ROTATION SENSITIVITY IN ORIENTED DETECTORS IS NOT ONE QUANTITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Oriented object detectors can be accurate without being equivariant to rotations of their input, and this lack of equivariance is usually summarised by a single number. We split the AP cost of the four axis-aligned views into a quarter-turn mode C, a half-turn mode H and a signed chirality Q, three fixed combinations of the discrete Fourier coefficients of per-view AP, and we measure the disagreement between predictions on different views separately from its AP cost. Both measurements need an exact map back to the original frame. Mapping predictions back with the pixel-index form of the inverse rotation is off by one pixel for continuous box coordinates, displaces the half-turn view along both axes and inflates H, so we check the mapping against the array rotation it inverts. In a 2×2 augmentation factorial on DOTA with paired seeds, each mode is removed by the augmentation that covers the rotations it measures. Adding the vertical flip, whose composition with the horizontal flip is the half turn, removes H (macro AP50 $0.0218 \rightarrow 0.0005$) but leaves most of C ($0.0505 \rightarrow 0.0418$), whereas adding rotation removes both ($C = -0.0049$, $H = -0.0015$). Bootstrap intervals over seeds and scenes separate H from zero only in cells trained without the half turn, and the pattern holds at a second model capacity, a second input size and a stricter overlap threshold. A control with rotations of at most 15° improves accuracy but removes neither mode. After rotation training no rotated view has a detectable AP cost, although, averaged over classes, more than a tenth of the predictions have no counterpart when the canonical and a rotated view are matched in either direction. A detector with a rotation-equivariant backbone and the same flips keeps about a third of the C that flips alone leave in our detectors (0.0132 against 0.0418). Because a half turn leaves box orientations unchanged, an explanation based on box orientation can only account for C. On DOTA, classes that are more anisotropic under a quarter turn lose more C under rotation training (Spearman $\rho = -0.62$, permutation $p = 0.018$), and H shows no such relation. This class-level relation does not hold on SODA-A, where C is indistinguishable from zero. Some small residuals remain, including a half-turn cost in the equivariant detector and a negative chirality when rotation is added without the vertical flip, and we report the tests we ran on them.

1 INTRODUCTION

A detector trained on aerial imagery can be accurate and still fail to commute with rotations of its input. If the image is rotated by a right angle and the predictions are rotated back, they differ from the predictions made on the original image. This residual is usually reported as a single number, such as the accuracy gained by fusing rotated views or the accuracy drop averaged over them. One number is a poor summary, because averaging over rotations mixes failures that respond to different parts of training.

We consider only the four axis-aligned views, so rotation sensitivity here means sensitivity to multiples of 90°, and the geometry of these views suggests a split. A quarter turn shifts the orientation of every object's dominant edge by $\pi/2$, while a half turn leaves the orientation of every box unchanged. We therefore track a quarter-turn mode C, a half-turn mode H and the signed difference Q between the two quarter turns. These are fixed combinations of the discrete Fourier coefficients

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.

of AP over the four views (Section 2), and this elementary relation makes the geometric argument exact, since an effect that depends only on box orientation is unchanged by a half turn and can therefore contribute to C but not to H or Q.

All three measure the AP cost of a rotation, which differs from the distance to equivariance because two views can reach the same AP with different predictions. We therefore also measure how much a model's predictions on the canonical and rotated views disagree, and report the two levels separately. They behave differently. After rotation training no rotated view has a detectable AP cost, yet, averaged over classes, more than a tenth of the predictions have no counterpart when the canonical and a rotated view are matched in either direction. Both levels require the predictions on each rotated view to be mapped back to the original frame exactly. If the inverse rotation is written in its pixel-index form, the mapping is one pixel off for continuous box coordinates and inflates H in every cell of our factorial (Appendix K), so we check it against the array rotation it inverts.

Released checkpoints cannot separate these effects. The library defaults describe horizontal flips only, but the four released YOLO-OBB checkpoints for DOTA that we inspected record vertical flips and rotation augmentation in their training arguments, and every rotated view of `yolo11n-obb` is within 0.004 AP50 of its canonical view at both input sizes we evaluate. A checkpoint trained with all transformations at once shows that the modes can be removed, but not which transformation removes which.

We therefore intervene on training with a 2×2 augmentation factorial. Horizontal flipping is always on, the two factors are vertical flipping and rotation, and each cell has three paired seeds, with every other setting, including the byte-identical initialisation, fixed within a seed. Because the two flips are drawn independently, their composition, the half turn, is part of the vertical-flip cell, and only rotation adds the quarter turns. Figure 1 shows that the modes follow this split, which is what coverage of the corresponding rotations predicts. The design tests that prediction mode by mode, with intervals, and identifies the residuals that do not follow it. Adding the vertical flip removes the half-turn mode ($0.0218 \rightarrow 0.0005$ macro AP50) but leaves most of the quarter-turn mode ($0.0505 \rightarrow 0.0418$); adding rotation removes both. Intervals that resample seeds and scenes together separate H from zero only in the horizontal-flip cell, and the pattern holds at a second model capacity, a second input size and a stricter overlap threshold. A control cell with rotations of at most 15° improves accuracy but removes neither mode, and a detector with a rotation-equivariant backbone, trained with the same flips as the vertical-flip cell, keeps about a third of that cell's C.

The Fourier relation also constrains what a class-level test of the orientation prior can find. On DOTA, classes whose orientation distribution is more anisotropic under a quarter turn lose more C when rotation is added (Spearman $\rho = -0.62$, permutation $p = 0.018$), and H shows no such relation. On SODA-A, C is indistinguishable from zero before any rotation and the relation does not hold (Appendix I), so we treat it as a correlation specific to DOTA.

Some effects remain after these interventions. The equivariant detector keeps a small half-turn cost whose interval excludes zero, some rotation-trained models keep positive half-turn costs whose intervals include zero, and the chirality is negative at both capacities when rotation is added without the vertical flip. A sub-pixel localisation bias tied to the frame shown to the detector explains part of that chirality but none of the half-turn costs in the models we tested. We report these residuals with the tests we ran and do not assign them a cause.

Contributions:

- A decomposition of the AP cost of a rotation into C, H and Q, kept separate from the prediction-level disagreement whose AP cost it measures. The modes are fixed combinations of the discrete Fourier coefficients of per-view AP, an elementary identity that organises the measurements and restricts explanations based on box orientation to C.
- A check on how rotated-view predictions are mapped back, and a demonstration that the pixel-index form of the inverse rotation, one pixel off for boxes, inflates the half-turn mode in every cell (Appendix K).
- A 2×2 augmentation factorial with paired seeds and seed-and-scene intervals, showing that the half turn and rotation remove different modes, replicated at a second capacity, input size and overlap threshold, with a small-angle control.

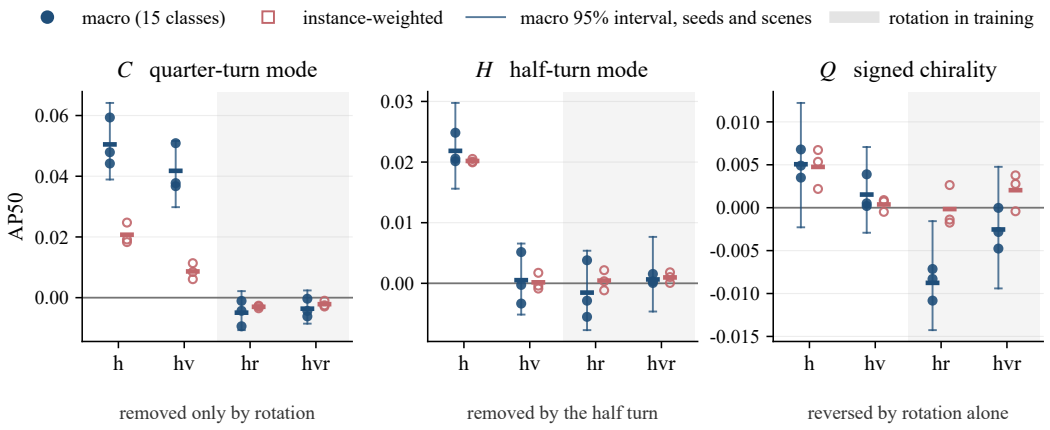

Figure 1: **The two modes are removed by different transformations.** A 2×2 augmentation factorial on DOTA with `yolo11n-obb`: three paired seeds per cell (points), cell means (bars) and, under macro weighting, 95% intervals that resample seeds and scenes together (whiskers). Cells are named for the augmentations they use: H horizontal flip, HV adds the vertical flip, HR adds rotation and HVR adds both, so horizontal flipping is present in all four. HV contains the half turn exactly, as the composition of its two flips, and HR and HVR sample all rotations. The shaded half of each panel marks the cells trained with rotation. $C = \frac{1}{2}(L_{90} + L_{270})$ and $H = L_{180}$ are the AP50 lost relative to the canonical view, and $Q = L_{90} - L_{270}$ is the signed difference between the two quarter turns. *Left:* the vertical flip leaves most of C ($0.0505 \to 0.0418$ macro), while rotation removes it. *Middle:* H disappears in every cell that contains the half turn ($0.0218 \to 0.0005$ in HV), and only the interval for H excludes zero. *Right:* Q is negative when rotation is added without the vertical flip. Canonical-view AP50 is 0.697–0.709 in all twelve models, so the differences are not explained by differences in accuracy.

