YOUR "CLEAN SPLIT" MAY NOT BE CLEAN: MEASURING RECORDING-LEVEL LEAKAGE IN VIDEO SEGMENTATION BENCHMARKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video-derived segmentation benchmarks are routinely split at frame or clip level, so near-copies of one recording fall on both sides of the split. That this leaks information is known; we ask whether the *standard way of measuring it* is itself reliable. The conventional comparison subtracts scores from **different test sets**, confounding the memorisation gain with test-set difficulty. We propose two tools. **Protocol-matched evaluation** re-scores both protocols' checkpoints on a *common* test set, separating the test-set swap from the contrast between the two training protocols without retraining: on infrared, visible and a second public benchmark the matched gain is +6.4 to +8.8 IoU, of which the conventional comparison captures 36%, 68% and $-$**17%**. **Near-duplicate auditing** against adjacent-frame and random-pair references tests whether a "clean" split is clean: that benchmark's recording-disjoint split is as duplicated as adjacent frames (72.5% against 70.6%), which is consistent with why the conventional measurement fails there, while the primary benchmark's split passes on both bands (0.5% visible, 6.9% infrared). On the common clean test set the visible-band night score rises from 0.73 to 0.91, and comparisons on the protocols' respective test sets reverse the observed modality ordering. We release the audit and protocol-matched evaluation code with every per-architecture result.

1 INTRODUCTION

A video benchmark gives every recording hundreds of frames a fraction of a second apart, sharing a background, a camera and — in tiny-target tasks — an object at nearly the same pose (Fig. 1a). Splitting at random puts near-copies of test frames into training, where a model scores by matching rather than generalising. Prior work documents this leakage in correlated and hierarchically structured samples (Woo, 2026; Zhang et al., 2026; Glazner et al., 2025), and the recommended remedy is a recording-disjoint split. We test two assumptions underlying that remedy, and both fail in ways that change conclusions.

Assumption 1: the size of the leak can be read off by comparing the two protocols. The standard measurement subtracts a clean-protocol model's test score from a leaky-protocol model's, but the two test sets are not equally hard for the two checkpoints (§4.1), so the gap adds that offset to the memorisation gain. On the splits we evaluate that confound is not second-order: it can cancel the leakage signal entirely.

Assumption 2: a recording-disjoint split is clean. Holding out whole recordings removes within-recording leakage but says nothing about near-duplicate recordings. Whether the result is free of near-copies is asserted, not measured (Woo, 2026).

We replace both assumptions with measurements.

Protocol-matched evaluation (§4) re-evaluates both protocols' saved checkpoints on one common test set, removing the test-set swap (Fig. 1c). Across eight architectures on infrared, visible and the public CST benchmark, this fixed-test protocol contrast is +6.4, +8.5 and +8.8 IoU, of which the conventional comparison captures 36%, 68% and $-$17%: it under-reports leakage by 1.5–3$\times$ on two benchmarks and reverses its sign on the third. The gain is architecture-agnostic and largest

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

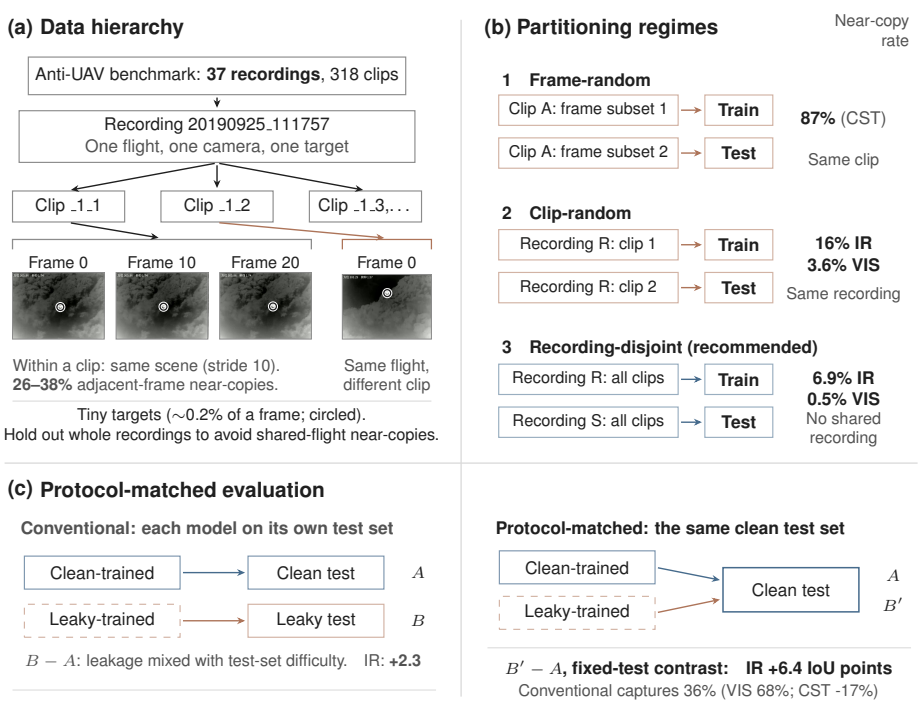


Figure 1: *From recording-level dependence to protocol-matched evaluation.* **(a)** Recordings are cut into clips, clips into frames (real infrared; circles mark the target); frames one training stride apart are near-copies, while another clip of the same flight is minutes apart. **(b)** Three partitioning regimes and their near-copy rates. **(c)** The conventional measurement subtracts scores from *different* test sets; protocol-matched evaluation re-scores both checkpoints on one clean test set.

where models are weakest, raising visible-band night-time IoU from 0.733 to 0.913 on the same clean frames.

Near-duplicate auditing (§5) measures, for every test frame, the nearest-neighbour similarity to the training set, referenced against adjacent frames of a clip and against random pairs. It is CPU-only. Four of five audited splits are healthy, with cross-split near-copy rates of 0.5–6.9% against adjacent-frame rates of 26–71%; CST's recording-disjoint split instead scores **72.5%**, indistinguishable from adjacent frames (70.6%), evidently because near-identical content sits under different sequence identifiers. A contaminated clean split inflates the clean baseline too, so across our three settings the more contaminated that split, the less leakage the conventional comparison recovers — a reading, not a fitted law (§5.4).

What leakage does to conclusions (§6). The observed modality ordering, and the observed ordering of an architecture pair, both change with the protocol.

Scope. Both video benchmarks are public and their SAM 2 masks come from concurrent work (§3.1), so the splits and every number rebuild from public frames with released code — the matched protocol on any benchmark, the pixel screen only on static-camera, tiny-target footage (§5.3(iii)).

2 Related Work

Leakage as a measurement problem. Data leakage — test-distribution information reaching the model in training — is long known (Kaufman et al., 2012) and a leading cause of scientific irreproducibility (Kapoor & Narayanan, 2023). That literature mostly treats leakage as a bug to remove; we treat it as a quantity to *measure*. Near-duplicate images inflate reported accuracy on CIFAR (Barz & Denzler, 2020), and a freshly collected test set of the same nominal distribution moves ImageNet accuracy by a large margin (Recht et al., 2019) — the difficulty confound we isolate here. The established remedy is splitting by recording, scene or subject (Yagis et al., 2021), and a

reassessment of a published gesture-recognition pipeline attributes its accuracy to a frame-level split (Varga, 2026).

The closest works. Woo (2026) advocates scene-level splits, reporting that on MPI Sintel a frame-level split inflates intrinsic-decomposition reflectance PSNR by 1.6–2.0 dB. Zhang et al. (2026) audit case-level overlap in whole-slide-image benchmarks, tracing identifiers to 92–100% train–test overlap and partitioning the test set into leaked and clean subsets under a fixed model. Glazner et al. (2025) and Figueiredo & Mendes (2024) cluster visually similar frames into better splits for video-derived data, the latter measuring train–test similarity. We differ in two ways. First, none re-scores both protocols' checkpoints on a *common* evaluation set; a per-protocol test partition confounds the protocol contrast with test-set difficulty, and can invert its sign (§4). Second, such similarity is measured to *build* a split, not to audit a published one; we audit against adjacent-frame and random-pair references and find a widely used public benchmark highly duplicated (§5). Our three regimes span the random / temporal / scene gradient across datasets rather than within one; a within-dataset sweep is left to future work (§8).

Benchmarks and reporting. Anti-UAV benchmarks are videos (Jiang et al., 2023; Xie et al., 2025), so segmentation inherits the recording structure. Concurrent work (Anonymous, 2026) builds the paired infrared–visible segmentation benchmark used here, re-splits it by recording, measures its leak with the conventional comparison that §4 examines, and reports its size-stratified leaderboard; we take that benchmark as given and ask how its leakage should be measured. Frames within a recording are highly dependent, making the recording the effective sampling unit (Cameron & Miller, 2015; Miller, 2024; Reinke et al., 2024).

3 SETUP

3.1 DATASETS

Anti-UAV (primary; infrared + visible). 37 recordings of a single UAV from the public Anti-UAV tracking benchmark (Jiang et al., 2023), each a flight cut into consecutive clips, from a paired thermal-infrared / visible camera, 640×512 and 1920×1080 native and both resized to the 512×640 network input, with a day/night label. Targets are tiny: the median occupies 0.24% of the frame and 99% under 1%. It ships boxes and no masks; the masks and the recording-disjoint partition used here are those of concurrent work (Anonymous, 2026), made by prompting SAM 2 (Ravi et al., 2024) with the released boxes, so every IoU here is agreement with pseudo-masks, not with human labels (§8).

CST Anti-UAV (public; infrared). 220 thermal sequences, each a single continuous recording (Xie et al., 2025). It ships no segmentation split either; its masks and its 70/15/15 sequence partition come from the same concurrent pipeline.

IRSTD-1k (Zhang et al., 2022) and NUDT-SIRST (Li et al., 2023) (public; still images). 1,001 real and 1,327 synthetic infrared frames with masks, under the authors' train/test lists. Having no temporal structure, they are the audit's controls.

3.2 PARTITIONING PROTOCOLS

Frames are sampled at stride 10 for training and evaluation alike. Under the **recording-disjoint ("clean")** protocol whole recordings go to train, validation or test, day/night-stratified on Anti-UAV: 26 / 5 / 6 recordings and 225 / 43 / 50 clips, and a seeded 154 / 33 / 33 sequence split on CST. Under the **leaky** protocol — the common practice — Anti-UAV clips are shuffled regardless of recording (223 / 48 / 47 clips drawn from 37 / 26 / 25 recordings), so clips of one flight fall on both sides: the *clip-random* regime. On CST the sequence is the recording, so the leaky protocol is *frame-random*, with all 220 sequences on both sides. Both protocols share stride, resolution, loss, optimiser and epoch budget, and stride-10 training-set sizes are within 1.5% in every setting; only the assignment of frames differs.

The two protocols therefore differ in more than exposure to the test recordings, and §4 measures their contrast rather than exposure alone. Of the 50 clips of the common clean test set, 35 appear in leaky training and 6 in leaky validation; the leaky arm also sees more distinct recordings at the same clip

budget (37 against 26), and each checkpoint is selected on its own protocol's validation set. A.9 gives the full composition.

3.3 Models and training

Five classic encoder–decoders with an ImageNet-pretrained ResNet-34 encoder — U-Net, PSP-Net, PAN (the pyramid attention network of `segmentation_models_pytorch`), FPN and DeepLabV3+ — plus three modern configurations on CST: FPN and UPerNet with a ConvNeXt-T backbone, and SegFormer-MiT-B2 (He et al., 2016; Ronneberger et al., 2015; Zhao et al., 2017; Li et al., 2018; Lin et al., 2017; Chen et al., 2018; Liu et al., 2022; Xiao et al., 2018; Xie et al., 2021; Iakubovskii, 2019). *Ours*, an earlier ResNet-34 FCN, is a sixth classic architecture in the cross-validated leaderboard. All train for 30 epochs under one AdamW recipe with the checkpoint selected on validation UAV-IoU (A.4). The recipe is tuned for the ResNet family, so the modern architectures, which use it unchanged, are reported for the *leakage delta* against their own clean run, not for absolute ranking.

3.4 Evaluation

Metric. UAV-IoU: foreground intersection-over-union pooled over the evaluated frames, so that false positives on frames without a target enter the union; reported overall and by day/night and target-size tercile, the terciles taken from the held-out frames. The cross-validated leaderboard instead averages per-frame IoU within each recording over frames containing a target, so its levels are not comparable to the pooled numbers.

Four evaluation cells (§4). Each architecture yields two checkpoints, one per protocol, and each is scored on both test sets: A (clean-trained, clean test), B (leaky-trained, leaky test), B' (leaky-trained, clean test) and A'' (clean-trained, leaky test). $B - A$ is the conventional comparison and $B' - A$ the contrast on a fixed evaluation set. Nothing is retrained.

Near-duplicate audit (§5). For each test frame we take the maximum Pearson similarity to any training frame in an 80×64 grayscale, mean-centred, L2-normalised representation, and call the fraction above 0.99 the near-copy rate. §5.1 gives the references that contextualise it; the target-layout variant is in A.2.

Cross-validated leaderboard (§6.3). The concurrent benchmark's five-fold recording-level cross-validation over the 37 recordings — per-recording means, 95% t-intervals and recording-paired differences — which §6.3 cites and A.1 reproduces (A.4).

Statistical reporting. Leakage deltas are seed-42 unless stated (A.7, A.11). The single split holds out six recordings, so we claim no absolute levels from it and report only paired differences on identical frames (§4) and orderings that §6.3 retests under cross-validation.

