Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.36995

Cross-Domain Robustness of Six Established Object Detector Configurations under a Unified Protocol

Yifei Wang

School of Computer Science and Engineering, University of New South Wales, Sydney, Australia
z5529672@ad.unsw.edu.au

Abstract. This paper compares the cross-domain robustness of six established object detector configurations under a unified data protocol, a fixed training budget, and a single evaluation pipeline. All configurations are trained on the union of PASCAL Visual Object Classes (VOC) 2007 and VOC 2012 trainval, comprising 16551 images, under a fixed budget of 18 epochs of source-data exposure, with horizontal flipping at probability 0.5 as the only augmentation and Common Objects in Context (COCO)-pretrained complete detector initialisation. Each configuration is trained under three random seeds. Evaluation covers the VOC 2007 test set, Clipart1k, and Watercolor2k, with all predictions routed through the same pycocotools evaluator. Watercolor degradation is measured against a matched six-category VOC baseline. The Spearman rank correlation between in-domain and Clipart1k average precision at an intersection-over-union threshold of 0.50 is 0.257 across the six configurations. Single Shot Detector (SSD300) ranks fifth in-domain at 0.726 and first on both target domains, whereas Real-Time Detection Transformer (RT-DETR-l) ranks first in-domain at 0.831 and second on both. Observed cross-domain seed variability exceeds in-domain variability for every configuration by factors between 3.4 and 16.2.

Keywords: object detection, cross-domain robustness, ranking stability, evaluation protocol, seed variability

1. Introduction

Object detection is a core task in computer vision. Over the past decade, two-stage, single-stage, and end-to-end Transformer detectors have been introduced in succession [1-3]. Standard benchmarks evaluate models on held-out data from the same distribution, whereas shifts in visual appearance can cause substantial performance loss [4].

Much cross-domain research has focused on domain adaptation, where target-domain images are available during training and used for feature alignment or pseudo-labelling [5]. More recent work studies source-only single-domain generalization, including VOC-to-artistic-domain settings [6-8]. These studies aim to improve target-domain generalization rather than to examine whether the in-domain ranking of established configurations predicts their ranking on unseen domains under one common protocol.

This study establishes a unified data, training-budget, and evaluation protocol to compare six established object-detector configurations across in-domain and cross-domain settings. It shows that

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.36995

in-domain average precision has limited predictive value for cross-domain performance, that detector rankings are not preserved under domain shift, and that cross-domain seed variability substantially exceeds in-domain variability. These findings indicate that detector selection based solely on in-domain benchmarks may misestimate real-world cross-domain robustness.

The unit of comparison is the detector configuration, that is, the combination of a model and its training configuration, rather than the architecture in isolation. No claim is made that an architectural mechanism causes a given model to be more robust; conclusions are stated per configuration.

2. Related work

2.1. Representative object detectors

Two-stage detection is represented by Faster Region-based Convolutional Neural Network (R-CNN), which generates candidate regions with a region proposal network and then classifies and regresses them [1]. Single-stage detection is represented by SSD, which predicts densely over multi-scale feature maps [2]. RetinaNet introduces a focal loss to address foreground-background imbalance [9]. Fully Convolutional One-Stage (FCOS) removes the preset anchor mechanism and predicts object boundaries per pixel [10]. YOLOv8 revises the detection head and loss design of the YOLO family [11]. RT-DETR is an end-to-end Transformer detector that requires no non-maximum suppression (NMS) at inference [3]. The six detectors span two-stage, single-stage, and end-to-end paradigms and cover CNN and Transformer backbones.

2.2. Cross-domain evaluation sets

Clipart1k and Watercolor2k were introduced by Inoue et al. as artistic-domain datasets for cross-domain object detection [4]. Clipart1k shares the 20 PASCAL VOC categories, while Watercolor2k covers six of them [12]. Their texture, colour, and edge statistics differ markedly from natural images, and their category systems remain compatible with VOC, which makes them suitable as unseen target domains. Prior work has mainly used them to validate adaptation methods rather than to compare detectors under one protocol.