- Evidence that AP cost and equivariance diverge: after rotation training no rotated view has a detectable AP cost, yet more than a tenth of the predictions, matched in either direction and averaged over classes, have no counterpart, and a detector with an equivariant backbone leaves fewer unmatched than our rotation-trained models.
- A class-level test on two datasets that ties the removal of C to orientation anisotropy on DOTA only, and an audit showing that four released DOTA checkpoints were trained with every transformation the factorial separates.
- A report of the residuals left after these interventions and of the tests run on them.

2 Measuring the residual

For a detector f and a view r_θ from the four axis-aligned rotations, we export predictions on the rotated image, map them back to the original frame, and score each view alone. The map back must be the exact inverse of the image rotation in continuous pixel coordinates, in which pixel i covers $[i, i+1)$. The inverse written for pixel indices, with $w - 1 - y$ where $w - y$ belongs, is one pixel off for boxes. It displaces each quarter-turn view along one axis and the half-turn view along both, and leaves the canonical view unchanged. We check the mapping on synthetic boxes against the array rotation it inverts, and Appendix K shows what the index form does to the modes. Writing AP_θ for the AP50 of view θ and $L_\theta = \mathrm{AP}_0 - \mathrm{AP}_\theta$,

$$H = L_{180}, \qquad C = \tfrac{1}{2}\left(L_{90} + L_{270}\right), \qquad Q = L_{90} - L_{270}. \tag{1}$$

H and C are the two modes; Q is a signed chirality, positive when the 90° view is the worse of the two quarter turns. Every quantity is computed under two class weightings, an unweighted mean over the 15 DOTA classes (*macro*) and a mean weighted by validation instance count (*instance-weighted*), and we report both because they can disagree. The definitions were fixed before any of the experimental cells below were run.

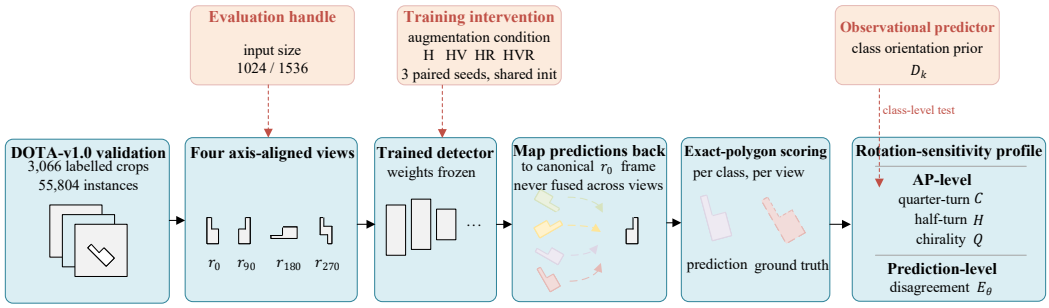

Figure 2: **The measurement pipeline.** Each of the four axis-aligned views is scored separately against the same ground truth and never fused, so we report the residual rather than the accuracy that fusing views would recover. The view glyphs are drawn asymmetric on purpose. An oriented box is unchanged by a half turn, so a deficit H on that view cannot come from the label. The three boxes above the pipeline enter at different points. Training augmentation is an intervention on the detector (Section 4), input size is a choice made at evaluation (Section 3), and the class orientation prior is an observed property tested against C and H (Section 5). The profile on the right keeps AP-level and prediction-level quantities separate.

The three quantities are fixed combinations of the discrete Fourier transform of AP over the four views, an elementary identity that organises the measurements. With $a_k = \mathrm{AP}_{k\pi/2}$ and $\hat{a}_m = \frac{1}{4}\sum_{k=0}^{3} a_k e^{-i\pi mk/2}$,

$$H = 4\,\mathrm{Re}\,\hat{a}_1, \qquad Q = 4\,\mathrm{Im}\,\hat{a}_1, \qquad C = 2\,\hat{a}_2 + \tfrac{1}{2}H, \tag{2}$$

where $\hat{a}_2$ is real. Together with AP_0, the triple (C, H, Q) determines all four per-view values, so the decomposition loses nothing. The relation also restricts possible explanations, because a half turn maps an oriented box to a box with the same orientation modulo π and the two quarter turns map it to boxes that share an orientation with each other. An effect on AP that depends on the boxes only through their orientation modulo π therefore gives $a_0 = a_2$ and $a_1 = a_3$, which sets $\hat{a}_1 = 0$ and hence $H = Q = 0$, so such an effect can produce only C.

C, H and Q describe the AP cost of a rotation, which is not the same as equivariance. Equivariance is a property of the predictions, namely whether $f(r_\theta x)$, mapped back to the original frame, equals $f(x)$, and two views can reach the same AP with different predictions. Section 6 reports models with no detectable AP cost on any rotated view, although, averaged over classes, more than a tenth of their predictions have no counterpart when the canonical and a rotated view are matched in either direction. We therefore also measure agreement at the prediction level by matching a model's r_0 and r_θ outputs to each other rather than to the ground truth. We record the fraction of canonical-view predictions without a match on the rotated view, the overlap of matched pairs, the change in predicted orientation and the change in confidence. We call this profile E_θ and keep its four parts separate, as we keep L_{90}, L_{180} and L_{270} separate, because they move independently. A composite such as $H - C$ is not stable under a change of class weighting; Appendix E reports it and shows the instability.

All statistics come from an exact-polygon evaluator. Training-time validation uses a Gaussian approximation to rotated IoU and reports AP50 values 0.029–0.050 higher; Appendix A explains the difference.

3 Auditing the released checkpoints

The released `yolo11{n,s,m,l}-obb` checkpoints store their own training arguments (we did not inspect `yolo11x-obb`). All four were trained with `fliplr=` 0.5, `flipud=` 0.5 and `degrees=` 180, for 200 epochs at batch 32, with `imgsz=` 1024 and `scale=` 0.9, and all four name their dataset as `DOTAv1-ms.yaml`, which denotes multi-scale crops. These values are read directly from the checkpoints. Despite the library defaults, the checkpoints are therefore not horizontal-flip-only models, and an explanation of their per-view behaviour cannot rely on the absence of vertical flips and rotation.

Table 1: The two modes and the chirality across the augmentation factorial at input size 1024: mean ± s.d. over three paired seeds. Canonical-view AP50 is 0.697–0.709 in all twelve models.

		augmentation condition			
weighting	statistic	H	HV	HR	HVR
macro	C	$+0.0505$ ±0.0079	$+0.0418$ ±0.0079	-0.0049 ±0.0042	-0.0036 ±0.0030
	H	$+0.0218$ ±0.0026	$+0.0005$ ±0.0043	-0.0015 ±0.0048	$+0.0006$ ±0.0008
	Q	$+0.0051$ ±0.0016	$+0.0015$ ±0.0020	-0.0088 ±0.0019	-0.0025 ±0.0024
inst.-weighted	C	$+0.0207$ ±0.0035	$+0.0086$ ±0.0026	-0.0030 ±0.0004	-0.0021 ±0.0010
	H	$+0.0202$ ±0.0003	$+0.0002$ ±0.0014	$+0.0005$ ±0.0017	$+0.0010$ ±0.0009
	Q	$+0.0048$ ±0.0023	$+0.0004$ ±0.0007	-0.0002 ±0.0024	$+0.0020$ ±0.0022

Their per-view behaviour is consistent with that recipe. With the weights and the prediction protocol fixed, `yolo11n-obb` gives $L_{90}, L_{180}, L_{270} = 0.0000, -0.0035, 0.0038$ at input size 1024 and $0.0012, -0.0005, 0.0023$ at 1536, every view within 0.004 of the canonical one. Our HVR cell, which uses the same three transformations, likewise keeps every view within 0.005 (Section 4). Such checkpoints cannot show which transformation removes which mode, which is why the rest of the paper intervenes on training. The input size affects the released checkpoint and our models differently. Its canonical-view AP50 rises from 0.763 to 0.780 at 1536, whereas for our twenty-four models, trained on single-scale crops, it changes by -0.033 to $+0.007$ (Appendix H).

4 A CONTROLLED AUGMENTATION FACTORIAL

Horizontal flipping is on in every cell, and the two factors are vertical reflection and exposure to rotational transformations: H uses the horizontal flip only, HV adds the vertical flip, HR adds continuous rotation and HVR adds both. We call HR and HVR the rotation-rich cells and H and HV the rotation-poor ones. Every cell is trained with seeds 2027, 2028 and 2029. Within a seed, all cells start from the same byte-identical COCO-initialised checkpoint and share every other hyperparameter, so the augmentation setting is the only thing that varies. All contrasts are formed within a seed.

The two flips are drawn independently, with `fliplr` and `flipud` both at 0.5, so in HV about a quarter of the samples receive both, and the composition of the two flips is exactly the half turn. HV therefore trains on the 180° view but on neither quarter turn, whereas HR and HVR draw rotation angles uniformly from $[-180°, 180°]$. We write "exposure to rotational transformations" instead of naming a group because the augmentation is a finite sample of discretised, interpolated and padded rotations applied to images and boxes, not an average of the learned function over the group. These experiments make no statement about the rotation group.