4 Protocol-matched evaluation removes the test-set swap

Table 1's $B' - A$ column scores both protocols' checkpoints on one common clean test set: the leaky-trained ones win by +6.4 UAV-IoU points (§3.4; "points" hereafter) on Anti-UAV infrared, +8.5 on visible and +8.8 on CST, against +2.3, +5.7 and -1.5 conventionally (five-architecture means; Fig. A2).

4.1 The conventional measurement compares different test sets

$B - A$ scores each protocol's model on its own test set, and the two differ: on Anti-UAV, six held-out recordings against 47 clips drawn from 25 of the 37 (A.9). Exactly, $B - A = (B' - A) + (B - B')$ — the matched gain plus a *swap term* of -4.1 (IR), -2.7 (VIS) and -10.3 (CST). The leaky checkpoint scores *higher* on the clean test set than on its own; the clean-trained checkpoints move the other way on Anti-UAV and downwards on CST (A.7). The offset is a property of the test-set–checkpoint pair, not a fixed difficulty difference, so no fixed correction to $B - A$ recovers the gain; part of the Anti-UAV shift is contamination, the clean-trained model having trained on 27 of the 47 leaky-test clips.

4.2 PROTOCOL-MATCHED EVALUATION

We hold the test set fixed (Fig. 1c): each architecture's two checkpoints are scored on both test sets, giving §3.4's four cells with nothing retrained and budgets matched (§3.2). The *matched* gain is $B' - A$, the contrast between the two training protocols on a fixed evaluation set; the *captured fraction* is $(B - A)/(B' - A)$ on per-dataset means. The contrast includes direct training exposure, recording coverage and validation-based checkpoint selection, and does not isolate their contributions. The clean test set is the right common set because it alone consists entirely of recordings the clean-trained checkpoint has never seen, whereas the leaky test set draws only 9 of its 47 clips from such recordings (A.7).

4.3 RESULTS

The conventional number recovers only part of the gain, or reverses it (Table 1, Fig. A2): 36% of the matched gain on infrared, 68% on visible, -17% on CST. On the common *leaky* test set the two checkpoints nearly converge instead ($B - A''$ is -1.4, +0.1 and -0.1) — a control, both having seen its recordings.

The direction does not depend on the architecture. All 15 architecture–dataset entries (eight distinct architectures; §3.3) have a positive matched gain, from +4.9 (PSPNet, IR) to +15.6 (FPN, CST), all single-seed and with the modern CST entries carrying §3.3's reporting limit (Table A6). The conventional column is not consistent: on CST it is negative for four of five architectures, positive only for ResNet-34 FPN at a single seed, and the inversion survives that entry's exclusion (A.7).

CST is the decisive case. Its leaky protocol is frame-random rather than clip-random (§3.2), so the stronger leak plausibly raises B, *against* the inversion (A.6). Its clean split holds near-copies of 72.5% of its test frames (Table 2), which §5.4 offers as a reading of the captured fraction, not a demonstrated cause.

4.4 WHERE LEAKAGE HELPS MOST

Splitting the visible-band matched gain by day and night on the same clean frames (Fig. A3, appendix), the gain is +4.1 points by day (0.885 $\rightarrow$ 0.925) and +18.0 at night (0.733 $\rightarrow$ 0.913). On the two held-out night recordings the clean protocol leaves a 15.2-point night deficit against day — the visible-band night weakness concurrent work reports (Anonymous, 2026) — on all five architectures at seed 42 and the two with seeds 43–44; leaky training cuts that deficit to 1.2 points on identical frames. Whether it holds on other night recordings is untested on this split (§6.1). Leakage does not add a constant: the gain is largest where the clean score is lowest — visible night — and within infrared the weaker clean cell is day rather than night (A.7).

4.5 WHAT THE CONTRAST CONTAINS

The matched gain is not pure memorisation: at the same clip budget the leaky-trained checkpoint has also seen more distinct recordings and selected its epoch on a contaminated validation set, so +6.4 to +8.8 is their sum, and the test set is six recordings. Table 1 is seed 42, but the four configurations with a matched pair of checkpoints at further seeds give ten runs whose gains are all positive, from +5.6 to +9.3, with infrared and visible captured fractions not overlapping (32–42% against 62–83%; A.11). §8 states these limits and A.7 the detail. What this section establishes is that the conventional comparison mismeasures leakage in a direction that depends on the clean split — which §5 makes measurable.

5 AUDITING A "CLEAN" SPLIT FOR NEAR-DUPLICATES

§4 treated the recording-disjoint split as the uncontaminated reference. This section audits that assumption (Assumption 2 of §1). The sequence-disjoint CST split that the recording-level remedy prescribes scores as highly duplicated — a split drawn by the concurrent pipeline, the benchmark shipping none (§3.1–3.2).

Table 1: **Protocol-matched evaluation removes the test-set swap.** A/B: clean-/leaky-trained checkpoint on its own test set, whose difference is the conventional comparison. B': the leaky-trained checkpoint re-scored on the *clean* test set, so $B' - A$ is the fixed-test protocol contrast; A'' is the mirror construction on the leaky test set. Means over the architectures of each dataset, UAV-IoU, seed 42. *Captured* $= (B - A)/(B' - A)$. Per-architecture entries in Table A6.

		Own test sets		Common clean test		Common leaky
Dataset	n_{arch}	A	$B - A$	$B' - A$	captured	$B - A''$
Anti-UAV IR	5	0.859	+2.3	**+6.4**	36%	-1.4
Anti-UAV VIS	5	0.837	+5.7	**+8.5**	68%	+0.1
CST (public)	5	0.669	-1.5	**+8.8**	-17%	-0.1

5.1 THE AUDIT

For every test frame we take the maximum similarity to any training frame in the representation of §3.4, by full enumeration, and use *near-copy* operationally for a cross-split similarity above 0.99. At 80×64 the background is expected to dominate a target under 1% of the frame, so a near-copy is a frame with the same background and camera view.

An absolute correlation is uninterpretable on its own: unrelated frames of our datasets correlate at medians spanning nearly eight-fold. Two references in the same representation fix the scale — the random-pair null, 0.0% on all five datasets, and the within-clip adjacent-frame rate, which is what a recording-disjoint split is meant to remove. Both are single-pair statistics while the cross-split score is a maximum over the training set, so they contextualise it rather than calibrating a false-positive rate or certifying a split free of leakage (A.2). We report the ratio adj/clean and, because its terms are not like-for-like, also clean/leaky.

5.2 RESULTS

Adjacent frames are near-copies at 25.9% (VIS), 37.9% (IR) and 70.6% (CST), and 0.0% on the still sets. Across the recording-disjoint splits the rates are 0.5% (VIS), 6.9% (IR), 2.5% (IRSTD-1k) and 4.8% (NUDT-SIRST) — and **72.5%** on CST (Table 2, Fig. 2a). The ratio adj/clean is about 56× (VIS), 5.5× (IR) and 0.97× (CST); on the clean/leaky statistic — two rates, each on its own test set — CST is at 83% against 43% (IR) and 13% (VIS). Rates are frame-level but recording-clustered, so the comparisons rest on order-of-magnitude differences rather than nominal precision. The video rows are one partition that screens clean on both of its bands — so the 6.9%/0.5% gap is modality rather than split — and one benchmark that does not (A.2).

5.3 THREE OBSERVATIONS

(i) The audit returns nothing where it must. On the still sets, which have no temporal structure, adjacent images score 0.0% while the same rows return above-null cross-split matches, so a zero is not insensitivity.

(ii) The CST result is specific. Its sequence-disjoint split scores indistinguishably from adjacent frames within a sequence (72.5% against 70.6%): holding out whole sequences removed nothing this audit can see, which implies near-identical content — at minimum the same site and camera view — under different sequence identifiers. We have not traced the dataset's provenance, inspected the matched pairs, or audited a second draw, so this is what the audit assigns to the dataset rather than a verified duplicate count. It does not contradict Table 1's +8.8: changing the evaluation set moves both A and B', so the observed gap does not determine how the checkpoints would compare on a genuinely disjoint test set. The CST verdict survives a 0.95 threshold while the healthy ones do not (A.2).

(iii) Video-derived datasets of this kind are the risk class. With adjacent rates of 26–71%, a frame-random split puts near-copies of a comparable fraction of test frames into training (86.9% on CST), a clip-random split far fewer (3.6% VIS, 16.0% IR), and a recording-disjoint split fewer still — if the recordings are distinct. The still benchmarks have no temporal leakage to prevent; their

Table 2: **Near-duplicate audit of the recording-disjoint (*clean*) split.** Near-copy rate: fraction of test frames whose nearest training frame has Pearson similarity > 0.99 (full enumeration). *Adjacent*: consecutive stride-10 frames within a clip (a single-neighbour reference, not a ceiling). *Random*: unrelated pairs (null; median shown, rate is 0% everywhere). A scene-similarity screen, not a duplicate detector: a split screens clean when its rate sits far below the adjacent reference; on CST it does not.

				Random	Adjacent		Clean split		
Dataset	Type	n_{test}	n_{train}	median	median	>0.99 (%)	median	>0.99 (%)	adj/clean
Anti-UAV VIS	video	4375	20147	0.24	0.958	25.9	0.939	**0.5**	56.3×
Anti-UAV IR	video	4734	20646	0.36	0.981	37.9	0.953	**6.9**	5.5×
CST (public)	video	323	1540	0.05	0.997	70.6	0.995	**72.5**	1.0×
IRSTD-1k	still	201	800	0.34	0.341	0.0	0.877	**2.5**	–
NUDT-SIRST	still	664	663	0.05	0.101	0.0	0.764	**4.8**	–

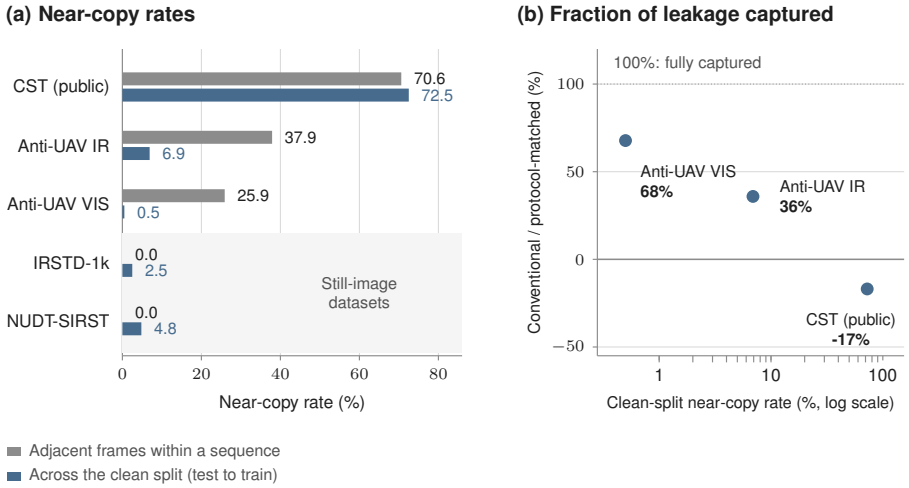


Figure 2: *Near-copy contamination and hidden leakage.* **(a)** Near-copy rate (§3.4) for adjacent frames within a sequence (grey) and across the recording-disjoint split (blue); the random-pair null is 0% everywhere, and the shaded rows are still-image datasets with no adjacency reference. **(b)** The conventional gain as a percentage of the gain on the common clean test set, against the clean-split near-copy rate. Three observations; no trend is fitted.

2.5–4.8% above the null we read as incidental duplicates (Barz & Denzler, 2020), and there the check degenerates to clean-versus-null. We have not audited moving-camera or large-object video.

5.4 Relating the audit to the measurement failure

Fig. 2b plots the clean-split near-copy rate against the fraction of the fixed-test contrast that the conventional comparison captures. The contrast varies little across the three settings and shows no monotone relation to the rate, whereas the captured fraction is 68% (VIS) and 36% (IR) on the partition that screens clean, and -17% on the one that does not, where §4.1's swap term outgrows it. That CST's clean test set is not a held-out sample offers a reading of why it is easier for a model that has seen every sequence — a reading, not a demonstrated cause: three settings, two of them the two bands of one partition, cannot separate contamination from other sources of test-set difficulty. What they establish is §7's rule: the conventional estimate fails whenever the two test sets differ in difficulty for the leaky checkpoint, which a fixed evaluation set removes.

6 What leakage does to conclusions

The leak does not lift every stratum alike: it helps most where the clean model is weakest (§4.4) and varies by architecture and seed. We use the primary benchmark only; CST has one modality and single-seed runs on a ResNet-tuned recipe (§3.3), so we draw no ranking conclusions from it.

6.1 MODALITY INVERSION

Under the leaky protocol the five ResNet-34 decoders (seed 42) all score higher on visible than on infrared (means 0.895 against 0.881); under the clean protocol they rank the other way (0.859 against 0.837; Table 1). Our own model reproduces it on every matched seed (A.7). These are conventional comparisons on different test sets (§4), so the inversion is what a reader inherits, not a property of the modalities.

The day/night decomposition locates the cause: under the clean protocol only the visible band has a night-time deficit on the held-out recordings, 0.733 at night against 0.885 by day. Six recordings cannot settle whether that is a property of the modality (§8). On the common clean frames the two bands' leaky-trained checkpoints score almost alike (0.9225 against 0.9220), because leaky training lifts the visible night score by +18.0 against +4.1 by day (A.7).