2.3. Consistency of evaluation protocols

Framework-native detection outputs and evaluation settings are not necessarily comparable across implementations. Paniego et al. report that confidence thresholds, per-image detection limits, and post-processing conventions differ between frameworks and affect reported accuracy [13]. Comparing values across frameworks therefore conflates model differences with evaluation differences, so this study routes all predictions through one pipeline.

3. Methodology

3.1. Datasets and domain partition

The source training set is the union of VOC 2007 trainval and VOC 2012 trainval, comprising 16551 images and 47223 annotated objects across the 20 VOC categories. All six configurations use exactly the same source data with no further subdivision.

DOI: 10.54254/2755-2721/2026.36995

Evaluation covers four views. The in-domain views are the 20-category evaluation of the VOC 2007 test set, denoted VOC20, and its six-category subset, denoted VOC6. The cross-domain views are the 20-category evaluation of Clipart1k and the six-category evaluation of Watercolor2k. Watercolor2k contains only six categories, so comparing it against VOC20 would introduce a category-set mismatch; the six matched categories of the VOC 2007 test set therefore form its degradation baseline. Table 1 summarises the partition.

Table 1. Dataset partition statistics

Partition	Images	Objects	Classes	Role
VOC 2007 trainval	5011	15662	20	source training
VOC 2012 trainval	11540	31561	20	source training
Source union	16551	47223	20	source training
VOC 2007 test (VOC20)	4952	14976	20	in-domain
VOC 2007 test (VOC6)	4952	8633	6	matched baseline
Clipart1k test	500	1526	20	cross-domain
Watercolor2k test	1000	1654	6	cross-domain

Object counts come from the COCO-format conversion of the original annotations and retain every instance, including those flagged difficult in the VOC devkit; they therefore exceed the non-difficult subsets often quoted. For Clipart1k and Watercolor2k the official test splits of 500 and 1000 images are used and counted directly, since the source paper reports only full-dataset totals of 3165 and 3315 instances; the equally sized training splits are not used. The six matched categories are bicycle, bird, car, cat, dog, and person, with identifiers 2, 3, 7, 8, 12, and 15. No evaluation set is accessed during training or protocol development.

3.2. Study-level controls

Experimental parameters are separated into two layers. Layer one comprises study-level controls identical across all six configurations. Layer two comprises model-specific optimisation settings, which may differ but must derive from the official reference implementation of the framework used. Table 2 lists the layer-one controls.

Table 2. Study-level controls across all six configurations

Control	Value
Source training data	VOC 2007 and VOC 2012 trainval union, 16551 images
Training budget	18 epochs of source-data exposure
Augmentation	horizontal flip, probability 0.5; no mosaic, no mixup
Initialisation	COCO-pretrained complete detector
Class-output policy	VOC class outputs reinitialised; no name-based partial transfer
Random seeds	0, 1, 2

DOI: 10.54254/2755-2721/2026.36995

Table 2. (continued)

Mixed precision	disabled
Target isolation	VOC 2007 test, Clipart1k, Watercolor2k excluded from training and protocol selection
Evaluator	single pycocotools pipeline
Confidence floor	0.001
Detections per image	100

The budget means every model passes over the same 16551 images 18 times. What is held constant is the number of image presentations, not optimiser updates, since batch sizes differ. Augmentation is a layer-one control because it is itself a domain-generalisation intervention; had each model used its native pipeline, cross-domain differences could not be separated from augmentation differences.

3.3. Model-specific optimisation

Layer-two parameters follow a single-GPU adaptation rule. All torchvision models are implemented in PyTorch [14]. The official torchvision detection reference configuration is defined for eight-GPU training, so the reference per-GPU batch size is preserved and the reference learning rate divided by eight. The rule involves no per-model selection.

Learning-rate milestones are compressed proportionally into the 18-epoch budget with round-half-up applied uniformly. The linear warmup specified for epoch zero is retained unchanged, since it does not conflict with the budget. For the two Ultralytics configurations, the optimiser and learning rate resolved by the framework under an 18-epoch budget are frozen explicitly and automatic resolution is disabled at run time. Table 3 gives the settings.