The half turn removes H. Adding the vertical flip lowers H from 0.0218 to 0.0005 under macro weighting, and adding rotation lowers it to -0.0015 in HR and 0.0006 in HVR; under instance weighting the four cells give 0.0202, 0.0002, 0.0005 and 0.0010. To see whether the three small values differ from zero, we resample seeds with replacement and scenes with replacement within each drawn seed, 1000 times per cell, which carries both sources of variation at once. The 95% interval for H is $[+0.0156, +0.0298]$, and each of the other three includes zero, with $P(H > 0)$ between 0.41 and 0.67. The half-turn mode is therefore present only in the cell that never sees the half turn. For the same reason, the interaction term on H ($+0.0235$) is about as large as H in H and larger than either main effect (Appendix D), as expected in a 2×2 contrast for a quantity that either factor removes on its own.

Only rotation removes C. Adding the vertical flip leaves most of the quarter-turn mode in place ($0.0505 \to 0.0418$ macro), whereas adding rotation removes it (-0.0049 in HR, -0.0036 in HVR). The intervals agree, since C excludes zero in H and HV but not in the rotation-rich cells, so the small overshoot below zero is not resolved. The vertical flip adds the half turn but no quarter turn, and under macro weighting it removes only a small part of C: the rotation main effect is 13.6× the reflection main effect under macro weighting and 3.1× under instance weighting, so the two weightings agree on the ordering but not on how much the flip removes. Most of what the flip does

remove is the half-turn term inside C, since $C = 2\hat{a}_2 + \frac{1}{2}H$ (Section 2). The part of C that does not involve H, $2\hat{a}_2$, is 80–119% of its H value in HV at both capacities, both input sizes and IoU 0.75, under either weighting, while rotation leaves it between -0.0074 and $+0.0049$. The removal by rotation does not depend on the overlap threshold, and at IoU 0.75 rotation removes 106% of the H cell's C under macro weighting and 114% under instance weighting (Appendix J).

Rotation without the vertical flip reverses Q. The chirality is positive in H for every seed ($+0.0051$ macro), near zero in HV and HVR, and negative in HR for every seed (-0.0088, interval $[-0.0143, -0.0016]$). The reversal holds under macro weighting at both capacities but not under instance weighting, where HR gives -0.0002 and one seed is positive, and at `yolo11n` the interval of Q in H includes zero. Because rotation training covers both quarter turns, a missing view cannot explain a negative Q in HR, and Section 6 reports what we tested about it.

The same pattern holds at a second capacity, input size and overlap threshold. We repeated the factorial with `yolo11s-obb`, with three paired seeds per cell and every other setting unchanged. The vertical flip lowers H from 0.0147 to 0.0009 while C stays at 0.0499 against 0.0509, and rotation brings C to -0.0028 in HR and 0.0019 in HVR. H in those two cells is 0.0042 and 0.0053; both intervals include zero, with $P(H > 0)$ of 0.94 and 0.93, so at this capacity the rotation-rich cells may keep a small half-turn cost that three seeds and 456 scenes cannot resolve. Under instance weighting the two capacities agree within 0.003 in every cell. At input size 1536, H in HV, HR and HVR is within 0.002 of zero at both capacities while H keeps 0.0184 and 0.0183, and at IoU 0.75 H falls from 0.0503 in H to 0.0007 in HV, -0.0060 in HR and 0.0029 in HVR (Appendices B, H and J).

A small-angle control. Rotation also perturbs the input more strongly, so HR might remove C through the size of the perturbation and not through covering the quarter turns. A fifth cell, H15, separates the two. It uses `degrees=` 15 and `flipud=` 0, with the same three seeds, the same shared initialisation and everything else identical, so it adds neither the half turn nor a quarter turn. The perturbation is not negligible, since it raises canonical-view AP50 from 0.7036 to 0.7177, the highest of any `yolo11n` cell. Formed within a seed, however, it removes only 0.0030 of C under macro weighting, with a sign that differs between seeds, against 0.0554 for `degrees=` 180. Under instance weighting the figures are 0.0032 against 0.0237, and the half-turn mode behaves the same way (0.0038 against 0.0234, macro); both intervals for H15 exclude zero. A perturbation that improves the detector therefore changes neither mode much, and perturbation strength alone does not explain the effect of rotation. Because the control compares only one small range with the full one, it cannot exclude a sharply non-linear dependence on angle (Section 7). H15 is reported separately and is not part of the 2×2 design, whose cells and definitions were fixed before any of them ran.

A detector with an equivariant backbone keeps less of C. Every intervention above is an augmentation, so we also measure a detector with the symmetry built in. ReDet (Han et al., 2021) has a C_8-steerable backbone (Weiler & Cesa, 2019) and rotation-invariant RoI features, and we train it on our train split alone and score it with the same export and evaluator. Its box regression is not equivariant, so this is a weaker test than an end-to-end equivariant design would give. Its recipe flips each image horizontally, vertically or diagonally (both axes at once), each with probability a quarter, which is the flip distribution of HV and includes the half turn, and it adds no rotation. Head, optimiser, schedule, NMS and crop handling all differ from ours, so the comparison across architectures is not a controlled one (Appendix F). Against HV, ReDet removes 68% of C under macro weighting and 92% under instance weighting, leaving 0.0132 and 0.0007. Its H is 0.0041 with interval $[+0.0003, +0.0078]$, which excludes zero although its flips include the half turn, and 0.0012 under instance weighting. ReDet reaches 0.7601 canonical-view AP50, above every model we trained. We read this comparison as suggestive, not as evidence about which part of the architecture removes C.

5 On DOTA, orientation anisotropy tracks the removal of C

By the relation of Section 2, an effect that depends on box orientation alone can produce C but not H, so the orientation statistics of the boxes are a natural covariate for C. For class k we measure how far its orientation distribution moves under a quarter turn,

$$D_k = \mathrm{TV}(p_k(\theta),\, p_k((\theta + \pi/2) \bmod \pi)) \tag{3}$$

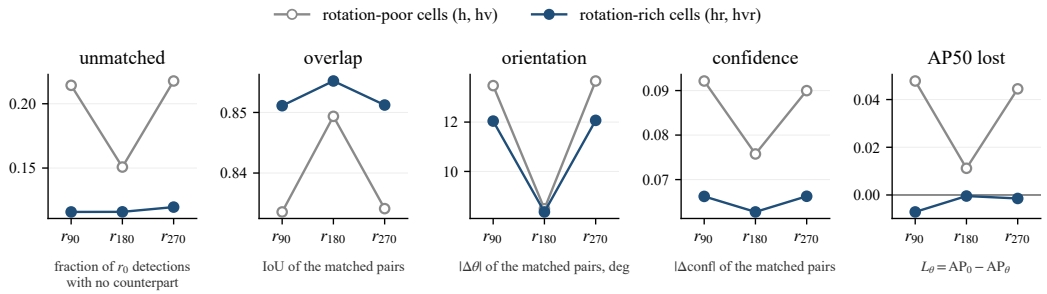

Figure 3: **Rotation training removes the AP cost of the rotated views but not the disagreement.** Four measures of disagreement between a model's own r_0 and r_θ predictions, beside the AP50 lost on each view, averaged over the `yolo11n` models in each regime (six each) and over the 15 classes. Predictions are matched to each other at IoU ≥ 0.5 above confidence 0.25. *Grey, before rotation training:* the quarter turns are the harder views on every measure, and the half turn, which HV sees in training, is the least disturbed. *Blue, after rotation training:* L_θ is at or below zero on all three views, while 0.116, 0.116 and 0.120 of the predictions still have no counterpart, matched pairs overlap at 0.851–0.855 and the confidence changes by 0.063–0.066.

over 36 bins of $[0, \pi)$. Because every class is measured in all four cells, we can test whether rotation exposure removes more C from classes with a larger D_k. The Spearman correlation between D_k and the rotation-rich minus rotation-poor difference in C is $\rho = -0.618$ (permutation $p = 0.018$), so classes with a larger D_k lose more C, and a pooled interaction has the same sign ($\beta = -0.192$, class-level permutation $p = 0.008$). The corresponding tests for H are null ($p = 0.48$ and $p = 0.71$), consistent with the relation.

The effective sample size is 15 classes, since the pooled test uses 180 rows but adds no classes. The prior is also not manipulated. The treatment is the augmentation, and D_k is an observed class property that may covary with other properties. The result is therefore a correlation on DOTA and not an estimate of how much of C the orientation statistics cause. It does not replicate on SODA-A, where C is indistinguishable from zero before any rotation and the class-level relation reverses sign (Appendix I). Appendix C plots the per-class values against the permutation null.

6 AP COST IS NOT EQUIVARIANCE

The factorial measures the AP cost of each rotation. To see how far the same models are from equivariant, we export each model's predictions on r_0 and on each rotated view in the same frame, match the two sets to each other at IoU ≥ 0.5 above confidence 0.25, and average the four parts of E_θ over the six `yolo11n` models of each regime and over the 15 classes. Figure 3 shows them beside the AP50 lost on each view.