6.2 PROTOCOL-SENSITIVE ORDERINGS

The observed ordering of an architecture pair also changes with the protocol. On visible with seeds 42–44, the leaky protocol places U-Net above PAN in all three seeds (means 0.905 against 0.899) with non-overlapping ranges; under the clean protocol the margins are +0.1, -0.1 and +0.7, the ranges overlap, and none exceeds PAN's own seed spread of 0.7. Neither comparison is significant, and non-significance is not evidence of equality; the narrower claim is that from the same models, seeds and budget a reader reporting seed means and win counts would state U-Net $>$ PAN from the leaky split and could not from the clean one — one architecture pair on one modality (A.7).

6.3 WHERE ARCHITECTURES ACTUALLY DIFFER

Under a single aggregate UAV-IoU the five architectures span 1.1 points on the clean infrared split. Concurrent work (Anonymous, 2026) shows with recording-level cross-validation that on this benchmark they separate only on the smallest-target tercile, by decoder family; that leaderboard is its result, reproduced in Table A2, A.1 and Table A7 for reference only. What it adds here is a second failure alongside leakage: on the same folds an unstratified recording-level aggregate spans 2.2 points over six models with every 95% interval overlapping (A.7), so a reader who reports one number misses the only stratum where architectures differ.

6.4 TAKEAWAY

The two failures are distinct. The ordering a reader takes from a leaky split need not survive a clean one (§6.1–§6.2), and a clean split read as one number still hides the stratum where architectures differ (§6.3). Neither fix suffices alone: stratifying a leaky split does not remove the leak, and splitting cleanly does not stratify, so a trustworthy comparison on this benchmark needs the recording-disjoint split, the audit of §5 and stratified, clustered reporting together.

7 RECOMMENDATIONS AND RELEASED MATERIAL

§5.4's rule in operational form is a five-item protocol: partition by recording; **audit the split before trusting it**, reporting the adjacent-to-clean ratio rather than a pass/fail bit; if the audit fails, treat the clean baseline as inflated rather than discarding the benchmark; **report protocol-matched numbers for any leakage claim**, since $B - A$ alone is not a leakage estimate and no fixed offset repairs it (§4.1); and report size-stratified, recording-clustered means with recording-paired tests (§6.3). A.8 states each item in full, A.5 what each rests on and costs.

Cost. The audit is CPU-only and needs no model; protocol-matched evaluation is two inference passes over checkpoints the comparison already produced. Only item 5 is expensive, at 60 fold-runs (A.5).

Release. We release the near-duplicate audit, the protocol-matched evaluation driver, the seeded split generators, the split files evaluated here and the result files behind every table. The masks, the cross-validation folds and the per-recording leaderboard results belong to the concurrent benchmark (Anonymous, 2026); we include the latter two only so that reviewers can recompute Table A2

and Table A7. Frames come from the benchmarks' authors, and we redistribute none. What CST reproduces is the measurement failure and its detection, not a clean-reference leakage estimate (A.6).

8 LIMITATIONS

The matched gain is a contrast, not a decomposition. At a matched clip budget the leaky checkpoint holds three advantages at once — the test recordings, more distinct recordings, and epoch selection on a contaminated validation set — and the third arm that would separate them we did not train, so +6.4 / +8.5 / +8.8 is their sum. A.6 states this and every limitation below in full.

Table 1 is seed 42. Re-scoring the four configurations with a matched pair of checkpoints at further seeds keeps the gain positive over ten runs, from +5.6 to +9.3, with infrared and visible captured fractions not overlapping (A.11); that covers four of fifteen entries and no CST entry, so the captured fractions order three settings rather than estimate them.

The audit is one representation at one threshold. Another feature space could rank the five splits differently, and the target-layout variant does (A.2); the healthy verdicts do not survive a 0.95 cut while the CST failure does. The grid sweep and the pair inspection were run on Anti-UAV only, where a finer grid strengthens the verdict (A.2).

One draw, six recordings, three settings, one domain. Each protocol is one seeded partition, never resampled, and only for CST do we read the contamination as the dataset's, in the one draw we audited (§5.3). The clean test set is six recordings, so §4's levels are descriptive and the direction rests on the 15 consistent entries. The +18.0-point night gain rests on two held-out night recordings, and the visible cross-validation over 13 finds no small-target night deficit to erase: read $0.73 \rightarrow 0.91$ as a stratum-level weakness, not a modality-level one.

Every mask here is a SAM 2 pseudo-label prompted from the benchmarks' boxes and never validated against human labels, and a mask is close to a deterministic function of its box. Both protocols draw on one label pool per benchmark, but their test sets differ and we did not test label quality against exposure, so part of what a leaky checkpoint memorises may be the labeller (A.6).

The mechanism is unresolved at the frame level. On the six held-out infrared recordings a frame's similarity to the *clean* training pool shows no detectable association with its gain (recording-clustered Spearman +0.11, ± 0.37); power is low and the predictor is not exposure itself (A.10).

REFERENCES

Anonymous. Concurrent submission (title withheld for double-blind review), 2026. Under review.

Björn Barz and Joachim Denzler. Do we train on test data? purging CIFAR of near-duplicates. *Journal of Imaging*, 6(6):41, 2020. doi: 10.3390/jimaging6060041.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems (MLSys)*, volume 3, pp. 747–769, 2021.

A. Colin Cameron and Douglas L. Miller. A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2):317–372, 2015. doi: 10.3368/jhr.50.2.317.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Ravi B. D. Figueiredo and Hugo A. Mendes. Analyzing information leakage on video object detection datasets by splitting images into clusters with high spatiotemporal correlation. *IEEE Access*, 12: 47646–47655, 2024. doi: 10.1109/ACCESS.2024.3383047.

Noam Glazner, Noam Tsfaty, Sharon Shalev, and Avishai Weizman. Find the leak, fix the split: Cluster-based method to prevent leakage in video-derived datasets. *arXiv preprint arXiv:2511.13944*, 2025. Accepted to ICSEE 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Pavel Iakubovskii. Segmentation models pytorch, 2019. `https://github.com/qubvel/segmentation_models.pytorch`.

Nan Jiang, Kuiran Wang, Xiaoke Peng, Xuehui Yu, Qiang Wang, Junliang Xing, Guorong Li, Guodong Guo, Qixiang Ye, Jianbin Jiao, Jian Zhao, and Zhenjun Han. Anti-UAV: A large-scale benchmark for vision-based UAV tracking. *IEEE Transactions on Multimedia*, 25:486–500, 2023. doi: 10.1109/TMM.2021.3128047.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Shachar Kaufman, Saharon Rosset, Claudia Perlich, and Ori Stitelman. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6 (4):1–21, 2012. doi: 10.1145/2382577.2382579.

Boyang Li, Chao Xiao, Longguang Wang, Yingqian Wang, Zaiping Lin, Miao Li, Wei An, and Yulan Guo. Dense nested attention network for infrared small target detection. *IEEE Transactions on Image Processing*, 32:1745–1758, 2023. doi: 10.1109/TIP.2022.3199107. NUDT-SIRST dataset.

Hanchao Li, Pengfei Xiong, Jie An, and Lingxue Wang. Pyramid attention network for semantic segmentation. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2018.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *PMLR*, pp. 5389–5400, 2019.

Annika Reinke, Minu D. Tizabi, Michael Baumgartner, Matthias Eisenmann, Doreen Heckmann-Nötzel, A. Emre Kavur, Tim Rädsch, Carole H. Sudre, et al. Understanding metric-related pitfalls in image analysis validation. *Nature Methods*, 21(2):182–194, 2024. doi: 10.1038/s41592-023-02150-0.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

Domonkos Varga. On the evaluation protocol of gesture recognition for UAV-based rescue operation based on deep learning: A subject-independence perspective. *arXiv preprint arXiv:2602.17854*, 2026.

Jihwan Woo. The frame-level leakage trap: Rethinking evaluation protocols for intrinsic image decomposition, with source-separable uncertainty as a case study. *arXiv preprint arXiv:2605.06359*, 2026.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

Bin Xie, Congxuan Zhang, Fagan Wang, Peng Liu, Feng Lu, Zhen Chen, and Weiming Hu. CST anti-UAV: A thermal infrared benchmark for tiny UAV tracking in complex scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 6216–6225, 2025. doi: 10.1109/ICCVW69036.2025.00647.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Ekin Yagis, Selamawet Workalemahu Atnafu, Alba García Seco de Herrera, Chiara Marzi, Riccardo Scheda, Marco Giannelli, Carlo Tessa, Luca Citi, and Stefano Diciotti. Effect of data leakage in brain MRI classification using 2D convolutional neural networks. *Scientific Reports*, 11(1):22544, 2021. doi: 10.1038/s41598-021-01681-w.

Mingjin Zhang, Rui Zhang, Yuxiang Yang, Haichen Bai, Jing Zhang, and Jie Guo. ISNet: Shape matters for infrared small target detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 867–876, 2022. doi: 10.1109/CVPR52688.2022.00095. IRSTD-1k dataset.

Wenhao Zhang, Zhongliang Zhou, John Kang, and Sheng Li. Auditing data leakage in whole-slide image multimodal benchmarks. *arXiv preprint arXiv:2607.12278*, 2026.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

USE OF AI ASSISTANCE

Per the ICLR 2027 AI policy we disclose the following. This section does not count toward the page limit.

Direction and responsibility. The research question, the protocols, the decisions about what this paper claims and how strongly, and the revisions made in response to two rounds of independent pre-submission review were the authors'. The frames of both video benchmarks are their authors', and the segmentation masks and partitions come from the concurrent work cited in §3.1; the compute and the earlier segmentation model referred to as *ours* are this paper's authors'. The figures of this paper are the authors' own TikZ sources.

Where an AI assistant was used. An LLM-based coding assistant was used substantially. It implemented the analysis and evaluation code — the near-duplicate screen, the cross-evaluation driver, the cross-validation and paired-test scripts, the per-recording and per-frame exports, the representation sweep and matched-pair inspection of A.2, and the scripts that build the tables from the result files — and ran those programs on the authors' machines at the authors' direction. It drafted and revised the text of this paper, including this section, under the authors' editing. It verified the bibliography against arXiv, Crossref and publisher records, checked numerical consistency between the text, the tables and the result files, and checked the manuscript's factual claims against the repository and the data; corrections that check surfaced, including to the description of the data's provenance, were made by the authors. Earlier draft figures produced by the assistant were replaced by the authors' own.

What was not produced by a model. Every number reported here is the output of code executed on the authors' hardware over the benchmark data. No result, table entry or figure value was generated, estimated or completed by a language model. The released result files reproduce every number in the paper, and the released code reproduces those files from the data.

Model use inside the method. Segment Anything 2 was prompted with each benchmark's released bounding boxes to obtain the segmentation masks that neither video dataset ships (§3.1). This is part of the experimental setup rather than of the writing process.

Responsibility. The authors read the submission, checked its numbers against the released files, and are responsible for its contents.

A APPENDIX

Three subsections read the three appendix tables, a fourth records the procedural detail that §3 states only in summary, A.5 and A.6 hold the qualifications that §7 and §8 cite, and A.7 holds what §4 and §6 state in summary form. No conclusion of §4–§6 changes if the appendix is removed, but several earlier claims are sourced here and nowhere else: the target-layout figures of §3.4 and the unrounded null medians of §5.1 (Table A4), and the per-architecture, per-seed and per-cell detail behind §4 and §6 (A.7). A.1, A.3, Table A2 and Table A7 reproduce the concurrent benchmark's leaderboard and its cost context for reference. The target-layout variant is the only measurement that appears here for the first time.

A.1 VISIBLE-BAND CROSS-VALIDATED LEADERBOARD (TABLE A3)

The cross-validated leaderboard of this section, of Table A2 and of Table A7 is the concurrent benchmark's result (Anonymous, 2026), reproduced here for reference; none of it is claimed as a contribution of this paper. Table A3 covers the same six ResNet-34-family architectures as Table A2; the three modern configurations of §3.3 were trained on CST only and are cross-validated on neither band (the eight architectures of Table 1 are the cross-evaluation's, not this leaderboard's). It supplies the six per-model rows and the medium and large columns behind the concurrent work's visible-band result.

The small tercile separates into the same two clusters — DeepLabV3+ 0.711, FPN 0.708, PAN 0.687 against PSPNet 0.668, ours 0.666, U-Net 0.664 — but adjacently rather than apart: the cluster-mean gap is +3.6, against +10.6 on infrared, and the worst pyramid model exceeds the best plain one by 1.9 points as printed (1.95 unrounded) against 7.9. The $\pm$ are marginal recording-clustered intervals and overlap for every pair in the table (DeepLabV3+ 0.711 $\pm$ 0.083 against U-Net 0.664 $\pm$ 0.094); they bound each model's own level, not a difference. The only inferential support on this band is the single paired row that was run here, FPN $-$ ours +4.1 [+1.3, +7.0] (Table A7, last row), whose reference is our own FCN — so the visible cluster claim is weaker than the infrared one by construction; A.7 gives that row's sign-test support and its day/night split. Rank inside a cluster with care: the three plain models span only 0.4 points here (0.664–0.668) and reverse their order relative to infrared, where they span 3.2 (Table A2), and the top two pyramid models swap within 0.3 points. Outside the small tercile the cluster means differ by +0.6 (medium) and +0.8 (large), the whole spread being PSPNet (0.818, 0.762); the other five sit within 0.6 points of each other in both columns (0.5 as printed). The day/night placement of the small-target gap also inverts relative to infrared (+5.0 by day, n_rec = 18; +1.7 at night, n = 13; against +5.9 and +18.2): at these sample sizes we read both as the direction of the effect. Tercile membership gives n = 31 / 34 / 32 against the 37 recordings in the pool, because a recording enters a tercile only if it has frames there (A.7) and the visible terciles use this band's own cut-offs; membership is checkable in the released per-recording CSVs. The medium column peaks at 34 of 37, and the three recordings absent from it do contribute elsewhere: `20190925_152412` (741 frames, all small), `20190925_194211` (706, all small) and `20190926_111509` (805, all large). The same holds either way — each of the six recordings absent from the small tercile and each of the five absent from the large one has frames in another — and the union over the three terciles is the full 37, matching the released folds. All six architectures agree on tercile membership and on the per-recording frame counts (`results/cv_per_recording_visible.csv`).