Table 3. Model-specific optimisation settings

Configuration	Batch	Optim.	LR	Decay	Steps	Warmup
Faster R-CNN R50-FPN	2	SGD	0.0025	0.0001	11, 15	1000 iter.
SSD300-VGG16	4	SGD	0.00025	0.0005	12, 17	1000 iter.
RetinaNet R50-FPN	2	SGD	0.00125	0.0001	11, 15	1000 iter.
FCOS R50-FPN	2	SGD	0.00125	0.0001	11, 15	1000 iter.
YOLOv8n	16	AdamW	0.000417	0.0005	linear	3 epochs
RT-DETR-l	8	AdamW	0.000417	0.0005	linear	3 epochs

The decay factor is 0.1 for all four torchvision configurations. The two Ultralytics configurations use a nominal batch size of 64 and a final learning-rate factor of 0.01.

3.4. Unified evaluation protocol

Detections from all six configurations pass through the same pipeline, comprising a common category mapping, a common box validity check, a confidence floor of 0.001, and a limit of 100

DOI: 10.54254/2755-2721/2026.36995

detections per image. Framework-native post-processing is retained upstream of this pipeline: RT-DETR-l applies no NMS, whereas the torchvision configurations apply their reference NMS thresholds of 0.50 for Faster R-CNN and RetinaNet, 0.45 for SSD300, and 0.60 for FCOS, and YOLOv8n retains its framework NMS. Every configuration is evaluated using the weights saved at the end of epoch 18, with no checkpoint selection based on validation accuracy.

3.5. Degradation measures

Let APs denote average precision on the source view and APt denote average precision on the corresponding target view. Absolute degradation and relative degradation are defined as follows.

$$D_{abs} = AP_s - AP_t \tag{1}$$

$$D_{rel} = \frac{AP_s - AP_t}{AP_s} \times 100\% \tag{2}$$

Here Dabs is the absolute fall in average precision and Drel the same quantity expressed as a percentage of the source value. Clipart degradation is referenced to VOC20 and Watercolor degradation to VOC6. Ranking preservation is measured by the Spearman rank correlation coefficient.

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{3}$$

Here di is the difference in rank of configuration i between two evaluation views and n is the number of configurations compared. Since n is six, the coefficient is used descriptively and no significance level is reported.

3.6. Protocol freeze and pre-registration

All protocol parameters were frozen before any training curve was examined, and the frozen state is recorded under version control; every run verifies that the code matches that record. Before formal training, each torchvision configuration completed 1500 consecutive source-only minibatches under each seed, checking loss, gradient, and parameter finiteness with memory occupancy. All twelve checks passed and no evaluation set was accessed.

It is stated in advance that if a configuration yields an in-domain average precision below expectation, the result is reported as obtained and discussed as an observation about fixed-budget adaptation; the learning rate, budget, and schedule are not adjusted and the run is not repeated.

4. Results and analysis

All 18 model-seed combinations completed. Values below are means over three seeds with sample standard deviations.

DOI: 10.54254/2755-2721/2026.36995

4.1. Experiment 1: in-domain performance

RT-DETR-l attains the highest Average Precision at an IoU Threshold of 0.5 (AP50) at 0.831, followed by Faster R-CNN at 0.799, RetinaNet at 0.784, and FCOS at 0.778; SSD300 reaches 0.726 and YOLOv8n 0.690. Figure 1 plots these values with error bars over three seeds; the in-domain bars are closely grouped while the cross-domain bars are lower and reordered. Under the stricter Average Precision averaged over IoU thresholds 0.50–0.95 (AP50:95) measure, given in Table 4, the margin of RT-DETR-l widens to 0.650 while SSD300 falls to 0.453, and FCOS overtakes both Faster R-CNN and RetinaNet. The two measures therefore separate the configurations differently.