Rotation training removes the AP cost, not the disagreement. Before rotation training, the two levels point the same way. In the rotation-poor cells the unmatched fraction is 0.214, 0.151 and 0.218 for the three views, and the half turn, the view with the smallest AP cost, is the least disturbed on all four measures. Rotation training lowers the unmatched fraction to 0.116, 0.116 and 0.120 and raises the overlap of matched pairs from 0.834–0.849 to 0.851–0.855, but it does not remove the disagreement. Counted in the other direction, as the fraction of rotated-view predictions without a counterpart in the canonical view, it is 0.122, 0.124 and 0.122, and the symmetric rate $1 - 2m/(n_0 + n_\theta)$, with m matched pairs among n_0 and n_θ predictions, is 0.119–0.121. On AP, L_{90} is -0.0071 in the rotation-rich cells, and in neither rotation-rich cell, at either capacity, does a rotated view cost AP beyond its seed-and-scene interval. A detector in these cells therefore scores every rotated view about as well as the canonical one, yet more than a tenth of its predictions, matched in either direction, have no counterpart. The predicted orientation moves least under the half turn in both regimes ($8.4°$ against $12.8°$ and $12.9°$ over all twelve models), which is the view that leaves box orientation unchanged. ReDet, whose backbone is equivariant by construction, disagrees less across views. Its unmatched fractions are 0.097, 0.092 and 0.097 (0.101, 0.093 and 0.101 in the

other direction) and its matched pairs overlap at 0.895–0.897, but it still leaves nearly a tenth of its predictions without a counterpart (Appendix F).

Remaining residuals. Three residuals remain. When rotation is added without the vertical flip, Q is negative at both capacities, with intervals $[-0.0143, -0.0016]$ and $[-0.0151, -0.0015]$, although rotation training covers both quarter turns. ReDet keeps a half-turn cost of 0.0041 whose interval $[+0.0003, +0.0078]$ excludes zero, although its flips include the half turn. At `yolo11s` the rotation-rich cells keep half-turn costs of 0.0042 and 0.0053, and on SODA-A the rotation cell keeps 0.0033, with intervals that include zero. We tested one possible explanation for these residuals, a localisation bias tied to the frame shown to the detector, which the map back would turn into different offsets on different views. For each model we fit the mean offset of matched box centres from their labels on view v as $o_v = g + M_v d$, where g is shared by all views, d is the bias in the shown frame and M_v rotates it back, and the fits leave residuals of at most 0.11 pixels. Removing d and re-scoring moves Q in HR from -0.0088 to -0.0068 and from -0.0087 to -0.0072, about a fifth of it. The bias does not explain the half-turn costs. At `yolo11s` they become 0.0056 and 0.0053, and on SODA-A the rotation cell's falls only to 0.0028. We did not run this test on ReDet, nor did we isolate the part of its pipeline, such as the box regression or the rotated NMS, in which its half-turn cost arises, and we do not assign a cause to any of these residuals.

7 LIMITATIONS

Scope. The full factorial covers DOTA-v1.0 and one detector family (YOLO-OBB), at two capacities with three paired seeds each. On a second dataset, SODA-A, we ran only the two ends of the rotation factor. There the half-turn mode behaves as on DOTA and is separable from zero only without rotation. C, however, is indistinguishable from zero before any rotation, so SODA-A cannot show how C is removed, and the class-level relation between orientation anisotropy and C reverses sign (Appendix I). The equivariant baseline of Section 4 is a single architecture measured with our protocol and not a controlled comparison. It does not speak to other equivariant detectors, in particular designs that are equivariant end to end and not only in the backbone (Lee et al., 2024).

Our detectors are weaker than the released checkpoints. The models in the factorial reach 0.697–0.709 canonical-view AP50, against 0.763 for the released `yolo11n-obb` checkpoint measured with the same pipeline. The training recipes differ. The released checkpoint was trained for 200 epochs at batch 32 on multi-scale crops (Section 3), whereas our cells train for 100 epochs at batch 8 on single-scale crops, a budget that allowed twelve paired runs instead of one. Whether the same structure holds for a detector trained to the level of the released checkpoint is untested. The released checkpoint, which uses the transformations of HVR, also has every rotated view within 0.004 of its canonical view.

Augmentation is not a group intervention. `degrees`$= 180$ differs from `flipud` in more than the group element it adds, though the difference is smaller than it may appear. Every cell runs with `scale`$= 0.5$, `translate`$= 0.1$ and `mosaic`$= 1.0$, so an affine warp is applied to every sample in every cell, including those with `degrees`$= 0$, and the extra padding introduced by `degrees`$= 180$ is small (Appendix G). Nor does HV average over its group, since it applies the half turn, as the composition of two independent flips, to about a quarter of the samples. The design also does not control perturbation strength, because a larger sampled angle moves pixels further and introduces more new content. The H15 cell addresses this. It improves canonical accuracy but leaves C and H almost unchanged, which leaves only the narrower possibility that the modes depend on the rotation angle in a sharply non-linear way.

Statistical limits. The orientation-prior tests have an effective sample size of 15, the number of DOTA classes. The pooled regression uses 180 rows but adds no classes, so its smaller p-value is not additional evidence beyond the paired test. Three seeds per cell resolve the large effects on C and H well but the small effects on Q only marginally. Q is close to zero in HV and HVR, in HVR the two class weightings disagree on its sign, and at `yolo11n` the reversal in HR holds under macro weighting only. DOTA's $3{,}066$ labelled validation crops come from 456 source scenes and are not independent, which is why our intervals resample scenes within each drawn seed. The scene level contributes more of the uncertainty than the seed level in all but three of the 36 cell-and-mode

intervals we report, so more seeds would narrow them less than more scenes would. Effects as small as the half-turn costs left in the `yolo11s` rotation-rich cells, with $P(H > 0)$ of 0.94 and 0.93, are at the limit of what these data can resolve.

Fixed measurement choices. Views are exported at confidence 0.001, NMS IoU 0.7 and at most 1000 detections, then scored at 300 detections per image with exact polygon IoU and 101-point AP. All of these were fixed in advance. We later varied the matching IoU for the modes and both thresholds for the disagreement measures (Appendix J), and the detection cap on SODA-A, where the cap binds but moves C and H by at most 0.0013 (Appendix I); how the modes depend on the other choices is untested. We measure only the four axis-aligned views, so "rotation sensitivity" here means sensitivity to rotations by multiples of $90°$.

8 Related work

Oriented object detection. In aerial benchmarks, arbitrarily oriented, densely packed and small objects are the norm (Xia et al., 2018; Ding et al., 2022; Sun et al., 2022; Cheng et al., 2016; 2019; Azimi et al., 2018). A large body of work improves rotated localisation with rotated proposals, aligned features, refined one-stage heads and geometry-aware regression losses (Ding et al., 2019; Yang et al., 2019; 2021b; Han et al., 2022; Xie et al., 2021; Li et al., 2022; Yang et al., 2021c;d), and some methods learn orientation from horizontal boxes alone by enforcing consistency between two rotated views (Yang et al., 2023). We propose no new detector and instead measure a widely used detector family (Ultralytics, 2024).

Rotation equivariance by construction. Group-equivariant, harmonic and steerable networks build rotation structure into the architecture instead of learning it (Cohen & Welling, 2016; Worrall et al., 2017; Marcos et al., 2017; Weiler & Cesa, 2019), and oriented detectors have adopted them: an equivariant backbone (Han et al., 2021), designs that are equivariant end to end including the box regression (Lee et al., 2024), operators that carry rotation and symmetry priors (Zhan et al., 2026), and pre-training that preserves rotation information in the features (Zhang et al., 2024). A related line relaxes the constraint when the symmetry of the data is imperfect (Wang et al., 2022), as it is in aerial imagery. These works mainly report per-rotation results as inputs to an overall robustness summary, and where the equivariance error is measured directly it is also aggregated. For example, Wu et al. (2025) define it on feature maps and average it over the group. We instead study the per-rotation structure, intervene on its components, and separate the prediction-level error from its AP cost. In Section 4, a detector with a C_8-equivariant backbone, scored with our evaluator, keeps about a third of the quarter-turn mode that flips alone leave, together with a small half-turn cost.

Invariance acquired from augmentation. Augmentation is the standard practical route to approximate invariance (Perez & Wang, 2017; Shorten & Khoshgoftaar, 2019; Cubuk et al., 2019), and our factorial probes the gap between it and equivariance by construction. Theoretical work treats augmentation with a group as averaging over its orbits (Chen et al., 2020; Lyle et al., 2020) and shows that building a symmetry in gives a strict benefit over learning it (Elesedy & Zaidi, 2021; Bietti et al., 2021); the extent of an invariance can itself be learned (Benton et al., 2020). These analyses assume an average over the whole group, which augmented training does not compute (Section 4), so equivariance is expected to be approximate, as measurements confirm (Bouchacourt et al., 2021; Gruver et al., 2023). Less attention has been paid to how the approximation varies over the group, with different elements in the augmentation removing different parts of the residual.

Multi-view inference. Residual non-equivariance usually shows up as a test-time augmentation gain, after duplicate predictions from the rotated views have been merged by NMS variants or weighted box fusion (Neubeck & Van Gool, 2006; Bodla et al., 2017; Solovyev et al., 2021). Such a gain is a single number that mixes our modes and hides predictions that fusion corrects in some cases and corrupts in others (Shanmugam et al., 2021). We never fuse views.