A.2 NEAR-DUPLICATE AUDIT, FULL DETAIL (TABLE A4)

Table A4 is Table 2 with every measure and both thresholds; two of its columns and its extra rows do work Table 2 cannot. The `>0.95` column is the one §5.3(ii) reads: the healthy verdicts do not survive it (IR 52.1 against adjacent 68.3; VIS 41.7 against 55.0) while the CST failure does (80.8 against 80.1), and the per-pair null is no longer empty there (0.4% on IR, 0.2% on IRSTD-1k) where at 0.99 it is 0.0% everywhere — 0 of 10,323 unrelated infrared pairs against 0.4% of those same pairs at 0.95 — so shared low-texture background alone reached 0.99 in none of the pairs the null enumerates. That is a per-pair statement about a deterministic pairing, whereas the cross-split score is a maximum over

up to 20,646 candidates per test frame, so it does not calibrate the rate at which such a maximum clears 0.99 on background similarity alone. The operating point was fixed after the audit had been run: of the two thresholds we evaluated, 0.99 is the lowest at which the per-pair null is empty on all five datasets, and no intermediate value was tried. The healthy verdicts depend on that choice, which is why both thresholds are printed here rather than the chosen one alone. We swept the representation on the two Anti-UAV bands, holding the split, the stride and the 0.99 threshold fixed; the released audit takes a `--size` flag and the six runs are in `results/nearcopy_resolution_sweep.csv`. Every rate falls as the grid gets finer — infrared clean 10.4% at 40×32, 6.9% at 80×64, 3.4% at 160×128; visible 5.9%, 0.5%, 0.3% — so a near-copy rate is a quantity of its representation and has to be read against references computed in the same one, not as an absolute duplicate count. The references move with it, and the adjacent-to-clean ratio rises monotonically (infrared 4.4× → 5.5× → 10.0×, visible 6.0× → 56.3× → 78.9×), so of the three grids 80×64 is the more conservative choice for a healthy verdict: coarser weakens it, finer strengthens it. The random-pair null stays empty at 0.99 at all three resolutions, so the operating point is not an artefact of the grid. Two limits survive the sweep. The like-for-like clean/leaky statistic is the less stable of the two on the visible band (55% at 40×32 against 13% at 80×64 and 18% at 160×128), which is one more reason we report the ratio alongside it rather than either alone. And we swept only the two Anti-UAV bands: the CST verdict and the two still-image controls are measured at one resolution, so nothing here shows the *failing* verdict is resolution-robust. We have also not audited video benchmarks with moving cameras or large objects, where a frame is less completely its background and adjacent-frame correlation may be lower (§5.3(iii)).

What the infrared matches actually are. We inspected the ten highest-similarity cross-split pairs on the infrared clean split (`code/inspect_nearcopy_pairs.py` is released and writes both a pair list and a side-by-side contact sheet; we release the list, `results/nearcopy_pairs_infrared.csv`, but not the sheet, which would redistribute benchmark frames). All ten pair two recordings against three others made in the same session minutes apart — the test frames carry burnt-in timestamps of 18:40 on 2019-09-26 and their nearest training frames 18:38–18:39 — at azimuths about 11° apart. The frames themselves are near-featureless thermal sky with a small bright target, so the correlation that clears 0.99 is carried by an almost uniform background rather than by a distinctive scene recurring under two identifiers. Two things follow. The infrared 6.9% is dominated by low-texture frames from adjacent sessions, which is why we read the band as healthy rather than as containing duplicated content; and the screen cannot separate *same scene* from *both blank* on footage of this kind, which is the concrete form of the assumption stated above that a frame is largely its background. Over the whole split the 325 matching test frames fall into 50 distinct recording pairs, concentrated in two same-session clusters. We did not run this inspection on CST, so §5.3(ii)'s reading of its 72.5% remains what the screen assigns rather than a verified duplicate count.

The `p90` column shows the verdicts are not median artefacts: the 90th percentile of each healthy clean split sits below the threshold (IR 0.9864, VIS 0.9833, IRSTD-1k 0.9617, NUDT-SIRST 0.9649), so its near-copies are confined to the top decile, whereas CST's 90th percentile is already 0.9992. The column bounds that tail rather than adding independent evidence: p90 < 0.99 follows arithmetically from a `>0.99` rate under 10%. The null medians in the same table are the unrounded source for the nearly eight-fold background spread of §5.1 (0.3627 / 0.0470 = 7.7) — 0.3627 (IR), 0.2410 (VIS), 0.0541 (CST), 0.3392 (IRSTD-1k), 0.0470 (NUDT-SIRST); Table A4 prints four decimals.

Levels are not comparable down a dataset block, nor across blocks. Rows with `--` in `n_train` are single-pair statistics, whereas the clean and leaky rows are maxima over 663–20,646 training candidates and sit higher for that reason alone; within a block the two rows are computed over different populations, so the mismatch carries no guaranteed direction — the candidate-count effect alone would depress adj/clean and so make a clean verdict harder rather than easier to reach, but we do not assert that as a bias direction (§5.1) — which is why the like-for-like clean/leaky ratio is reported alongside adj/clean, and across blocks the maximum grows with the candidate count, which lifts the two Anti-UAV rows relative to the still-image ones and holds CST's 1,540-candidate rate down. The adjacent-frame reference is for the same reason a single-neighbour reference rather than either a ceiling or a bound: a frame-level split usually places several neighbours in training and the nearest of them can exceed the single successor, as it does on CST — 86.9% under its frame-random split against 70.6% adjacent (Table 2) — but nothing guarantees that any particular frame's neighbour lands in training, so on another split the frame-random rate can fall below the

adjacent one instead. The adjacent rows are capped at 4,000 pairs. The null rows are formed from the clean training pool by splitting it in half and pairing element *i* with element *i*+*h* — a deterministic, systematically cross-recording pairing rather than a random draw of pairs — giving $\lfloor$n_train/2$\rfloor$ = 10,323 / 10,073 / 400 / 331 on IR, VIS, IRSTD-1k and NUDT-SIRST; on CST the null alone was run with a 600-pair cap, i.e. from a 1,200-frame subsample of the 1,540 training frames. The CST clean and leaky rows are full enumerations over all 323 and 411 test frames; the 600-pair cap applies to the null construction alone and does not touch them. The clean/leaky ratio of §5.2 is therefore a ratio of two rates each measured on its own test set — 323 clean against 411 leaky frames on CST, 4,734 against 4,223 on infrared and 4,375 against 4,055 on visible — and not a before-and-after over a common set of queries. Clean candidate pools are 20,646 / 20,147 / 1,540 / 800 / 663; the leaky rows use 20,813 / 19,870 / 1,545, and the two still-image sets have no leaky row.

On video the audit separates the two protocols. On Anti-UAV it separates the leaky split from the clean one by 7.9× (VIS) and 2.3× (IR), and both from the adjacent reference; the inability to separate them is specific to CST (§5.3(ii)).

The two Anti-UAV bands audit the same partition: the same six held-out recordings and the same training recordings are used for both (§3.1–3.2), the pairing being at the recording and clip level rather than a pixel registration we check, so their 6.9% against 0.5% is a property of the modality's appearance — thermal backgrounds correlate more readily, random-null median 0.363 against 0.241 — and not of the split. A duplicated recording should appear in both bands, so — on the untested assumption that the screen is comparably sensitive on the two modalities — we read the visible-band 0.5% as the better indication of partition-level duplication and most of the infrared 6.9% as cross-recording thermal-background similarity. In counts, 0.5% is about 20 of 4,375 frames from six held-out recordings and 6.9% about 325 of the 4,734 frames of the same six.

The two still-image sets validate the tool where the answer is known in advance. IRSTD-1k and NUDT-SIRST have no temporal structure, so "adjacent" images — consecutive in the authors' list order — should behave like unrelated pairs, and the audit returns that to within the near-copy criterion: adjacent medians of 0.341 and 0.101 against random medians of 0.339 and 0.047 (the NUDT list order carries a mild correlation that never approaches 0.99), and 0.0% adjacent near-copies on both, identical to the null. These rows show that unrelated pairs of still infrared images do not reach 0.99. The control for the maximum-over-training statistic on video is the VIS clean split, whose maximum over 20,147 candidates still returns 0.5%. The same still rows show the screen is not blind on these sets — 2.5% and 4.8% of test images have a $>$0.99 match, roughly 5 of 201 and 32 of 664, none of them inspected individually — so its 0.0% on adjacent stills is not insensitivity.

Those 2.5–4.8% above the null we read as incidental duplicates of the kind documented on CIFAR (Barz & Denzler, 2020; §2), a different and smaller problem than temporal leakage, although the per-pair null cannot formally exclude coincidental matches at this rate and we have not inspected them individually. The two medians behind the clean-versus-null check are not comparable within either still set: IRSTD-1k's clean 0.8772 is a maximum over 800 training candidates and NUDT-SIRST's 0.7642 a maximum over 663, against single-pair null medians of 0.3392 and 0.0470 (§5.3(iii)).

The target-layout rows are a different statistic wearing the same threshold: similarity is $1 - \min(d, 1)$ for the Euclidean distance d in (c*x*/W, c*y*/H, $\sqrt{\mathrm{area}/(HW)}$), as implemented in the released audit script, so "$>$0.99" means a nearest training target within 0.01 of a three-dimensional layout space, not a correlation of 0.99. In three dimensions against 5,120, on a benchmark whose median target is 0.24% of the frame (§3.1), it saturates: every dataset reaches at least 96% at 0.95 (96.0–99.8), and at 0.99 the ranking inverts to IR 87.8%, VIS 86.4%, CST 35.1% against the pixel ranking CST 72.5% $\gg$ IR 6.9% $>$ VIS 0.5%. The statistic measures where targets are, not whether frames were seen, so these are not contamination figures (§3.4). Two regimes were not measured, which bounds what the table can be asked. No leaky-split layout row exists, so the supported claim is that the variant inverts the cross-dataset ranking and saturates, not that it fails to separate protocols. And no frame-random regime was audited on Anti-UAV: the 87% quoted for frame-random in Fig. 1b is CST's frame-random rate (86.9%), the only such split we audited, while the Anti-UAV leaky rows are clip-random (16.0% IR, 3.6% VIS). Layout row counts fall short of the pixel rows because a frame with an empty or missing mask has no layout vector (one test frame on each video set; 40 training frames on VIS, 13 on CST).

The audit's rates are what §5.4 reads against the matched gain, which lies within about 2.5 points across datasets with per-architecture values from +4.9 to +15.6 (Table 1); cf. the $A'' - A$ cells of §4.1.

A.3 Architecture cost (Table A5)

Parameters, GFLOPs and throughput at 512×640 for the six cross-validated architectures; the three modern CST configurations have no cost row, so no cost claim is made about them. Parameters span 21.44–24.44 M, a 14% band, and neither cost axis orders the small-tercile clusters. The three winners are in fact the more expensive models in FLOPs (68.8–79.3 against a plain-FCN mean of 50.2, pyramid mean 74.3), and two counterexamples inside that pattern break any cost reading. PSPNet is the cheapest (23.6 GFLOPs) and the fastest (378.3 FPS) and is last of six on the infrared small tercile at 0.609 — though it is the best plain model on the visible small tercile, so this is an infrared statement. U-Net has the most parameters and the lowest throughput (24.44 M, 141.0 FPS) at 78.6 GFLOPs and still sits in the losing cluster, 10.0 points below FPN (the paired FPN − U-Net of Table A7), which costs fewer GFLOPs (68.8). Params and GFLOPs are the load-bearing columns. FPS was measured once, with device, batch size, precision, warm-up and iteration count unrecorded, so we read only its ordering and quote no ratios from it; GFLOPs come from a single counter, so the FLOPs comparison holds within that counter only.

A.4 Implementation details

The procedures below are the released scripts (§7) and the settings they were run with; the audit and the cross-evaluation run without training, on the released split files and per-recording CSVs.

Training recipe (§3.3). All models are trained with AdamW (backbone learning rate 5e-5, decoder 5e-4, weight decay 1e-4), a cosine schedule, 30 epochs, batch 16, 512×640 input, cross-entropy with foreground weight 15 plus Dice loss, and mixed precision, with model selection on validation UAV-IoU. The recipe is identical under both protocols; the three modern CST configurations use it unchanged, which is why they are reported for the leakage delta only (§3.3). The five classic decoders are the `segmentation_models_pytorch` implementations (§3.3).