Table 4. Average precision at AP50:95 across evaluation views

Configuration	VOC20	VOC6	Clipart20	Watercolor6
Faster R-CNN	0.530 (0.000)	0.578 (0.001)	0.143 (0.003)	0.230 (0.006)
SSD300	0.453 (0.001)	0.506 (0.002)	0.156 (0.005)	0.258 (0.006)
RetinaNet	0.545 (0.001)	0.609 (0.001)	0.131 (0.001)	0.231 (0.005)
FCOS	0.548 (0.001)	0.607 (0.003)	0.146 (0.003)	0.233 (0.008)
YOLOv8n	0.486 (0.001)	0.532 (0.002)	0.112 (0.005)	0.177 (0.012)
RT-DETR-l	0.650 (0.003)	0.704 (0.004)	0.168 (0.007)	0.274 (0.011)

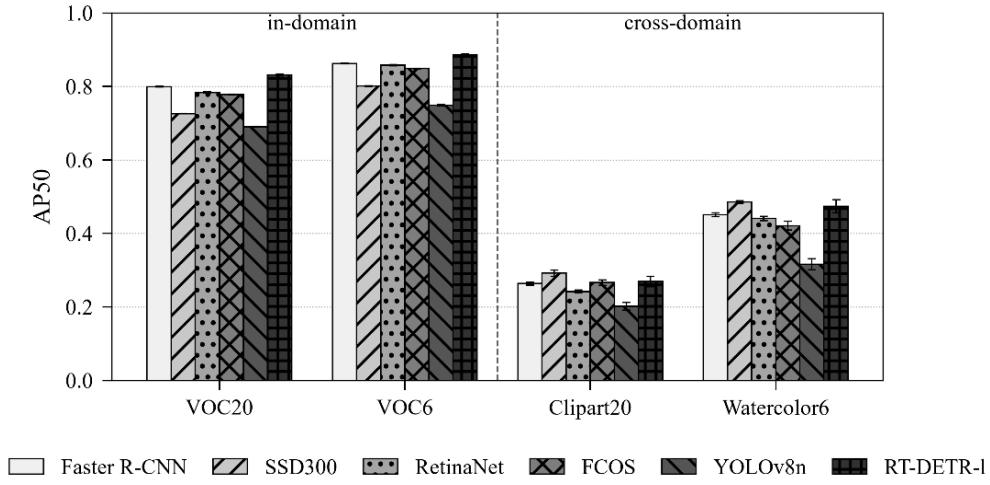

Figure 1. In-domain and cross-domain average precision by configuration (photo credit: original)

4.2. Experiment 2: cross-domain degradation

Degradation is shown in Figure 2. Measured on AP50, relative degradation ranges from 59.8% to 70.7% on Clipart1k and from 39.4% to 57.8% on Watercolor2k. Every configuration degrades more on Clipart1k than on Watercolor2k, consistent with the greater departure of clipart imagery from natural texture and colour statistics.

The two measures do not induce the same ordering. On Clipart1k, RT-DETR-l shows the largest absolute degradation at 0.562 while its relative degradation of 67.6% is below the 70.7% of YOLOv8n, because absolute degradation is bounded by the source baseline. Reporting one measure

DOI: 10.54254/2755-2721/2026.36995

alone can therefore support opposite conclusions, so both are given. Figure 2 contrasts the two measures, which rank the configurations differently.