AI USE STATEMENT

We used generative AI tools to write and debug code for analysis, evaluation, dataset conversion and plotting, to draft and edit parts of the text, and to check bibliographic records. The research question, the experimental design, every condition that was run and every claim in the paper were decided by the authors. No AI tool generated data or measurements: every reported statistic comes from our evaluation and analysis code and is recomputed from the archived measurements by a checker script in the supplementary material. The authors reviewed all AI-assisted work and take full responsibility for the content.

REPRODUCIBILITY STATEMENT

The quantities reported in this paper are defined in Section 2, and the definitions were fixed before the experimental cells were run. The per-view, per-class AP50 from which every quantity is derived is stored for every checkpoint, so other statistics can be computed without repeating inference. The supplementary material contains the per-model metrics for every cell, input size, class weighting and matching threshold we report, the seed-and-scene intervals, the prediction-level disagreement measures and their threshold grid, the frame-bias fits, the SODA-A, ReDet and released-checkpoint evaluations, the per-class orientation statistics and the back-mapping injection of Appendix K. It also contains the analysis code and a script that recomputes every statistic in the manuscript from these files, fails if any printed value disagrees, and lists what it cannot recompute from them, such as the export thresholds, which are recorded with the exports. The training recipe, including the seeds, the shared initialisation and the augmentation settings of each cell, is given in Section 4 and in the released training scripts; Appendix I describes how SODA-A was prepared.

REFERENCES

Seyed Majid Azimi, Eleonora Vig, Reza Bahmanyar, Marco Körner, and Peter Reinartz. Towards multi-class object detection in unconstrained remote sensing imagery. In *Asian Conference on Computer Vision*, pp. 150–165, 2018.

Gregory Benton, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. Learning invariances in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Alberto Bietti, Luca Venturi, and Joan Bruna. On the sample complexity of learning under geometric stability. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

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

Diane Bouchacourt, Mark Ibrahim, and Ari Morcos. Grounding inductive biases in natural images: Invariance stems from variations in data. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

Gong Cheng, Peicheng Zhou, and Junwei Han. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7405–7415, 2016.

Gong Cheng, Junwei Han, Peicheng Zhou, and Dong Xu. Learning rotation-invariant and Fisher discriminative convolutional neural networks for object detection. *IEEE Transactions on Image Processing*, 28(1):265–278, 2019.

Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023.

Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *International Conference on Machine Learning*, pp. 2990–2999, 2016.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. AutoAugment: Learning augmentation strategies from data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 113–123, 2019.

Jian Ding, Nan Xue, Yang Long, Gui-Song Xia, and Qikai Lu. Learning RoI Transformer for oriented object detection in aerial images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2849–2858, 2019.

Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Ying Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7778–7796, 2022.

Bryn Elesedy and Sheheryar Zaidi. Provably strict generalisation benefit for equivariant models. In *International Conference on Machine Learning*, pp. 2959–2969, 2021.

Nate Gruver, Marc Finzi, Micah Goldblum, and Andrew Gordon Wilson. The Lie derivative for measuring learned equivariance. In *International Conference on Learning Representations*, 2023.

Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. ReDet: A rotation-equivariant detector for aerial object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2786–2795, 2021.

Jiaming Han, Jian Ding, Jie Li, and Gui-Song Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2022.

Chanho Lee, Jinsu Son, Hyounguk Shon, Yunho Jeon, and Junmo Kim. FRED: Towards a full rotation-equivariance in aerial image object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2883–2891, 2024.

Wentong Li, Yijie Chen, Kaixuan Hu, and Jianke Zhu. Oriented RepPoints for aerial object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1829–1838, 2022.

Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*, 2020.

Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *IEEE International Conference on Computer Vision*, pp. 5048–5057, 2017.

Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *International Conference on Pattern Recognition*, volume 3, pp. 850–855, 2006.

Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation. In *IEEE/CVF International Conference on Computer Vision*, pp. 1214–1223, 2021.

Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, 2021.

Xian Sun, Peijin Wang, Zhiyuan Yan, Feng Xu, Ruiping Wang, Wenhui Diao, Jin Chen, Jihao Li, Yingchao Feng, Tao Xu, et al. FAIR1M: A benchmark dataset for fine-grained object recognition in high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 184:116–130, 2022.

Ultralytics. Ultralytics YOLO11. `https://docs.ultralytics.com/models/yolo11/`, 2024. Accessed 2026-06-28.

Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pp. 23078–23091, 2022.

Maurice Weiler and Gabriele Cesa. General E(2)-equivariant steerable CNNs. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5028–5037, 2017.

Xiuyu Wu, Xinhao Wang, Xiubin Zhu, Lan Yang, Jiyuan Liu, and Xingchen Hu. Measuring the impact of rotation equivariance on aerial object detection. In *IEEE/CVF International Conference on Computer Vision*, 2025.

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

Xingxing Xie, Gong Cheng, Jiabao Wang, Xiwen Yao, and Junwei Han. Oriented R-CNN for object detection. In *IEEE/CVF International Conference on Computer Vision*, pp. 3520–3529, 2021.

Xue Yang and Junchi Yan. Arbitrary-oriented object detection with circular smooth label. In *European Conference on Computer Vision*, 2020.

Xue Yang and Junchi Yan. On the arbitrary-oriented object detection: Classification based approaches revisited. *International Journal of Computer Vision*, 130(5):1340–1365, 2022.

Xue Yang, Jirui Yang, Junchi Yan, Yue Zhang, Tengfei Zhang, Zhi Guo, Xian Sun, and Kun Fu. SCRDet: Towards more robust detection for small, cluttered and rotated objects. In *IEEE/CVF International Conference on Computer Vision*, pp. 8232–8241, 2019.

Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15819–15829, 2021a.

Xue Yang, Junchi Yan, Ziming Feng, and Tao He. R3Det: Refined single-stage detector with feature refinement for rotating object. In *AAAI Conference on Artificial Intelligence*, volume 35, pp. 3163–3171, 2021b.

Xue Yang, Junchi Yan, Qi Ming, Wentao Wang, Xiaopeng Zhang, and Qi Tian. Rethinking rotated object detection with Gaussian Wasserstein distance loss. In *International Conference on Machine Learning*, pp. 11830–11841, 2021c.

Xue Yang, Xiaojiang Yang, Jirui Yang, Qi Ming, Wentao Wang, Qi Tian, and Junchi Yan. Learning high-precision bounding box for rotated object detection via Kullback-Leibler divergence. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18381–18394, 2021d.

Xue Yang, Gefan Zhang, Wentong Li, Xuehui Wang, Yue Zhou, and Junchi Yan. H2RBox: Horizontal box annotation is all you need for oriented object detection. In *International Conference on Learning Representations*, 2023.

Jialei Zhan, Li Liu, Jiehua Zhang, Yuhang Xie, Yongxiang Liu, Jiangming Chen, and Ming-Ming Cheng. Rotation invariant and symmetry aware pixel difference network for remote sensing object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13264–13274, 2026.

Yunsheng Zhang, Zijing Ren, Zichen Ding, Hong Qian, Haiqiang Li, and Chao Tao. Co-ECL: Covariant network with equivariant contrastive learning for oriented object detection in remote sensing images. *Remote Sensing*, 16(3):516, 2024.

A THE TWO EVALUATORS

Training-time validation and the evaluator used in this paper report different AP50 values for the same checkpoint. The trainer matches predictions to ground truth with a Gaussian approximation to rotated IoU, which is cheap and differentiable but overestimates the overlap of elongated boxes. Our evaluator computes exact polygon IoU after an axis-aligned prefilter, matches greedily by confidence at IoU ≥ 0.5, keeps at most 300 detections per image, and integrates AP over 101 recall points.

The gap is systematic. For all twelve models of the `yolo11n` factorial the trainer reports the higher value, by $+0.029$ to $+0.050$ AP50 (mean $+0.039$). All per-view AP values, C, H, Q and all contrasts in this paper come from the exact-polygon evaluator, and trainer AP50 is never used in a comparison. We report it so that readers comparing our canonical-view numbers with training logs do not mistake the difference for an inconsistent pipeline.

B THE SECOND CAPACITY

Table 2 gives the factorial at both capacities, and Table 11 gives the seed-and-scene intervals for both.

Table 2: The factorial at two capacities, under macro and instance weighting; all entries are means over three paired seeds. Canonical-view AP50 is 0.697–0.709 for `yolo11n` and 0.719–0.744 for `yolo11s`.

		macro			instance-weighted		
model	cell	C	H	Q	C	H	Q
`yolo11n`	H	$+0.0505$	$+0.0218$	$+0.0051$	$+0.0207$	$+0.0202$	$+0.0048$
	HV	$+0.0418$	$+0.0005$	$+0.0015$	$+0.0086$	$+0.0002$	$+0.0004$
	HR	-0.0049	-0.0015	-0.0088	-0.0030	$+0.0005$	-0.0002
	HVR	-0.0036	$+0.0006$	-0.0025	-0.0021	$+0.0010$	$+0.0020$
`yolo11s`	H	$+0.0509$	$+0.0147$	$+0.0099$	$+0.0228$	$+0.0206$	$+0.0061$
	HV	$+0.0499$	$+0.0009$	-0.0005	$+0.0112$	-0.0012	$+0.0005$
	HR	-0.0028	$+0.0042$	-0.0087	-0.0034	-0.0002	-0.0017
	HVR	$+0.0019$	$+0.0053$	-0.0034	-0.0003	$+0.0022$	-0.0001

The capacities differ in two places. At `yolo11s` the vertical flip removes almost none of C under macro weighting (0.0509 in H against 0.0499 in HV), and two of the three seeds reverse the order. Its rotation-rich cells also keep positive point estimates of H, 0.0042 in HR and 0.0053 in HVR, whose intervals include zero (Section 4); HVR also keeps slightly more C (0.0019) than its `yolo11n` counterpart (-0.0036).