Split sizes not quoted in §3.2. The clean stride-10 training pools are 20,646 (Anti-UAV infrared), 20,147 (Anti-UAV visible) and 1,540 (CST), against the leaky 20,813, 19,870 and 1,545: this is the comparison behind the statement in §3.2 that stride-10 training-set sizes are within 1.5% of each other in every setting. The Anti-UAV clean test sets are 4,734 (infrared) and 4,375 (visible) stride-10 frames, CST's 323. The CST leaky split is on a different scale: 14,480 / 3,177 / 3,090 are the frame-random split's own train / validation / test sizes, not the stride-10 quantities of Table A4's CST leaky row (1,545 train, 411 test). The leaky splits are implemented as a symlinked directory tree, so the loader is unchanged (§3.2).

Recording and clip naming (§3.1). Anti-UAV clips are numbered within their recording (e.g. `20190925_111757_1_1`, `_1_2`, ...), and the hour in the recording name is what supplies the day/night label used for stratification and for the day/night columns of §4.4 and Table A2.

Audit. The representation is §3.4's; what §3.4 omits is that the 80×64 grayscale frame is 5,120 float32 dimensions, that enumeration is complete on both sides — so the reported maximum is the true nearest neighbour of that split, though not comparable across splits of different size (A.2) — and that the largest problem is one dense 4,734 × 20,646 product evaluated in chunks of 64 test rows over a 423 MB training embedding, in numpy on CPU with no GPU. The null and adjacent constructions and the layout formula are those of the released script (A.2). We did not instrument wall-clock time, and we make no wall-clock claim anywhere: the cost we state is this problem size and the pair counts of A.5, *Cost*.

Cross-evaluation. All four cells of Table 1 come from one re-evaluation pass of four scorings per architecture, two of them extra relative to the training-time evaluations; there is no retraining and no fine-tuning, and day/night columns are kept per row. Seed 42 throughout; the pass differs from the training-time evaluations by at most 0.24 points on the visible band and by zero on infrared, for the cache reason set out in A.7. Both checkpoints were selected on their own protocol's validation set (§3.3). The leaky validation set draws from 26 recordings that all appear in leaky train and contains 6 of the 50 clean-test clips (Table A1), so contaminated epoch selection is a third ingredient of the

matched gain alongside the two named in §4.5; like them it is unavailable under a recording-disjoint split, so it is reported inside the gain rather than separated from it.

Cross-validation. The folds are the concurrent benchmark's (Anonymous, 2026); they are built deterministically: recordings are keyed from the sequence name and assigned round-robin within each day/night label (§3.1), so the partition is fixed and reproducible. Its variance is not thereby removed but unmeasured — Table A2 rests on one assignment of 37 recordings to 5 folds, the intervals do not include fold-assignment variability, and we did not repeat the CV under a second assignment. §3.4's 25 epochs per fold — against §3.3's 30 — come with no validation list, hence no checkpoint selection: the last-epoch model is evaluated, so the leaderboard is a fixed-budget comparison and the small-tercile gap is not separable from a convergence-rate difference. Frame IoU is accumulated only over frames with a non-empty mask and averaged within a recording (released CV runner). That runner is resumable: it skips training whenever a fold checkpoint already exists, saves weights alone, sets and resets no per-(fold, model) RNG, and encodes neither the seed nor the epoch, stride or configuration in the checkpoint name. A resumed run is therefore not guaranteed to be identical to one trained end to end, and a changed epoch budget silently reuses an existing checkpoint. Table A2 and Table A3 come from runs at the settings stated here, but the released runner does not itself record that; a future run should write per-(fold, model, seed) RNG state and the run configuration alongside each checkpoint. Besides putting CV levels below the pooled single-split numbers and out of comparison with them (§3.4), that metric change is not neutral: per-frame averaging scores a frame with a missed target as 0, where pooled UAV-IoU is foreground-union weighted, each frame counting in proportion to its share of the summed union (which on a target-absent frame is false positives alone; §3.4), so §6.4's contrast between 1.1 points pooled and 10–13 stratified reflects the stratification and the metric change together. The stratification carries it on its own: read without terciles, the same folds and the same recording-averaged metric put the six models within 2.2 points with every interval overlapping (A.7), so the separation is what the aggregation rule hides, not an artefact of changing the metric.

Paired test. For each cell we form per-recording mean IoU for both models over the recordings both were evaluated on, then take the mean difference, a 95% t-interval over those differences, and a two-sided exact sign test from which exactly tied recordings are dropped, as is conventional, so a tie reduces the count rather than scoring against either model; at most two recordings tie in any cell we report. The interval responds to the size of the per-recording gains and the sign test only to their count, so the two can disagree when a few recordings dominate the mean; A.7 gives the instances. Table A7 reports 15 paired cells (5 pairs $\times$ 3 terciles) and A.7 quotes further day/night cells; no multiplicity correction is applied to the individual p-values, and requiring both a CI excluding zero and a significant sign test, as we read the reproduced leaderboard, is the only guard against that — it is what the medium and large cells fail.

Aggregation and clustering (§2, §3.4). Treating frames as independent observations would understate the standard error and inflate apparent significance (Cameron & Miller, 2015; Miller, 2024). The same aggregation argument is made for hierarchical image-analysis data in (Reinke et al., 2024). Recording-level clustering is complementary to the benchmark-variance analysis of Bouthillier et al. (2021), which addresses how many runs to average over seeds and data sampling rather than how to aggregate correlated evaluation units. Size-stratified reporting follows the observation that aggregate IoU on tiny-target datasets is dominated by the easy strata.

Reproducibility of the audit. Of the five rows of Table 2, a third party can recompute CST, IRSTD-1k and NUDT-SIRST from public data with the released script; the two Anti-UAV rows are recomputable once that benchmark's frames are obtained from its authors, for which we release the split files, the script and the per-recording results. CST's SAM 2 masks (§3.1) are a prerequisite for its segmentation training and cross-evaluation, not for the audit, which reads frames only (A.5); for them we release the prompting script, whose documented invocation names the checkpoint used (`sam2.1_hiera_large`, config `configs/sam2.1/sam2.1_hiera_l.yaml`), and not the masks it produces: the benchmark states no dataset licence, so a reader regenerates them from frames obtained from its authors.

A.5 Notes on the checklist of §7

§7 states five procedures; this section holds what each rests on, where each stops working, and what each costs.

Item 1: the unit is a hypothesis. The covariate to stratify on is day/night on Anti-UAV; CST encodes no hour and is split by sequence alone (§3.2). The unit names a factor you believe is shared, and the belief can hold while the factor does not: on CST the identification held — each sequence is one continuous recording — but the factor we cared about did not, since near-identical sites and camera views recur under different sequence identifiers (§5.3(ii)).

Item 2: references, threshold, and what a pass shows. Compute both references — the random-pair null and the within-clip adjacent-frame rate — in the same representation at the same threshold, because a raw similarity is uninterpretable on its own: unrelated frames already correlate at a median of 0.36 on Anti-UAV IR against 0.05 on NUDT-SIRST (Table 2). Set the threshold from your own null rather than inheriting ours, at the lowest value where the null is empty; of the two thresholds we evaluated, 0.99 is the lowest at which it is empty on all five datasets, whereas at 0.95 the infrared null still fires on 0.4% of pairs (Table A4). A pass is a verdict at this measure's resolution: at 80×64 the background is expected to dominate a target under 1% of the frame, so a pass says the screen found no training frame sharing a test frame's background and camera view — not that nothing is memorisable, and not that the split has been certified free of duplicates (§5.1, §5.3(ii)). It also assumes a frame is largely its background; on moving-camera or large-object footage the adjacent reference itself falls and the ratio may be uninformative rather than reassuring (§5.3(iii)), and without sequence structure it degenerates to 0% and the check becomes clean-versus-null (§5.3(iii)). Do not substitute the target-layout variant: on tiny targets it saturates and inverts the ranking, scoring the healthy Anti-UAV splits at 87.8% and 86.4% against CST's 35.1% (Table A4, A.2).

Item 3: what a failure licenses. The like-for-like statistic is needed because the adjacent reference is a single-neighbour statistic measured on a different population from the maximum it is divided into, so the adj/clean ratio carries no guaranteed direction (§5.1, A.2). Treating the clean baseline as inflated points the matched gain downward relative to a genuinely held-out test set (+8.8 there) — though only under an assumption we cannot check, that the uncontaminated partition would cost the clean checkpoint more than the leaky one (A.6); we therefore leave the CST gain unbounded below and read it, like every matched gain, as a fixed-test protocol contrast rather than a bound on memorisation, for the reason in item 4. We keep CST as our public reproduction vehicle on exactly these terms.

Item 4: matching the budgets, and what the gain is. Match the training budgets first, since a budget difference enters $\mathrm{B}' - \mathrm{A}$ as data volume (ours: 225 versus 223 clips, frame counts within 1.5%; §3.2, A.4). The clean test set is the common set to report on because it alone is built entirely of recordings the clean checkpoint never saw (§4.2); no fixed offset repairs $\mathrm{B} - \mathrm{A}$, the confound being a property of the checkpoint–test-set pair (§4.1). $\mathrm{B}' - \mathrm{A}$ is a fixed-test protocol contrast, not a bound on memorisation: the leaky checkpoint also sees more recordings and selects its epoch on a validation set whose 48 clips include 6 of the 50 clean-test clips (§4.2; A.4), and since we measure neither the coverage nor the selection term, the contrast bounds memorisation from above only if both are non-negative — an assumption we cannot check. Set it against whatever seed-to-seed variability you have actually measured, which bounds interpretation from below without amounting to a significance test: ours is seed 42, so for eleven of the fifteen entries that variability is unmeasured, and across the four configurations we did re-run under both protocols the conventional delta moves by at most 1.3 points and A by at most 0.7, against a smallest matched gain of +4.9 (§4.5; enumerated in A.7, qualified in A.6).

Item 5: the unit of sampling. Bin held-out frames into target-size terciles, compute IoU per frame, average within each recording — a recording enters a tercile only if it has frames there, which is why the terciles carry different n (34, 37, 35 on infrared; Table A2, A.7) — then take means and 95% t-intervals over per-recording means. The conjunction of CI and sign test is deliberately conservative, not a procedure with a controlled error rate, and it applies no multiplicity correction over the 15 pair $\times$ stratum tests of Table A7. In our data a pooled aggregate hid the one stratum on which architectures differed.

Cost. The audit is CPU-only — plain numpy, runnable alongside a training job — and costs n_test $\times$ n_train dot products of 5,120-dimensional vectors: 9.8×10^7 pairs on Anti-UAV IR (4,734 $\times$ 20,646), 8.8×10^7 on VIS and 5.0×10^5 on CST, full enumeration without sampling, against references orders of magnitude smaller (4,000 adjacent and 10,323 / 10,073 null pairs on the two bands; 1,386 and 600 on CST; Table A4). Protocol-matched evaluation's two extra inference passes per architecture run over checkpoints any protocol comparison has already produced, against 30 training epochs per architecture per protocol (§3.3); it retrains nothing, but unlike the audit it needs

a GPU, the frames and the saved checkpoints. Cost grows as the product of the two set sizes, so a benchmark ten times larger on both sides is a hundred times this and should be indexed rather than enumerated. Recommendation 5 is the expensive item: ours is 5 folds $\times$ 6 architectures $\times$ 2 bands $\times$ 25 epochs — 60 fold-runs.

Release. The inventory is the two training scripts implementing the §3.3 recipe, the near-copy audit, the cross-evaluation driver, the CV fold builder and runner, the recording-paired test, the seeded split generators (Anti-UAV recording-disjoint and clip-random, CST sequence-disjoint), the box-to-mask conversion tool, the Anti-UAV split and CV fold files, the per-recording cross-validation results for both bands (six models $\times$ five folds $\times$ 37 recordings $\times$ size tercile), the audit lists for the two still-image controls, and the per-architecture, per-stratum and per-dataset result files behind every table in the paper together with the paired-test log — the size-stratified, recording-clustered leaderboard for six architectures on both bands included (Table A2, Table A3; the concurrent benchmark's results, shipped for recomputation) — so every published number is traceable to a released file. The CST path depends only on public data, but we redistribute none of it: the benchmark states no dataset licence, so we release the code that rebuilds our partition and prompts the masks, and the sequence names our split lists refer to, rather than any frame or mask of it. A reader obtains the benchmark from its authors under whatever terms they set. A reader can run the audit as soon as the CST frames are laid out under the released split lists — the audit reads frames only, never masks, so the SAM 2 mask step is not a prerequisite for it, and it needs no GPU — and the protocol-matched pipeline after training the two checkpoints they would need anyway.

A.6 Extended limitations

The matched gain is a contrast, not a decomposition. §4.5 says what the gain contains; two consequences belong here. At a matched clip budget (§3.2) the leaky-trained checkpoint holds three advantages at once: it trained on 35 of the 50 clean-test clips, it selected its epoch on a validation set holding 6 more, and it saw more distinct recordings — 37 against 26 on Anti-UAV (Table A1), 220 sequences against 154 on CST. Separating them needs a third arm — leaky training on 26 recordings that include the six clean-test ones (20 clean-train plus the 6 held out), so coverage is matched in count but not in identity — which we did not train, so we apportion +6.4 / +8.5 / +8.8 (per-architecture +4.9 to +15.6) between memorisation, model selection and coverage not at all. Reading it as an upper bound on memorisation alone would need both the coverage term and the selection term to be non-negative — plausible at a matched budget (37 recordings shallower against 26 deeper), but neither is something we measure, nor is their interaction, so we assert no bound in either direction. On CST, §5.3(ii) points the same +8.8 downward relative to a genuinely held-out test set, because the clean baseline A is inflated by cross-sequence twins; that reading assumes an uncontaminated partition would cost the clean checkpoint more than the leaky one, which has trained on all 220 sequences and would lose that exposure as well, so we leave the CST number unbounded on that side. Two checkpoints buy the contrast at no training cost; the decomposition costs a third run whose result we have not seen, so we describe it rather than prescribe it.