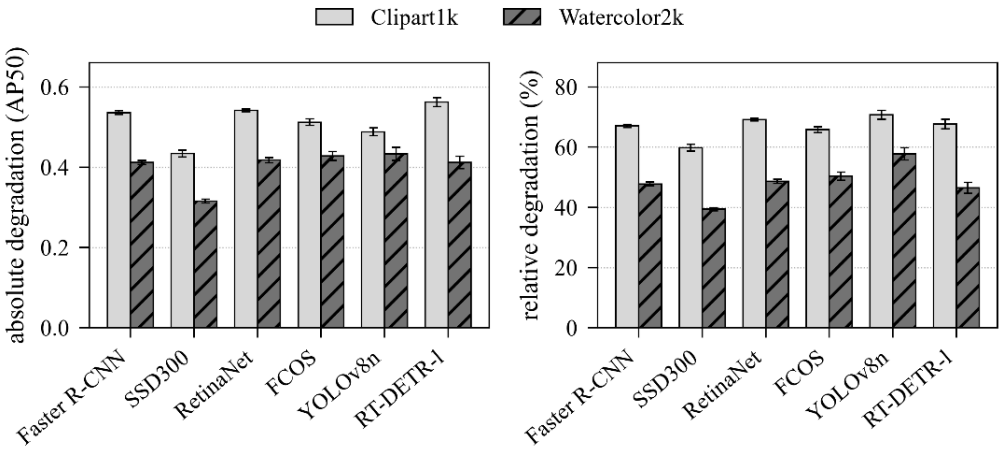

Figure 2. Absolute and relative degradation by configuration and target domain (photo credit: original)

4.3. Experiment 3: ranking preservation

The in-domain ranking is not preserved under domain shift. Measured on AP50, SSD300 ranks fifth on VOC20 but first on both Clipart1k and Watercolor2k, while RT-DETR-l ranks first in-domain and second on both target domains. FCOS rises from fourth in-domain to third on Clipart1k but falls to fifth on Watercolor2k, so the reordering is not consistent between the two target domains either.

Across the six configurations the Spearman rank correlation between in-domain AP50 and Clipart AP50 is 0.257; between VOC6 and Watercolor6 it is 0.429. Both indicate limited predictive value of in-domain performance for performance on an unseen domain. Because n is six, the confidence interval of these coefficients is wide; they are used descriptively and support no statistical inference. Figure 3 traces each configuration across the three views; SSD300 rises from fifth to first while YOLOv8n stays sixth throughout.

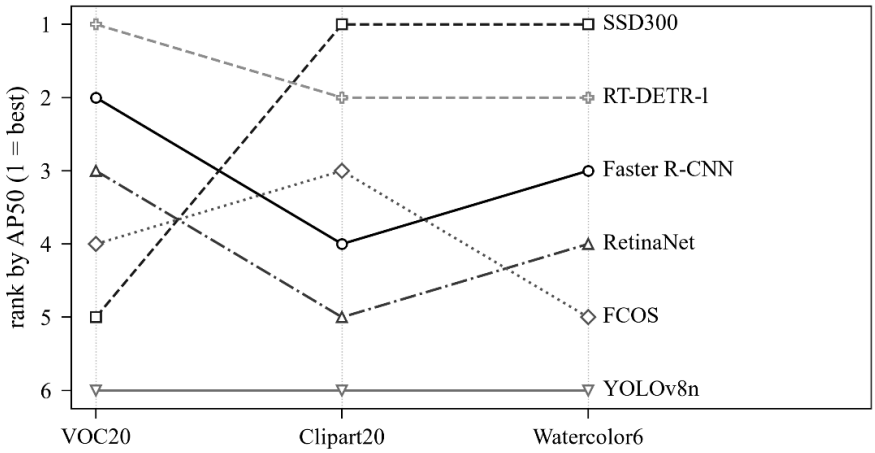

Figure 3. Change in rank position under domain shift (photo credit: original)

DOI: 10.54254/2755-2721/2026.36995

4.4. Experiment 4: seed variability

Table 5 reports the standard deviation of AP50 across three seeds for every configuration. In-domain standard deviations lie between 0.0005 and 0.0029, whereas Clipart1k standard deviations lie between 0.0038 and 0.0140. The ratio of cross-domain to in-domain variability is above one for all six configurations, ranging from 3.4 to 16.2 with a median of 8.7. Run-to-run variation is therefore observed to be larger cross-domain than in-domain for every configuration examined. Figure 4 shows the three seed values per configuration, almost coincident in-domain and visibly separated cross-domain.

This bears directly on between-configuration comparison. On Clipart1k the four middle-ranked configurations span 0.242 to 0.292 in AP50, and the gaps separating RT-DETR-l, FCOS, and Faster R-CNN are 0.003 and 0.003, both smaller than the standard deviation of either configuration. Under these conditions it is not warranted to claim that any of these three outperforms the others on that view, and only SSD300 at the top and YOLOv8n at the bottom are separated by margins exceeding seed variability.