The `yolo11s` H run with seed 2027 stopped early, at epoch 97, under the patience rule shared by all cells. The other two seeds show that the early stop did not favour this cell: the two runs that completed all 100 epochs have a higher macro C and a lower canonical AP50 than the run that stopped.

C THE ORIENTATION-PRIOR TEST

Figure 4 shows the test of Section 5 class by class, with the permutation null beside it.

The null distribution is wide because DOTA has only 15 classes, which is also why we report the result as a correlation on DOTA rather than as an estimate of effect size. The sign of the relation does not depend on any one class: leaving out a single class keeps ρ between -0.727 and -0.530. Its significance is less robust, since the permutation p then lies between 0.005 and 0.053 and exceeds 0.05 only when soccer ball field is left out. Without helicopter, the class that loses the most C, p is 0.047.

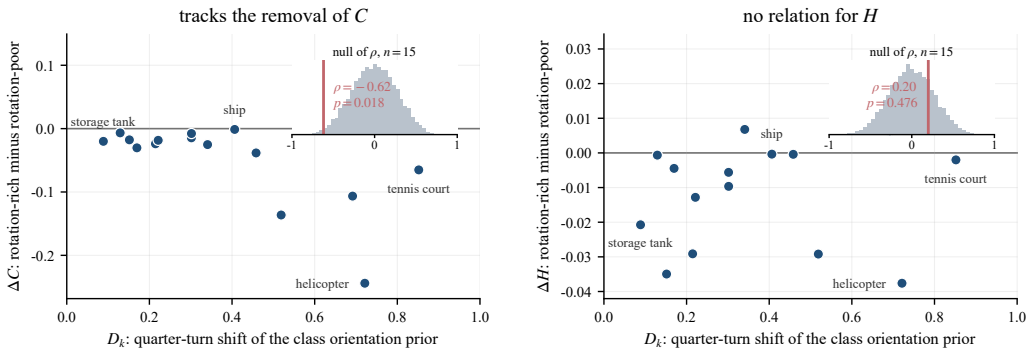

Figure 4: **On DOTA, orientation anisotropy tracks the removal of C but not of H.** Each point is one of the 15 DOTA classes, plotted by its quarter-turn orientation shift D_k against the change in each mode under rotation training (rotation-rich minus rotation-poor). *Left:* the more anisotropic a class is under a quarter turn, the more C it loses; helicopter loses the most, while ship, whose orientations shift by $D_k = 0.41$, loses almost none. *Right:* H shows no such relationship. The insets show the permutation null behind the p-values, 20,000 random reassignments of D_k across classes, with the observed ρ marked; its width reflects the small sample of $n = 15$ classes.

D Factorial contrasts

Table 3 gives the paired contrasts averaged over the three seeds, including the interaction term.

Table 3: Paired contrasts, as means over three seeds, where Y_c is a mode's value in cell c: $d_V = \frac{1}{2}[(Y_{\mathrm{HV}} - Y_{\mathrm{H}}) + (Y_{\mathrm{HVR}} - Y_{\mathrm{HR}})]$, $d_{\mathrm{Rot}} = \frac{1}{2}[(Y_{\mathrm{HR}} - Y_{\mathrm{H}}) + (Y_{\mathrm{HVR}} - Y_{\mathrm{HV}})]$ and $I = Y_{\mathrm{HVR}} - Y_{\mathrm{HR}} - Y_{\mathrm{HV}} + Y_{\mathrm{H}}$.

weighting	metric	d_V	d_{Rot}	I
macro	C	-0.0037	-0.0504	$+0.0100$
	H	-0.0096	-0.0116	$+0.0235$
	Q	$+0.0013$	-0.0089	$+0.0098$
inst.-weighted	C	-0.0056	-0.0173	$+0.0130$
	H	-0.0097	-0.0095	$+0.0205$
	Q	-0.0011	-0.0016	$+0.0066$

Under both weightings the interaction on H is larger than either main effect and close to H in H ($+0.0235$ against 0.0218 macro, $+0.0205$ against 0.0202 instance-weighted). A 2×2 contrast behaves this way when either factor on its own brings a quantity to zero. With H near zero in HV, HR and HVR, both main effects are close to half of $-H$ in H and the interaction is close to H in H. We read H from the cell means and their intervals rather than from the interaction.

E The composite statistic

Table 4 reports the composite $H - C$ and shows why we do not use a single scalar. In the baseline cell H, the three seeds give -0.039, -0.023 and -0.024 under macro weighting but -0.005, $+0.001$ and $+0.002$ under instance weighting, so the sign flips for two of the three seeds. In HV the means are -0.0413 and -0.0085, which agree in sign but differ by a factor of about five.

F The rotation-equivariant baseline

ReDet (Han et al., 2021) builds rotation structure into the backbone with C_8-steerable convolutions (Weiler & Cesa, 2019) instead of learning it from data. We train it on the train split of our own crops.

Table 4: The composite $H - C$ per cell, as means over three seeds. In H the two weightings disagree in sign for two of the three seeds; in HV they agree in sign but differ by a factor of about five.

weighting	H	HV	HR	HVR
macro	-0.0286	-0.0413	$+0.0034$	$+0.0043$
inst.-weighted	-0.0006	-0.0085	$+0.0035$	$+0.0031$

The released weights cannot be used, because their configuration trains on DOTA `trainval` and all 3,066 of our evaluation crops come from DOTA's validation split, so a released checkpoint has seen the evaluation images during training. Export and scoring use the same code as the rest of the paper, with the same confidence floor of 0.001. Its training pipeline applies a horizontal, vertical or diagonal flip, each with probability 0.25, and no rotation. A diagonal flip reverses both axes, so the flip distribution is that of HV.

Table 5: The equivariant baseline beside the factorial cells at input size 1024, under macro and instance weighting; all entries are means over three seeds. ReDet's seed standard deviation is 0.0024 on macro C, 0.0022 on macro H and 0.0003 on instance-weighted H.

	macro			instance-weighted		
	C	H	Q	C	H	Q
H	$+0.0505$	$+0.0218$	$+0.0051$	$+0.0207$	$+0.0202$	$+0.0048$
HV	$+0.0418$	$+0.0005$	$+0.0015$	$+0.0086$	$+0.0002$	$+0.0004$
HR	-0.0049	-0.0015	-0.0088	-0.0030	$+0.0005$	-0.0002
HVR	-0.0036	$+0.0006$	-0.0025	-0.0021	$+0.0010$	$+0.0020$
ReDet	$+0.0132$	$+0.0041$	$+0.0014$	$+0.0007$	$+0.0012$	$+0.0006$

ReDet's prediction-level error is lower on every measure but does not vanish. Its unmatched fractions are 0.097, 0.092 and 0.097, against 0.116, 0.116 and 0.120 in our rotation-rich cells; matched-pair overlap rises from 0.851–0.855 to 0.895–0.897; and the change in predicted orientation under a quarter turn falls from $12.0°$ and $12.1°$ to $8.8°$ and $8.9°$. Nearly a tenth of the canonical-view predictions still have no counterpart on a quarter-turn view, and the half-turn cost that remains (0.0041, interval $[+0.0003, +0.0078]$) is one of the residuals of Section 6.

Several differences make absolute AP incomparable. ReDet trains for 12 epochs with SGD against our 100 epochs with AdamW, it is a two-stage detector rather than a one-stage one, its rotated NMS uses IoU 0.1 instead of 0.7, and its pipeline drops crops without annotated objects, which ours keeps. Its canonical-view AP50 of 0.7601 is above that of all our models, so its remaining C and H are not artefacts of a weak detector, but we do not treat the comparison as like for like. The two weightings also disagree on whether ReDet's C exceeds its H (macro weighting says it does, instance weighting says it does not), the same instability that keeps us from reporting a composite.

G Measuring the warp confound

`degrees=` 180 differs from `flipud` in more than the group element it adds, since a sampled rotation resamples pixels and pads the border and a flip does neither. Two measurements bound how much of the factorial's rotation effect these differences could explain.

First, resampling happens in every cell. Every cell runs with `scale=` 0.5, `translate=` 0.1 and `mosaic=` 1.0, so an affine warp, with its interpolation, border padding and transformation of the boxes, is applied to every sample of every cell, including those with `degrees=` 0. In 800 draws for each of `degrees=` 0 and `degrees=` 180, the composed matrix was never the identity.

Second, the extra padding is small. Measured on the images that training actually sees, the mean padded fraction is 0.004 for `degrees=` 0 against 0.011 for `degrees=` 180 while

mosaic augmentation is on, and 0.216 against 0.236 after it is switched off. The script `tools/measure_warp_confound.py` produces both measurements.

