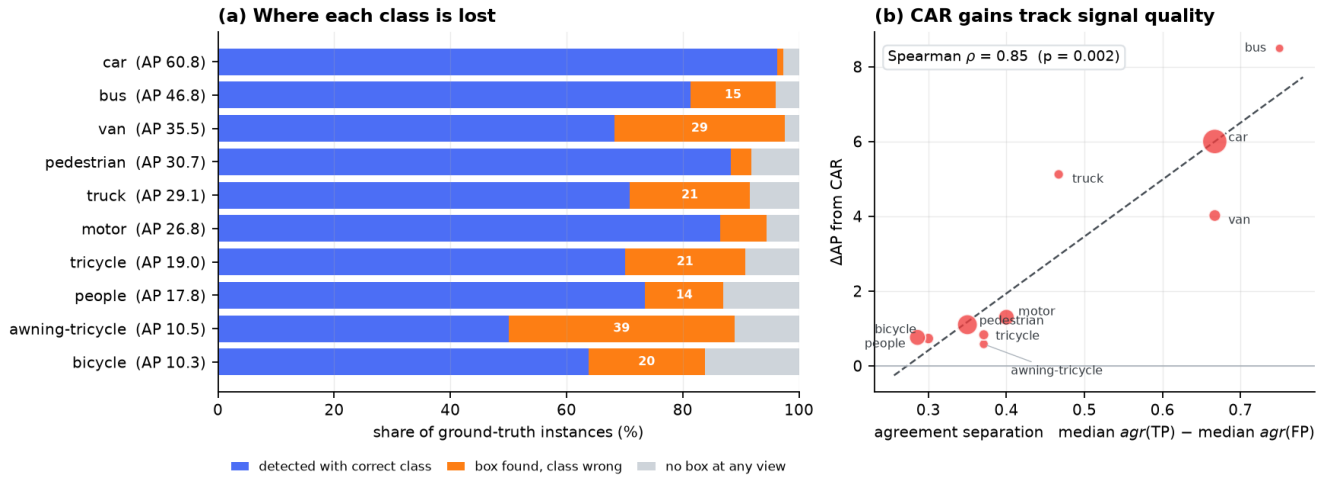
**Figure 11.** Overlap-gain curves at fixed slice size. Additional overlap degrades SAHI while improving CAR; the two methods consume identical compute in every pair.



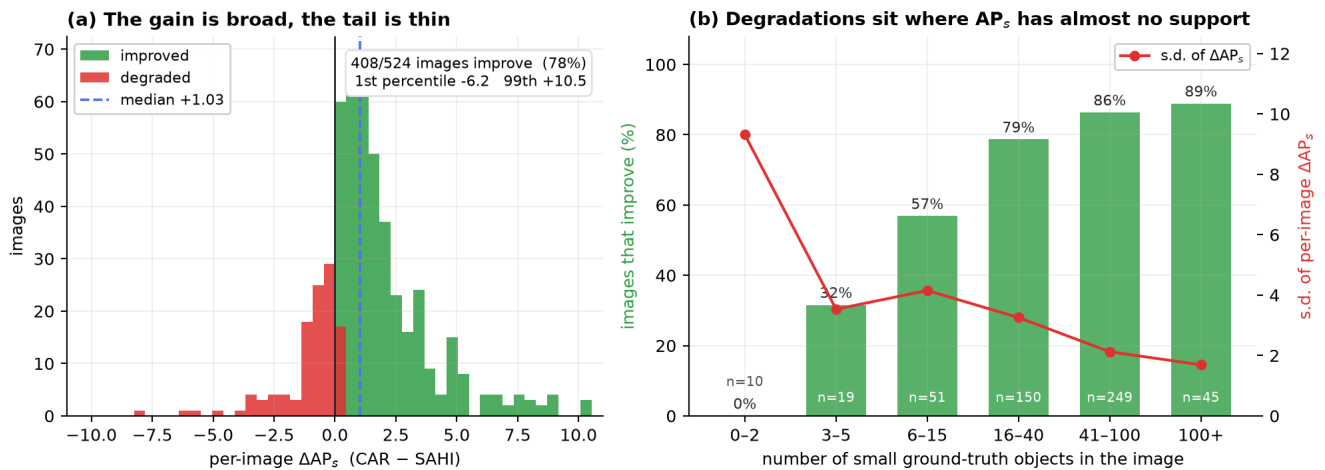
**Figure 12.** Agreement is discriminative at fixed score. Within every score quantile, detections with high agreement are markedly more precise — the signal is orthogonal to the detector score.



**Figure 13.** Twelve qualitative cases, three per row. **Row 1:** correct low-confidence detections rescued by unanimous cross-view agreement. **Row 2:** high-confidence background errors that no view reproduced (a vendor’s cart, a traffic barrel, a traffic-light head). **Row 3:** redundant boxes on real objects — the dashed box is the better-localised detection that survives, so the object is kept; the rightmost panel is the rare case where the demoted box is the only good one. **Row 4:** what rescoring cannot reach — the box is correct but the predicted class is wrong. Annotations:  $s$  detector score,  $agr$  agreement,  $|C(d)|$  number of views that could see the box.



**Figure 14.** (a) Where each class is lost. The three segments are mutually exclusive shares of ground-truth instances: detected with the correct class, boxed but assigned another class, and never boxed at any view. Only the first segment is what rescoring reorders; the other two lie outside its reach. (b) Per-class gain against per-class agreement separation. Marker area is proportional to the number of ground-truth instances.



**Figure 15.** Per-image effect on DEV. (a) Distribution of per-image  $\Delta AP_s$ : 78% of images improve and the negative tail is thin. (b) Share of improving images (bars) and the standard deviation of the per-image change (line), binned by the number of small ground-truth objects. Degradations concentrate where  $AP_s$  has almost no support; where tiled inference is actually needed, the gain is near-uniform.