Seeds: every cross-evaluation cell is seed 42. The indirect evidence bounds little: the observed range of B $-$ A is at most 1.3 points over the configurations with extra seeds, and that 1.3 is our own FCN, which is not a Table 1 architecture — the widest range over Table 1 architectures is 1.0 (PAN on infrared, two matched seeds), the widest over three visible seeds is 0.7 (U-Net), and only PAN and FPN have extra infrared seeds at all — while A ranges by at most 0.7 (§4.5; the per-configuration values are in A.7). These are ranges over two or three runs, not bounds. The captured fractions inherit that and have no interval of their own: the widest Table 1 seed range in the numerator alone moves the visible 68% by about 8 points, and our own FCN's 1.3-point range would move it by 15; the CST -17% rests on a -1.5-point mean whose per-architecture entries run from -3.7 to +3.7 (Table 1). We report them as an ordering of three settings, not as estimates with precision. The size-stratified leaderboard carries the symmetric weakness, which §6.4 defers here: §6.2 shows a ranking that survives three seeds under one protocol and changes sign under the other, whereas Table A2 is one training run per model per fold (30 runs on infrared, 60 across both bands with Table A3, no seed loop, 25 epochs per fold against the 30 of §3.3, and terciles recomputed per modality, so "small" is $\leq$652 px on infrared and $\leq$453 px on visible). Its defence is fold averaging and recording-paired tests over the 34–37 recordings of Table A2, not seed averaging, and A.7 shows where that strains — the one pair not referenced against our own model, FPN $-$ U-Net, splits its two criteria. None of those intervals or p-values is corrected for multiplicity.

The audit is one representation at one threshold. The one alternative feature space we tried is not a substitute: the target-layout variant reverses the ranking (IR 87.8%, VIS 86.4%, CST 35.1%; A.2) and saturates at 0.95 on every dataset (96.0–99.8%, Table A4). The per-pair null is empty at 0.99, which — on the assumption that its deterministic half-shift pairing behaves like a representative draw, an assumption we have not checked — puts the per-pair rate below about 0.03% (Anti-UAV), 0.5% (CST) and 1% (stills) rather than establishing zero. And no matched pair was inspected individually — neither the 2.5% and 4.8% above-null matches on the still controls (§5.3(iii); A.2) nor a sample of CST's 72.5%.

One draw, six recordings, three settings, one domain. The matched gain is conditional on the clip-random draw we shuffled and §5's rates on the clean one. The recording-level variability of the matched gain is unmeasured — the cross-validation bears on the concurrent work's ranking (§6.3), not on the leakage gain — and the night result is the narrowest of these: the +18.0-point visible night gain (Fig. A3) is measured on two held-out night recordings carrying 17 clips. What §4.4 establishes is that leakage buys most where the clean model is weak; that the visible band is weak at night is not established beyond this split. The manufactured-ranking evidence is itself one architecture pair on one modality — U-Net versus PAN on visible; the one other pair we ran across seeds, FPN versus PAN on infrared, is uninformative rather than confirming (§6.2; both are enumerated in A.7). Three settings, two of which audit the same partition, cannot establish a functional relation between contamination and captured fraction, and we fit none (§5.4); they also differ in leak regime (frame-random on CST, clip-random on Anti-UAV, §3.2) and in coverage gap, so contamination is one of at least three things that vary across the three points. §4.3's remark that CST's stronger frame-random leak works against the inversion follows from that regime difference rather than from a measurement: no dataset here is split under both regimes, so the two were never compared on one dataset. Nor did we audit a second draw of the CST sequence partition. What §5.3(ii) reports is a high operational similarity in the draw we evaluated; its prevalence across other draws of the same benchmark is unmeasured, as is the gap the two checkpoints would show on an uncontaminated CST test set. The gain is measured against whatever common test set exists, and where that set is contaminated — CST, 72.5% — the baseline A is inflated with it, so §5's audit is a prerequisite for §4 rather than a companion to it. The audit's own assumption transfers less far still: its representation and thresholds would have to be re-derived for moving cameras or large objects (§5.3(iii)). The modern architectures also run the ResNet-tuned recipe (§3.3) and are reported for the leakage delta only.

Reproducibility and unfinished checks. We redistribute no frames of either video benchmark; the releases of §7 — above all Table A1's split composition, the Anti-UAV clean and leaky splits and the five CV folds, the seeded script that regenerates the CST sequence partition, the audit lists for the two still-image controls (whose train/test split is the authors' own, §3.1), and the recording-clustered aggregates behind Table A2 and Table A3 — are the substitute for it. The split files, the five CV folds and the per-recording CV results for both bands — re-scored from the saved fold checkpoints with no retraining — are what let every clustered interval and paired test of §6 be recomputed without the frames. The audit is CPU-only and needs no model. The failing CST split is the concurrent pipeline's: the benchmark ships no segmentation split — it is a tracking set — so that pipeline drew the 154/33/33 sequence partition and prompted the masks (SAM 2, `sam2.1_hiera_large`, box prompts from the released annotations), never validating them against human labels (§8). Every CST number is therefore agreement with those pseudo-masks. Both protocols draw from the one generated pool, so no systematic labelling difference separates the two training sets; that does not equalise the annotation-error distribution across the two different test sets, nor show that both checkpoints reproduce pseudo-label error equally well, so an interaction between label quality and exposure is not ruled out — we did not test one. Validating the masks against human annotation would be needed only to generalise to human-label segmentation accuracy (A.9). What an outsider can reproduce on CST is the audit's contamination verdict against its null and adjacent references, and matched $>$ conventional once they rebuild the frame-random tree from the recipe of §3.2 — we document it but ship no generator (§7) — and train both arms per architecture; that the screen can also return a healthy verdict (adj/clean 56.3$\times$ VIS, 5.5$\times$ IR) is shown only on data we do not release, and the only public passing rows are the two still-image sets, which have no temporal structure to remove.

A.7 SPLIT COMPOSITION, PER-ARCHITECTURE, PER-SEED AND PER-CELL DETAIL (§4 AND §6)

This section holds the per-architecture read-outs, split arithmetic, seed enumerations, evaluation provenance and per-cell test results that §4 and §6 state in summary form.

The aggregate that hides the difference (§6.3). So that the comparison does not rest on two different evaluations, we recompute an unstratified aggregate from the same folds as Table A2: weight each recording's strata by frame count, average within the recording, then take the mean and 95% t-interval over the 37 recordings. The six models come out at 85.4 (FPN), 85.2 (PAN), 85.1 (U-Net), 84.9 (DeepLabV3+), 84.5 (ours) and 83.1 (PSPNet) — a 2.2-point span, each interval half-width between ± 2.5 and ± 2.9 ($t_{36} = 2.03$), all six overlapping. The +10 to +13-point small-target separation of Table A2 is therefore hidden by the aggregation rule rather than produced by a change of evaluation: it is the same 60 fold-runs read two ways. The released per-recording file reproduces both readings.

Per-architecture matched gains (§4.3). Per-dataset ranges are narrow (IR +4.9 to +7.1; VIS +7.8 to +8.9), and the three modern configurations on CST — FPN + ConvNeXt-T (+8.0), UPerNet + ConvNeXt-T (+5.9), SegFormer MiT-B2 (+6.6) — have matched gains in the same range as the ResNet-34 decoders (single seed, under the ResNet-tuned recipe of §3.3).

The CST FPN entry (§4.3). That entry is the run with the lowest clean score on CST (A = 0.580 against 0.676–0.717 for the other four) and the largest matched gain (+15.6). Excluding it, the CST means are +7.2 matched and -2.8 conventional, a captured fraction of -39%: the inversion is not carried by that run.

CST arithmetic (§4.3). Arithmetically, $\mathrm{B} - \mathrm{A} = (\mathrm{A}'' - \mathrm{A}) + (\mathrm{B} - \mathrm{A}'') = -1.4 - 0.1$ on CST.

Test-set swap for the clean-trained checkpoints (§4.1). Scored on the leaky test set instead of their own, the clean-trained checkpoints gain 3.7 (IR) and 5.6 (VIS) points and lose 1.4 on CST ($\mathrm{A}'' - \mathrm{A}$; means of five checkpoints, unrounded).

Split composition behind §4.1, §4.2 and §4.3. Of the 47 Anti-UAV leaky-test clips, 27 come from clean-training recordings, 11 from clean-validation recordings and only 9 from four of the six held-out recordings (Table A1). The clean checkpoint has therefore trained on 27 of the 47 clips themselves — every clip of a clean-train recording is in clean train — and selected its epoch on 11 more, so $\mathrm{B} - \mathrm{A}''$ compares literal-frame exposure on 57% of the leaky test set (clean checkpoint) with sibling-clip exposure on all of it (leaky checkpoint), every leaky-test recording also appearing in leaky train. In the other direction, the 50 clean-test clips divide by leaky split as 35 in leaky train, six in the leaky checkpoint's validation set and nine in leaky test (Table A1; §4.2); the 15 that are not in leaky train are sibling clips of the same flight.

Day and night (§4.4). The six clean-test recordings are four day and two night (Table A1), the night ones carrying 17 clips. The per-architecture matched gains are +3.2 to +4.7 by day and +17.4 to +19.2 at night. Clean-trained night IoU on the visible band is 0.731–0.736 for all five architectures, and 0.73–0.75 across seeds 42–44 for every multi-seed configuration. Within infrared the weaker clean cell is day (0.840, matched gain +7.3) rather than night (0.885, +5.0).

Seed spread behind §4.5. The multi-seed configurations are these: the visible U-Net/PAN comparison of §6.2 and our own model use three seeds (42–44), PAN and FPN on infrared two (42–43). Across their seeds the conventional delta moves by at most 1.3 points, that widest range being our own FCN rather than a Table 1 architecture (A.6): three seeds on the visible band (ours +5.1 to +6.4, U-Net +5.6 to +6.3, PAN +5.5 to +6.0) and two on infrared (PAN +1.8 to +2.8, FPN +2.2 to +2.4), nearly four times smaller than the smallest matched gain in Table 1 (+4.9). The clean-trained score A moves by at most 0.7 points across seeds for the four Table 1 configurations with extra seeds (PAN VIS 0.839–0.846; PAN IR 0.857–0.863). §8 and A.6 state which architectures these ranges do and do not cover, and that they are ranges over two or three runs rather than bounds.

Input cache (§4.5; A.4, *Cross-evaluation*). For the Table 1 architectures, seed 42 is quoted from the re-evaluation pass of Table 1; seeds 43–44 (and all seeds of *ours*) are training-time evaluations, which differ from that pass by at most 0.24 points on the visible band and by zero on infrared. The cause is an input-pipeline difference, which we verified directly: the visible frames are 1920×1080, so the training runs read a pre-resized cache re-encoded as JPEG at quality 98, while the re-evaluation decodes the originals. Scoring the same five seed-42 visible checkpoints both ways reproduces

the training-time number for every architecture to four decimals (U-Net 0.8439 vs 0.8420, PSPNet 0.8221 vs 0.8221, PAN 0.8426 vs 0.8440, FPN 0.8370 vs 0.8363, DeepLabV3+ 0.8409 vs 0.8385; `results/cache_effect_visible.csv`). The infrared frames are already 640×512 and were not cached, which is why they agree exactly. Every number in Table 1 comes from the uncached re-evaluation pass, so the table is internally consistent; the effect is smaller than the smallest matched gain by a factor of 20.

Modality levels (single clean split, seed 42; Table 1; §6.1). Under the leaky protocol the five ResNet-34 decoders score 0.873–0.907 on the visible band (mean 0.895) against 0.867–0.885 on infrared (mean 0.881), a per-architecture gap of +0.6 (PSPNet) to +2.2 (U-Net). Under the clean protocol they score 0.852–0.864 on infrared (mean 0.859) against 0.822–0.844 on visible (mean 0.837), a gap of −1.4 (PAN) to −3.0 (PSPNet). Our model's per-seed margins are the ones §6.1 summarises: leaky visible exceeds infrared on all three seeds (+1.4, +1.9, +1.6), and clean infrared exceeds visible on both seeds with matched clean runs (43 and 44: −1.5 and −2.7); no infrared clean seed-42 run of ours is available. Under the clean protocol our model's best visible seed (0.845) lies below its worst infrared seed (0.859); under the leaky protocol its worst visible seed (0.895) lies above its best infrared seed (0.881).

Modality tie arithmetic (§6.1). The modality whose clean model is weaker on this test set gains more: the matched gains (+8.5 visible against +6.4 infrared) cancel the 2.1-point clean gap (2.2 from the rounded entries of Table 1) almost exactly, which is the tie §6.1 reports. On the common clean test frames the two leaky-trained models score 0.9225 (infrared) against 0.9220 (visible), $B' = 0.922$ for both in Table 1.