Neither measurement settles the question of perturbation strength, which the H15 cell of Section 4 addresses.

H THE RESOLUTION COMPARISON

Table 6 repeats Table 2 at input size 1536. The pattern of Section 4 is unchanged: at both capacities H is within 0.002 of zero in HV, HR and HVR, the vertical flip leaves most of C, and rotation removes it.

Table 6: The factorial at input size 1536, laid out as Table 2; means over three paired seeds.

		macro			instance-weighted		
model	cell	C	H	Q	C	H	Q
yolo11n	H	+0.0523	+0.0184	+0.0090	+0.0198	+0.0195	+0.0087
	HV	+0.0425	−0.0005	+0.0000	+0.0107	−0.0009	+0.0007
	HR	−0.0060	−0.0003	−0.0055	−0.0019	−0.0003	−0.0023
	HVR	−0.0034	+0.0005	+0.0010	−0.0027	−0.0009	−0.0000
yolo11s	H	+0.0471	+0.0183	+0.0127	+0.0202	+0.0175	+0.0097
	HV	+0.0435	−0.0004	−0.0016	+0.0136	+0.0000	−0.0001
	HR	−0.0065	+0.0017	−0.0029	−0.0042	+0.0001	−0.0020
	HVR	−0.0056	−0.0012	+0.0047	−0.0011	+0.0006	+0.0001

The change of input size moves our models little. Across the twelve `yolo11n` models the cell means of C and H move by at most 0.0034 under macro weighting and 0.0021 under instance weighting, while between-seed standard deviations range from 0.0008 to 0.0098 (macro) and reach 0.0035 (instance-weighted). For the twelve `yolo11s` models the shifts reach 0.0076 (macro) and 0.0031 (instance-weighted) and differ in sign between cells, except for macro C, which falls in all four. The released checkpoint behaves differently: its canonical-view AP50 rises from 0.763 to 0.780, whereas for our twenty-four models it changes by -0.033 to $+0.007$. The released checkpoint was trained on multi-scale crops and ours were not, so it is the only model here for which 1536 lies within the training distribution. This explanation is circumstantial, and confirming it would require retraining the factorial on multi-scale crops, which we did not do.

I A SECOND DATASET

All controlled results in the main text are on DOTA. We repeated the two ends of the rotation factor on SODA-A (Cheng et al., 2023), an oriented aerial benchmark of small objects: cell H (`fliplr` only) and cell HR (`fliplr` and `degrees`$= 180$), with three paired seeds each, the same recipe and launcher, the same byte-identical COCO-initialised checkpoint, and the same four-view protocol, with nine classes instead of fifteen. We tile SODA-A's own `train` and `val` source splits separately, with the same $1024/824$ geometry as DOTA, into 15,901 training and 8,795 evaluation crops. No source image contributes crops to both splits, so no image content is shared between them.

The half-turn mode behaves as on DOTA. Under macro weighting H is 0.0089 in H, with interval $[+0.0045, +0.0124]$, and 0.0033 in HR, with interval $[-0.0008, +0.0074]$; the within-seed change is -0.0056 ± 0.0030, and all three seeds agree on its sign. Under instance weighting H is small in both cells (0.0018 and 0.0019).

SODA-A has no C to remove. C is -0.0029 in H and -0.0013 in HR under macro weighting, with intervals $[-0.0085, +0.0035]$ and $[-0.0057, +0.0027]$, and 0.0009 and 0.0014 under instance weighting. Before any rotation there is no quarter-turn cost for rotation to remove, so SODA-A cannot show how C is removed.

Table 7: SODA-A cell means over three paired seeds, with between-seed standard deviations, beside the corresponding DOTA cells of Table 1. Both datasets use the main protocol at confidence 0.001.

		macro		instance-weighted	
data	cell	C	H	C	H
SODA-A	H	-0.0029 ± 0.0016	$+0.0089$ ± 0.0016	$+0.0009$ ± 0.0016	$+0.0018$ ± 0.0009
SODA-A	HR	-0.0013 ± 0.0025	$+0.0033$ ± 0.0015	$+0.0014$ ± 0.0001	$+0.0019$ ± 0.0005
DOTA	H	$+0.0505$ ± 0.0079	$+0.0218$ ± 0.0026	$+0.0207$ ± 0.0035	$+0.0202$ ± 0.0003
DOTA	HR	-0.0049 ± 0.0042	-0.0015 ± 0.0048	-0.0030 ± 0.0004	$+0.0005$ ± 0.0017

The detection cap binds on SODA-A but does not move the modes. Views are scored with at most 300 detections per image, a limit chosen on DOTA. SODA-A is denser. At the export confidence of 0.001 its canonical-view exports hold about 250 detections per image, and raising the cap changes accuracy substantially. Table 8 gives the cell means at three caps. Canonical AP50 rises from 0.7141 to 0.7482 in H and from 0.7202 to 0.7535 in HR, while C and H move by at most 0.0003 as cell means and by at most 0.0013 in any single model, less than the between-seed standard deviation of either mode.

Table 8: SODA-A at three detection caps: cell means over three paired seeds, macro weighting.

	H			HR		
detections per image	300	500	1000	300	500	1000
canonical AP50	0.7141	0.7371	0.7482	0.7202	0.7429	0.7535
C	-0.0029	-0.0030	-0.0028	-0.0013	-0.0015	-0.0016
H	$+0.0089$	$+0.0089$	$+0.0091$	$+0.0033$	$+0.0033$	$+0.0032$

The class-level test reverses for C. Section 5 tests $\rho(D_k, \Delta C_k)$, where ΔC_k is the rotation-rich minus rotation-poor difference in C for class k, against a permutation null that shuffles D_k across classes. Table 9 gives the same tests on SODA-A, with HR as the rotation-rich cell and H as the rotation-poor one.

Table 9: The class-level tests of Section 5 on both datasets.

	DOTA, 15 classes		SODA-A, 9 classes	
test	ρ or β	perm p	ρ or β	perm p
$\rho(D, \Delta C)$	-0.618	0.018	$+0.733$	0.032
$\rho(D, \Delta H)$	$+0.200$	0.476	-0.150	0.710
pooled $\beta_{D \cdot \mathrm{rot}}$ on C	-0.192	0.008	$+0.088$	0.001
pooled $\beta_{D \cdot \mathrm{rot}}$ on H	$+0.007$	0.712	$+0.036$	0.363

The H rows agree, since on neither dataset does orientation anisotropy track the change in the half-turn mode. The C rows disagree. They reverse sign, with nominal significance, under both the primary and the pooled test, and the reversal does not depend on any one class, because leaving out a single class keeps $\rho(D, \Delta C)$ between $+0.619$ and $+0.857$. On SODA-A the class that loses the most C under rotation is the least anisotropic one. `windmill`, with $D_k = 0.003$, loses 0.0149, while `helicopter`, with $D_k = 0.380$, starts with a negative C and moves the other way. These per-class changes are differences between quantities that are indistinguishable from zero at the cell level, which limits how much weight the reversal can carry, and we conclude only that the relation found on DOTA does not generalise.

Between datasets. SODA-A is far less anisotropic under a quarter turn than DOTA. Its mean D_k is 0.150 across its nine classes against 0.371 across DOTA's fifteen (0.131 against 0.284 with

instance weighting), and two of its classes are nearly rotationally symmetric (`windmill` at 0.003 and `storage-tank` at 0.029). Across the two datasets, lower anisotropy goes with a smaller C, whereas within SODA-A the per-class relation runs the other way. We rely on neither observation, because a comparison between two datasets is not evidence and we made it after seeing SODA-A's C.

The chirality does not reverse. On SODA-A, Q is negative in both cells (-0.0041 in H and -0.0043 in HR, macro), so we do not claim that the sign reversal of Section 4 generalises.

J STRICTER OVERLAP, OTHER THRESHOLDS AND SEED-AND-SCENE INTERVALS

A stricter overlap. Table 10 gives the modes at matching IoU 0.50 and 0.75 from the exports of the main protocol. At IoU 0.75 the pattern of Section 4 is unchanged. Under macro weighting H falls from 0.0503 in H to 0.0007 in HV, -0.0060 in HR and 0.0029 in HVR. The vertical flip keeps 64% of the H cell's C under macro weighting and 29% under instance weighting, and rotation removes 106% and 114% of it. The rotation main effect on C is $5.5\times$ the reflection main effect under macro weighting and $2.5\times$ under instance weighting. Under instance weighting H in HR is -0.0278 at this threshold, with the half turn scoring above the canonical view, and we have no interval for it at IoU 0.75.

Table 10: The two modes at matching IoU 0.50 and 0.75, as cell means over three seeds.

	C, macro		H, macro		C, inst.-weighted		H, inst.-weighted	
cell	.50	.75	.50	.75	.50	.75	.50	.75
H	$+0.0505$	$+0.0555$	$+0.0218$	$+0.0503$	$+0.0207$	$+0.0647$	$+0.0202$	$+0.0976$
HV	$+0.0418$	$+0.0353$	$+0.0005$	$+0.0007$	$+0.0086$	$+0.0187$	$+0.0002$	$+0.0025$
HR	-0.0049	-0.0033	-0.0015	-0.0060	-0.0030	-0.0090	$+0.0005$	-0.0278
HVR	-0.0036	-0.0004	$+0.0006$	$+0.0029$	-0.0021	-0.0008	$+0.0010$	$+0.0033$

Why the instance-weighted columns grow. A stricter overlap penalises small objects most, because a fixed localisation error costs a small box proportionally more IoU, and small classes dominate DOTA's instance count: ships, small vehicles and large vehicles account for $37{,}935$ of the $55{,}804$ validation objects. Instance weighting thus concentrates on the classes that IoU 0.75 penalises most, whereas macro weighting gives a bridge the same weight as a ship. The macro columns are the ones to compare with the rest of the paper.

Figure 3 under other thresholds. Figure 3 uses one confidence floor (0.25) and one matching IoU (0.5). The floor applies to both views, and predictions are matched one to one within each crop and class, greedily in descending IoU. We recomputed its statistics on a 5×3 grid, with confidence in $\{0.10, 0.15, 0.25, 0.35, 0.50\}$ and matching IoU in $\{0.30, 0.50, 0.70\}$, for the same twelve `yolo11n` models. That more than a tenth of the canonical-view predictions in the rotation-rich cells lack a counterpart on a quarter-turn view holds at thirteen of the fifteen settings. The quarter-turn unmatched fraction ranges from 0.238, at confidence 0.10 and matching IoU 0.70, down to 0.097, at confidence 0.50 and matching IoU 0.30, and the two settings below a tenth are both at confidence 0.50. Its level depends on the thresholds, since a stricter matching IoU raises it and a stricter confidence floor lowers it, but its flatness across θ does not. At every setting the half turn is within 0.0044 of the best of the three views on the unmatched fraction, the overlap and the confidence change, and the predicted orientation moves least under the half turn, 3.4–3.8° less than under the better of the two quarter turns.

Seed-and-scene intervals. DOTA's $3{,}066$ labelled crops come from 456 source scenes and are not independent, and a cell has only three seeds. The intervals of the main text therefore resample both levels: seeds are drawn with replacement, scenes are drawn with replacement within each drawn seed, and the mode is averaged over the drawn seeds, 1000 times per cell. Each model's predictions

are matched to the labels once per view, and a resample reweights the matched predictions by how often their crop is drawn, which gives exactly the AP of scoring the repeated crops. Two single-level references come from the same resamples. The *scene* reference keeps the three observed seeds fixed, and the *seed* reference uses every scene every time. Table 11 gives the intervals for every cell.

Table 11: Cell means of H, C and Q under macro weighting, with 95% intervals that resample seeds and scenes together (1000 resamples per cell).

model	cell	H	C	Q
yolo11n	H	$+0.0218$ $[+0.0156, +0.0298]$	$+0.0505$ $[+0.0389, +0.0641]$	$+0.0051$ $[-0.0023, +0.0122]$
	HV	$+0.0005$ $[-0.0052, +0.0066]$	$+0.0418$ $[+0.0298, +0.0516]$	$+0.0015$ $[-0.0029, +0.0071]$
	HR	-0.0015 $[-0.0077, +0.0054]$	-0.0049 $[-0.0107, +0.0021]$	-0.0088 $[-0.0143, -0.0016]$
	HVR	$+0.0006$ $[-0.0047, +0.0077]$	-0.0036 $[-0.0086, +0.0024]$	-0.0025 $[-0.0094, +0.0048]$
yolo11s	H	$+0.0147$ $[+0.0093, +0.0240]$	$+0.0509$ $[+0.0351, +0.0603]$	$+0.0099$ $[+0.0041, +0.0170]$
	HV	$+0.0009$ $[-0.0049, +0.0065]$	$+0.0499$ $[+0.0292, +0.0586]$	-0.0005 $[-0.0057, +0.0054]$
	HR	$+0.0042$ $[-0.0012, +0.0092]$	-0.0028 $[-0.0086, +0.0031]$	-0.0087 $[-0.0151, -0.0015]$
	HVR	$+0.0053$ $[-0.0013, +0.0120]$	$+0.0019$ $[-0.0026, +0.0074]$	-0.0034 $[-0.0100, +0.0042]$
yolo11n	H15	$+0.0180$ $[+0.0101, +0.0296]$	$+0.0475$ $[+0.0378, +0.0582]$	$+0.0086$ $[+0.0010, +0.0164]$
SODA-A	H	$+0.0089$ $[+0.0045, +0.0124]$	-0.0029 $[-0.0085, +0.0035]$	-0.0041 $[-0.0082, +0.0001]$
	HR	$+0.0033$ $[-0.0008, +0.0074]$	-0.0013 $[-0.0057, +0.0027]$	-0.0043 $[-0.0103, -0.0000]$
ReDet	—	$+0.0041$ $[+0.0003, +0.0078]$	$+0.0132$ $[+0.0071, +0.0214]$	$+0.0014$ $[-0.0025, +0.0059]$

H excludes zero only in the cells trained without the half turn (H at both capacities, H15, and H on SODA-A) and in ReDet. C excludes zero in H and HV at both capacities, in H15 and in ReDet, and in no rotation-rich cell. Q excludes zero, negative, in HR at both capacities and, only just, on SODA-A; it is positive and excludes zero in H at `yolo11s` and in H15, but not in H at `yolo11n`. The scene level contributes more of the uncertainty than the seed level in all but three of the 36 intervals, the exceptions being H in H15, Q in HVR at `yolo11s` and Q in ReDet.

K A CHECK ON THE BACK-MAPPING

Two forms of the inverse. Box coordinates are continuous: a box that covers pixel columns x_0 to $x_1 - 1$ spans $[x_0, x_1]$, and an image of width w spans $[0, w]$. In these coordinates the inverse of r_{90}, which is `np.rot90` with $k = 1$, sends a point (x, y) of the rotated view to $(w-y, x)$ in the original image. Written for pixel indices, the same inverse sends pixel (x, y) to pixel $(w - 1 - y, x)$, which is right for a pixel and one pixel off for a box. On r_{90}, r_{180} and r_{270} the index form moves every box one pixel towards the origin, along x, along both axes and along y respectively, and it leaves r_0 unchanged. In a test against `np.rot90`, rectangles are drawn pixel by pixel into square and non-square arrays, the arrays are rotated, and the continuous inverse has to recover each rectangle's extent exactly from the rotated array, whereas the index form misses it by that pixel.

What the index form does to the modes. We took the predictions of the twelve `yolo11n` models of Section 4, mapped back exactly, and moved every box on r_{90}, r_{180} and r_{270} one pixel towards the origin on the axes above, leaving r_0 alone. This reproduces the index form exactly, except that a coordinate clipped to the far edge of a crop can end up to a pixel from where the index form would put it; at most 4.3% of the moved coordinates lie on an edge. Scored by the same evaluator (Table 12), the index form raises H in every cell, by 0.0067–0.0149 at matching IoU 0.50 and by 0.0577–0.1017 at IoU 0.75, and it raises H more than C, which gains 0.0028–0.0061 and 0.0275–0.0572; the half turn is the only view it displaces along both axes. Under the index form H is positive in all twelve models at both thresholds, including the six in HR and HVR. Because r_0 is untouched, canonical-view AP does not change, so no check on the canonical view alone can reveal the error. The mean offset of matched box centres from their labels does reveal it. Matching predictions at confidence ≥ 0.25 to labels at IoU ≥ 0.5 and averaging over the twelve models, the index form moves that offset by 0.97–0.98 pixels along each displaced axis and by less than 0.01 pixels along an axis it does not displace.

Which implementations are affected. The error arises when the inverse of an array rotation is derived with pixel-index arithmetic and then applied to box coordinates, for instance when an evaluation routine rotates the image with `np.rot90` and maps each vertex back with $w - 1 - y$. Code that maps boxes back with the continuous formulas, or that reuses the geometric transforms of a detection framework, is not affected. The horizontal and vertical flips of Ultralytics, for example, send x to $w - x$ and y to $h - y$, and composing them reproduces our continuous half turn exactly. We have not surveyed how often the index form occurs in published evaluation code. It can be detected with the synthetic test above, or by comparing the mean offsets of matched box centres under the index and continuous forms, which differ by about one pixel along the displaced axes.

Table 12: H and C under the exact inverse (exact) and their change when the same predictions are mapped back with the pixel-index form (Δ), as cell means over three seeds under macro weighting.

	H				C			
	IoU 0.50		IoU 0.75		IoU 0.50		IoU 0.75	
cell	exact	Δ	exact	Δ	exact	Δ	exact	Δ
H	$+0.0218$	$+0.0149$	$+0.0503$	$+0.1017$	$+0.0505$	$+0.0061$	$+0.0555$	$+0.0572$
HV	$+0.0005$	$+0.0107$	$+0.0007$	$+0.0926$	$+0.0418$	$+0.0040$	$+0.0353$	$+0.0505$
HR	-0.0015	$+0.0067$	-0.0060	$+0.0577$	-0.0049	$+0.0028$	-0.0033	$+0.0275$
HVR	$+0.0006$	$+0.0090$	$+0.0029$	$+0.0722$	-0.0036	$+0.0046$	-0.0004	$+0.0388$