U-Net versus PAN across seeds (visible, seeds 42–44; §6.2). The conventional gains are +6.3, +6.1, +5.6 (U-Net) against +5.5, +5.6, +6.0 (PAN), the per-seed differential ranging from −0.4 to +0.8. The leaky margins are +0.9, +0.4 and +0.3 points, and the leaky seed ranges are 0.9022–0.9070 (U-Net) and 0.8976–0.9018 (PAN), which do not overlap; the clean seed means are 0.845 (U-Net) and 0.843 (PAN) and the clean seed ranges 0.8439–0.8460 and 0.8392–0.8458, which do. Under the clean protocol two seeds favour U-Net and one favours PAN, and U-Net's clean seed-to-seed spread is 0.2 re-evaluated and 0.4 training-time against PAN's 0.7. The one other pair we ran across seeds is FPN − PAN on infrared, seeds 42 and 43 under both protocols: clean +0.7, −0.3; leaky +0.0, +0.2.

Re-evaluation offsets in §6. Seed-42 visible clean scores quoted in §6.1 and §6.2 come from the re-evaluation pass of Table 1, so they carry the cache offset set out in *Input cache* above: the training-time day mean is 0.884 rather than 0.885 and the PAN clean gap −1.3 rather than −1.4; the seed-42 U-Net − PAN clean margin is −0.2 training-time and +0.1 re-evaluated.

Infrared terciles (Table A2). Of the 37 recordings in the CV pool, 34, 37 and 35 have frames in the small (mask area ≤652 px), medium (652–988 px) and large (>988 px) terciles; a recording enters a tercile only if it has frames there.

Infrared medium and large cells (Table A7, Table A2). DeepLabV3+ − ours on medium is +0.5 [+0.1, +0.9] but sign p = 0.32; FPN − ours on medium has sign p = 0.003 but CI [−0.0, +0.8]. The cluster-mean gaps are +0.8 (medium) and +1.2 (large). No medium or large pair is significant under both criteria.

Infrared night·small cell (Table A2). In the night-small cell (n_rec = 13) FPN reaches 0.773 while ours drops to 0.545, a cluster gap of +18.2 points against +5.9 on the day recordings; the plain models score 8.1 to 11.2 points lower on the night recordings than on the day recordings for small targets, the pyramid models do not (+0.2 to +5.2). At n = 13 the paired CIs are wide — only PAN − ours, +17.6 [+0.7, +34.6], excludes zero — so we read this as the direction of the effect rather than a significant result.

Visible-band paired cells (Table A3). A.1 reads this table's levels; the paired statistics behind them are these. On the small tercile (≤453 px, n_rec = 31) FPN − ours is +4.1 [+1.3, +7.0], positive on 87% of recordings, $p < 0.001$, against a cluster-mean gap of +3.6 points, while FPN − ours is +0.1 on medium and +0.0 on large (Table A7). The small-target gap lies on the day recordings (FPN − ours +6.4 [+1.9, +11.0], 94%, $p < 0.001$, n = 18) rather than the night ones (+1.0 [−1.4, +3.4], n = 13).

Table A1: **Composition of the two Anti-UAV protocols** (from the released split files; 37 recordings, 318 clips; the six clean-test recordings are four day and two night flights).

	recordings	clips	night clips
clean train / val / test	26 / 5 / 6	225 / 43 / 50	84 / 16 / 17
leaky train / val / test	37 / 26 / 25	223 / 48 / 47	82 / 18 / 17
the 50 clean-test clips, by leaky train / val / test		35 / 6 / 9	
the 47 leaky-test clips, by clean train / val / test		27 / 11 / 9	

A.8 THE CHECKLIST OF §7, IN FULL

§7 states the protocol in summary. Each item is given here with the qualification that governs it.

1. **Partition by recording.** Assign whole recordings — the coarsest unit believed to share camera, site, background and target identity — to train, validation and test, stratified on any covariate you report on (§3.2). Recording-disjointness is necessary but not sufficient: CST is recording-disjoint and fails (§5.3(ii)), which is what item 2 tests.
2. **Audit the split before trusting it.** Run §3.4's audit against its two references. Set the threshold at the lowest value where your own null goes empty rather than inheriting our 0.99, and report the adjacent-to-clean ratio rather than a pass/fail bit: passes sit several times below the adjacent reference and failures within a small factor of it — ours were 56.3×, 5.5× and 0.97× (VIS, IR, CST), with nothing in between and no evidence for a cutoff (§5.1–5.3). Do not substitute the target-layout variant (A.2). A pass is a verdict at this measure's resolution, not a certificate against memorisation.
3. **If the audit fails, report the number and reinterpret; do not discard the benchmark.** Confirm on the like-for-like clean-over-leaky statistic rather than the single-pair adjacent reference: a split that removed something scores a small fraction of the leaky split's rate, the two rates each measured on their own test set (323 clean against 411 leaky frames on CST, 4,734 against 4,223 on infrared) — 13% (VIS) and 43% (IR) against CST's 83% (§5.2). State only what that supports: near-identical content under different sequence identifiers, provenance untraced. Then treat the clean baseline as inflated, which leaves the matched gain a fixed-test protocol contrast whose CST value is left open below, and the conventional delta uninterpretable either way (§4.1).
4. **Report protocol-matched numbers for any leakage claim.** Match both protocols' training budgets, score both checkpoints on both test sets, and report $B' - A$ on the common clean test set (§3.2, §4.2). $B - A$ alone is not a leakage estimate and no fixed offset repairs it (§4.1). Read $B' - A$ as a fixed-test protocol contrast and not a bound on memorisation, since coverage and checkpoint selection also differ and we measure neither (§8), and set it against whatever seed-to-seed variability you have actually measured: a gain below that variability is not interpretable, but exceeding it is not a significance test either. With one seed the variability is unmeasured, so report the gain and say so rather than make a significance claim.
5. **Report stratified and clustered, and state the unit of sampling.** Give size-stratified, recording-clustered means (§3.4) and test architectures with a recording-paired mean difference plus a two-sided sign test, calling a difference firm only where both agree and otherwise saying which criterion carried it (FPN − U-Net on small: interval only, Table A7). Audit rates are frame-level but recording-clustered, so read them as order-of-magnitude quantities (§5.2). The conjunction is conservative rather than error-rate-controlled, and it is uncorrected for multiplicity.

One release exception belongs with item 3: the CST frame-random split is a symlink tree we document but ship no generator for.

A.9 COMPOSITION OF THE TWO ANTI-UAV PROTOCOLS

CST Anti-UAV (public; infrared). 220 thermal sequences, each a single continuous recording (Xie et al., 2025). It is a tracking set with no segmentation split, so the §3.2 partition and the masks (SAM 2 prompted with its per-frame boxes) come from the concurrent pipeline, and every CST score in this

paper is agreement with those pseudo-masks rather than with human labels. Both protocols draw their labels from the one generated pool, so no systematic labelling difference separates the two training sets; that does not make the annotation-error distribution the same on the two different test sets, nor does it establish that the two checkpoints reproduce pseudo-label error equally well. We did not test the interaction between label quality and exposure separately, and validation against human annotation would be needed only to generalise these numbers to human-label segmentation accuracy.

IRSTD-1k (Zhang et al., 2022) and NUDT-SIRST (Li et al., 2023) (public; still images). 1,001 real and 1,327 synthetic infrared frames with masks, under the authors' train/test lists. Having no temporal structure, they are the audit's controls.

A.10 Frame-level test of the mechanism

§8 reports that this analysis finds no detectable association between a frame's near-copy similarity and its leakage gain. The measurement: for every frame of the six held-out infrared recordings we take the maximum similarity to the *clean* training pool — the frames the clean-trained checkpoint saw, so the predictor is that checkpoint's exposure and not the leaky checkpoint's extra exposure, which we never measured directly — in the representation of §3.4, and the per-frame gain is the leaky-trained minus the clean-trained IoU on that same frame, averaged over the five ResNet-34 architectures. No training is involved; the checkpoints are those of Table 1. The per-frame values are released (`results/frame_gain_infrared.csv`, 4,733 frames with a target).

Treating frames as independent gives Spearman +0.09 over 4,733 frames, but frames within a recording are not independent (§2). Computed within each recording and aggregated over the six, the correlation is +0.11 with a 95% interval of $\pm$0.37 and a positive sign in three of six. Restricting to the small-target tercile, where leakage helps most, gives +0.17 $\pm$ 0.36, positive in four of six.

Binned by similarity, the gain does not increase with it: +9.2, +6.2, +7.3 and +13.5 points from the lowest to the highest quartile, with recording-clustered intervals of $\pm$6 to $\pm$24 points. Split at the 0.99 threshold that defines a near-copy, the 325 frames that have one gain +3.5 points and the 4,408 that do not gain +8.9. Per recording, the near-copy rate and the gain move independently: the recording with the largest gain (+21.7 points) has no near-copies, while the two with the highest near-copy rates (15.8% and 22.1%) gain +4.3 and +5.3.

Six recordings support none of these differences, so we draw no quantitative conclusion from them. The statistic establishes no clear relation in either direction: at +0.11 with a 95% interval of $\pm$0.37 it neither identifies a frame-by-frame memorisation mechanism nor excludes one, and with six recordings and a predictor that is not exposure itself it has little power to do either. A null of this kind is at least consistent with what recording-level leakage means — a model that has trained on other frames of a recording gains on all of that recording, whether or not any particular frame has a near-copy — and with the audit measuring background and camera view rather than what a model memorises (§8).

A.11 Seed stability of the matched gain

Table 1 is seed 42 throughout. Four configurations were trained at further seeds under both protocols, so their cross-evaluation can be repeated without any new training. All ten runs are given below; $B - A$ is the conventional comparison, $B' - A$ the matched gain, and the last column their ratio.

configuration	seed	A	$B - A$	$B' - A$	captured
infrared FPN	42	0.8635	+2.19	+6.89	32%
infrared FPN	43	0.8605	+2.35	+7.26	32%
infrared PAN	42	0.8570	+2.84	+6.84	42%
infrared PAN	43	0.8632	+1.84	+5.59	33%
visible PAN	42	0.8426	+5.50	+8.91	62%
visible PAN	43	0.8458	+5.60	+8.31	67%
visible PAN	44	0.8392	+5.98	+9.34	64%
visible U-Net	42	0.8439	+6.31	+8.36	75%
visible U-Net	43	0.8447	+6.07	+8.01	76%
visible U-Net	44	0.8460	+5.62	+6.80	83%

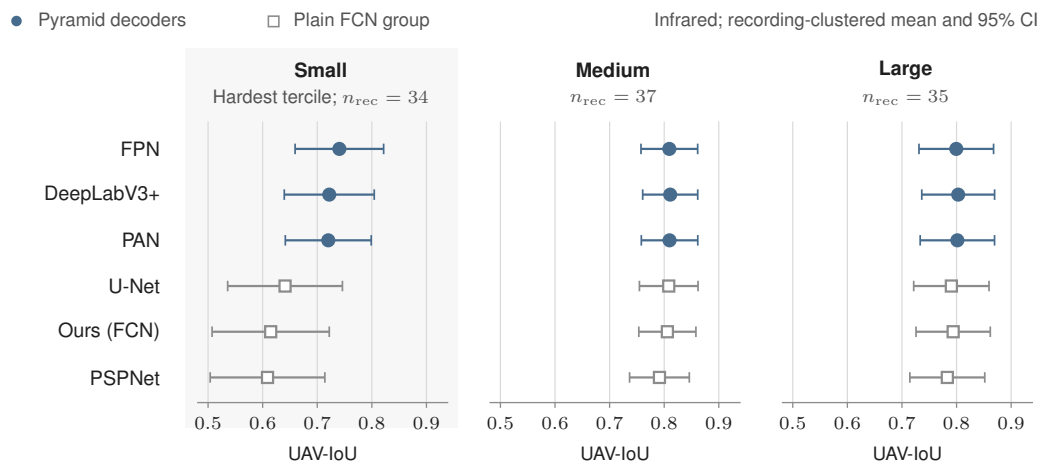


Figure A1: *Size-stratified infrared re-evaluation, reproduced from the concurrent benchmark (§6.3).* Recording-clustered mean UAV-IoU with 95% intervals, 5-fold recording-level cross-validation over 37 recordings (n = 34, 37, 35 by tercile); the pyramid/plain labels are an empirical grouping. The clusters differ by +10 to +13 IoU on the smallest tercile; elsewhere the tested pairs differ by under a point. The paired tests behind that claim are in Table A7 and concern paired differences, not the marginal intervals shown here.

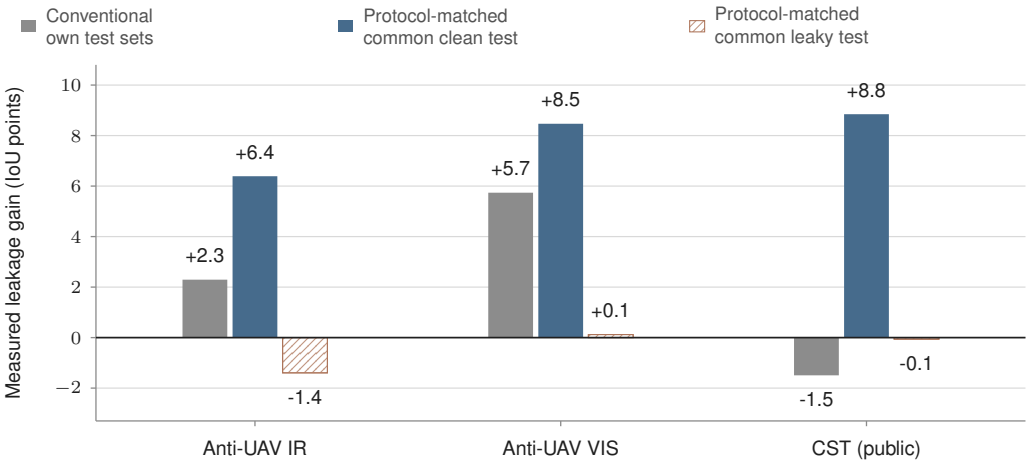


Figure A2: *The conventional comparison understates or hides leakage.* Architecture-mean leakage gain in UAV-IoU points, five architectures per dataset. Grey: conventional, each protocol on its own test set. Blue: protocol-matched on the common clean test set, the fixed-test protocol contrast. Orange hatched: protocol-matched on the common leaky test set, where both models score alike because that set is already contaminated.

Every run gives a large positive matched gain, the smallest being +5.6. Within a configuration the gain moves by 0.4 (infrared FPN), 1.3 (infrared PAN), 1.0 (visible PAN) and 1.6 points (visible U-Net) — at most a third of the smallest matched gain in Table 1. The captured fraction separates the two bands cleanly: 32–42% on infrared against 62–83% on visible, with no overlap, which is the seed-level form of the ordering §5.4 reads off the per-dataset means.

What this does not cover: eleven of the fifteen Table 1 entries, every CST entry, and every A'' cell, all of which remain single-seed. The runs also share their splits, so they bound seed variation and not partition variation (§8).

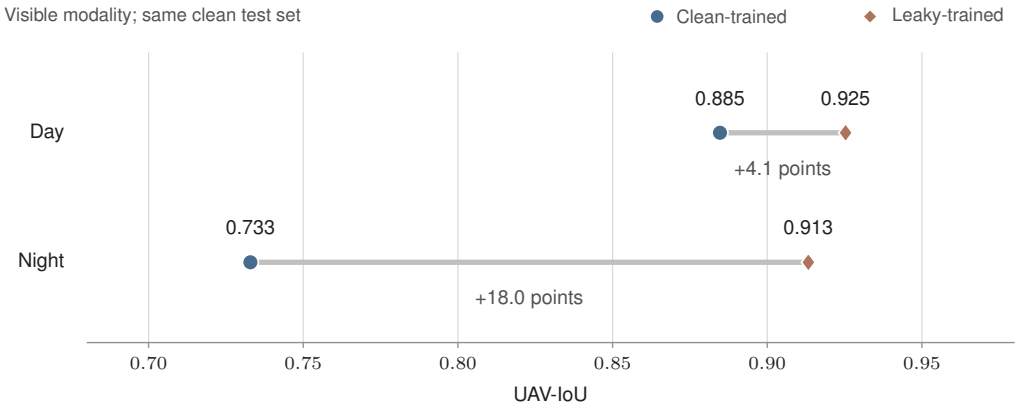


Figure A3: *Leakage masks the visible-modality night-time weakness.* Visible-band UAV-IoU on the same clean test frames for models trained on the clean (circles) and leaky (diamonds) splits, mean over five architectures. By day the gain is +4.1 points, at night +18.0 — a stratum-level weakness erased on these recordings, not a modality-level one (§8).

Table A2: **Size-stratified re-evaluation under recording-level 5-fold cross-validation (infrared).** Reproduced from concurrent work (Anonymous, 2026) for reference. Recording-clustered mean UAV-IoU ± 95% CI over all held-out recordings (n_{rec} per tercile in the header; the night·small column is over the 13 night recordings); terciles of target area. Architectures separate only on the smallest tercile; the recording-paired tests behind that claim are in Table A7.

Architecture	Family	small (n=34)	med (n=37)	large (n=35)	night·small
FPN	pyramid	0.741 ± 0.083	0.809 ± 0.053	0.799 ± 0.069	0.773 ± 0.151
DeepLabV3+	pyramid	0.722 ± 0.084	0.811 ± 0.051	0.803 ± 0.068	0.739 ± 0.161
PAN	pyramid	0.720 ± 0.080	0.810 ± 0.052	0.801 ± 0.069	0.722 ± 0.150
U-Net	plain FCN	0.641 ± 0.107	0.808 ± 0.054	0.790 ± 0.070	0.591 ± 0.220
ResNet34-FCN (ours)	plain FCN	0.615 ± 0.109	0.806 ± 0.053	0.794 ± 0.069	0.545 ± 0.224
PSPNet	plain FCN	0.609 ± 0.107	0.791 ± 0.055	0.783 ± 0.070	0.550 ± 0.214

Table A3: **Visible-band size-stratified leaderboard under recording-level 5-fold cross-validation.** Reproduced from concurrent work (Anonymous, 2026) for reference. Companion to Table A2. Recording-clustered mean UAV-IoU ± 95% CI; terciles of target area (small ≤453 px, medium ≤731 px). The small-target separation reproduces on the visible band but is smaller, and it sits on the day recordings rather than the night ones.

Architecture	Family	small (n=31)	med (n=34)	large (n=32)	day·small (n=18)	night·small (n=13)
DeepLabV3+	pyramid	0.711 ± 0.083	0.840 ± 0.039	0.791 ± 0.086	0.692 ± 0.131	0.738 ± 0.102
FPN	pyramid	0.708 ± 0.076	0.843 ± 0.039	0.793 ± 0.086	0.685 ± 0.116	0.740 ± 0.104
PAN	pyramid	0.687 ± 0.087	0.838 ± 0.040	0.788 ± 0.086	0.656 ± 0.136	0.731 ± 0.107
PSPNet	plain FCN	0.668 ± 0.083	0.818 ± 0.044	0.762 ± 0.088	0.642 ± 0.129	0.704 ± 0.101
ResNet34-FCN (ours)	plain FCN	0.666 ± 0.091	0.842 ± 0.038	0.793 ± 0.084	0.621 ± 0.144	0.730 ± 0.098
U-Net	plain FCN	0.664 ± 0.094	0.842 ± 0.046	0.792 ± 0.090	0.621 ± 0.146	0.723 ± 0.111

Table A4: **Near-duplicate audit, full detail.** Companion to Table 2. Similarity is the Pearson correlation of 80×64 mean-centred, L2-normalised grayscale frames. The *clean* and *leaky* rows are maxima over the whole training set; the *random* and *adjacent* rows are single-pair statistics and are therefore not directly comparable in level. The last row of each block is the target-layout variant (nearest neighbour in normalised centroid and area): it saturates on healthy splits and inverts the ranking, and is reported only to show that it does not discriminate (A.2).

Dataset	Measure	n_{test}	n_{train}	median	p90	>0.95 (%)	>0.99 (%)
Anti-UAV IR	random pairs (null)	10323	–	0.3627	0.8082	0.4	0.0
	adjacent frames	4000	–	0.9807	0.9993	68.3	37.9
	clean split	4734	20646	0.9527	0.9864	52.1	6.9
	leaky split	4223	20813	0.9382	0.9930	45.9	16.0
	clean split, target layout	4733	20646	0.9956	0.9980	99.5	87.8
Anti-UAV VIS	random pairs (null)	10073	–	0.2410	0.6291	0.0	0.0
	adjacent frames	4000	–	0.9584	0.9978	55.0	25.9
	clean split	4375	20147	0.9389	0.9833	41.7	0.5
	leaky split	4055	19870	0.9259	0.9830	37.1	3.6
	clean split, target layout	4374	20107	0.9953	0.9978	99.8	86.4
CST (public)	random pairs (null)	600	–	0.0541	0.5983	0.0	0.0
	adjacent frames	1386	–	0.9968	0.9996	80.1	70.6
	clean split	323	1540	0.9953	0.9992	80.8	72.5
	leaky split	411	1545	0.9989	0.9998	90.0	86.9
	clean split, target layout	322	1527	0.9880	0.9946	99.1	35.1
IRSTD-1k	random pairs (null)	400	–	0.3392	0.7815	0.2	0.0
	adjacent frames	799	–	0.3411	0.7629	0.0	0.0
	clean split	201	800	0.8772	0.9617	14.4	2.5
	clean split, target layout	201	800	0.9836	0.9928	96.0	19.9
NUDT-SIRST	random pairs (null)	331	–	0.0470	0.5080	0.0	0.0
	adjacent frames	662	–	0.1005	0.6570	0.3	0.0
	clean split	664	663	0.7642	0.9649	12.7	4.8
	clean split, target layout	664	663	0.9790	0.9926	96.5	14.8

Table A5: **Cost of the architectures in the cross-validated leaderboard.** Parameters, GFLOPs and throughput at 512×640, ResNet-34 encoder. Cost does not explain the small-target ordering (Table A2): the winning decoders are the more expensive ones in FLOPs, U-Net excepted. FPS was measured once, with device, batch size, precision, warm-up and iteration count unrecorded, so only its ordering is read; the FLOP counter and whether its figures are multiply-accumulates or floating-point operations were not recorded either, so the GFLOPs column is comparable within itself only (A.3).

Architecture	Family	Params (M)	GFLOPs	FPS
PSPNet	plain FCN	21.44	23.6	378.3
ResNet34-FCN (ours)	plain FCN	21.46	48.3	229.6
DeepLabV3+	pyramid	22.44	79.3	189.6
FPN	pyramid	23.16	68.8	177.5
PAN	pyramid	21.48	74.7	165.2
U-Net	plain FCN	24.44	78.6	141.0

Table A6: **Protocol-matched evaluation, every architecture.** Expansion of Table 1. A/B: clean-/leaky-trained checkpoint scored on its own test set (the conventional comparison, $B - A$). B': the leaky-trained checkpoint re-scored on the *clean* test set; $B' - A$ is the fixed-test protocol contrast. Last column: the same on the common leaky test set. UAV-IoU, seed 42. *Captured* $= (B - A)/(B' - A)$.

		Own test set		Conv.	Common clean test		Common leaky test	
Dataset	Architecture	A	B	$B - A$	B'	$B' - A$	A''	$B - A''$
Anti-UAV IR	U-Net	0.862	0.885	+2.3	0.924	**+6.2**	0.902	-1.7
	PSPNet	0.852	0.867	+1.5	0.902	**+4.9**	0.872	-0.6
	PAN	0.857	0.885	+2.8	0.925	**+6.8**	0.907	-2.2
	FPN	0.864	0.885	+2.2	0.932	**+6.9**	0.895	-1.0
	DeepLabV3+	0.858	0.885	+2.7	0.929	**+7.1**	0.900	-1.5
	mean (n=5); captured 36%	0.859	0.881	**+2.3**	0.922	**+6.4**	0.895	-1.4
Anti-UAV VIS	U-Net	0.844	0.907	+6.3	0.927	**+8.4**	0.905	+0.2
	PSPNet	0.822	0.873	+5.1	0.910	**+8.8**	0.871	+0.2
	PAN	0.843	0.898	+5.5	0.932	**+8.9**	0.897	+0.1
	FPN	0.837	0.898	+6.1	0.922	**+8.5**	0.897	+0.0
	DeepLabV3+	0.841	0.898	+5.7	0.918	**+7.8**	0.897	+0.1
	mean (n=5); captured 68%	0.837	0.895	**+5.7**	0.922	**+8.5**	0.893	+0.1
CST (public)	U-Net	0.678	0.641	-3.7	0.760	**+8.2**	0.652	-1.1
	FPN	0.580	0.618	+3.7	0.737	**+15.6**	0.618	-0.1
	FPN + ConvNeXt-T	0.695	0.681	-1.4	0.775	**+8.0**	0.678	+0.3
	UPerNet + ConvNeXt-T	0.717	0.686	-3.0	0.775	**+5.9**	0.685	+0.2
	SegFormer MiT-B2	0.676	0.644	-3.1	0.741	**+6.6**	0.640	+0.4
	mean (n=5); captured -17%	0.669	0.654	**-1.5**	0.758	**+8.8**	0.655	-0.1

Table A7: **Recording-paired architecture comparisons (infrared unless marked VIS).** Reproduced from concurrent work (Anonymous, 2026) for reference. Companion to Table A2. Mean paired difference $b - a$ in IoU points over the recordings that have frames in the stratum, its 95% t-interval, the fraction of recordings with $b > a$, whether the interval excludes zero, and the two-sided sign-test p. The two criteria disagree when a few recordings carry the mean; we call a difference firm only where both agree. No multiplicity correction is applied.

Pair ($b - a$)	Cell	mean diff	95% CI	% rec. $b > a$	CI$\not\ni 0$	sign p
FPN − ResNet34-FCN (ours)	small	+12.6	[+2.5, +22.7]	85	**yes**	<0.001
	med	+0.4	[-0.0, +0.8]	76	no	0.003
	large	+0.6	[-0.4, +1.5]	71	no	0.009
PAN − ResNet34-FCN (ours)	small	+10.6	[+2.4, +18.7]	85	**yes**	<0.001
	med	+0.4	[-0.0, +0.8]	68	no	0.047
	large	+0.8	[-0.1, +1.6]	66	no	0.058
DeepLabV3+ − ResNet34-FCN (ours)	small	+10.7	[+1.8, +19.7]	85	**yes**	<0.001
	med	+0.5	[+0.1, +0.9]	59	**yes**	0.324
	large	+0.9	[-0.1, +1.9]	69	no	0.024
FPN − U-Net	small	+10.0	[+0.7, +19.2]	65	**yes**	0.121
	med	+0.1	[-0.4, +0.7]	41	no	0.324
	large	+0.9	[-0.7, +2.5]	40	no	0.392
FPN − ResNet34-FCN (ours) (VIS)	small	+4.1	[+1.3, +7.0]	87	**yes**	<0.001
	med	+0.1	[-0.4, +0.7]	68	no	0.058
	large	+0.0	[-0.6, +0.7]	75	no	0.007