Table 5. Seed variability of AP50 and cross-domain amplification

Configuration	VOC20	VOC6	Clipart20	Watercolor6	Clipart/VOC20
Faster R-CNN	0.0010	0.0008	0.0038	0.0053	3.8
SSD300	0.0005	0.0011	0.0081	0.0037	16.2
RetinaNet	0.0012	0.0017	0.0041	0.0059	3.4
FCOS	0.0006	0.0007	0.0075	0.0118	12.5
YOLOv8n	0.0008	0.0026	0.0107	0.0147	13.4
RT-DETR-l	0.0029	0.0025	0.0140	0.0174	4.8

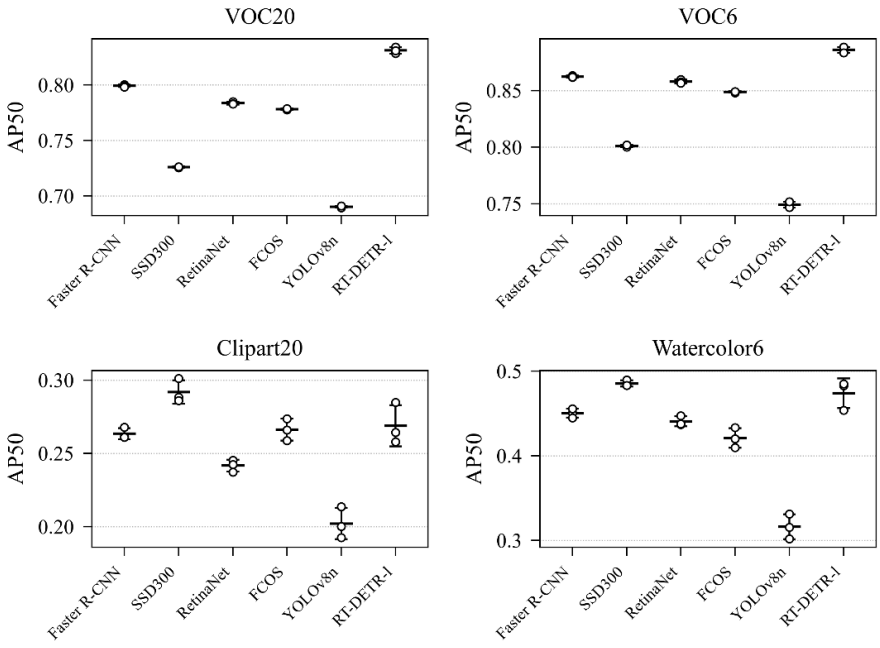

Figure 4. Seed variability on in-domain and cross-domain views (photo credit: original)

DOI: 10.54254/2755-2721/2026.36995

4.5. Experiment 5: detection count and truncation

The common per-image detection limit constrains the configurations to markedly different degrees, as shown in Table 6 and Figure 5. RT-DETR-l averages exactly 100.0 detections per image on VOC20 and Clipart20, meaning every image reaches the cap; RetinaNet and FCOS average above 96 on both views; SSD300 averages above 86. Faster R-CNN and YOLOv8n, by contrast, average between 18 and 27, using roughly a quarter of the allowance.

The cap is therefore active for four configurations and inactive for two. It takes the same value everywhere, but its binding strength is not uniform, which should be considered when interpreting recall-related measures. The difference originates upstream: framework-native detection limits are 100 for Faster R-CNN and FCOS, 200 for SSD300, and 300 for RetinaNet, so a larger native limit delivers more candidates to the shared truncation step. Figure 5 shows the per-image distributions on VOC20 and Clipart20; the percentage above each box is the share of images reaching the cap, and boxes collapse to a line where the interquartile range is zero.

Table 6. Mean detections per image across evaluation views

Configuration	VOC20	VOC6	Clipart20	Watercolor6
Faster R-CNN	26.1	10.4	24.8	10.4
SSD300	91.2	50.7	89.2	45.9
RetinaNet	99.8	35.4	99.7	34.1
FCOS	99.1	35.6	97.5	35.4
YOLOv8n	26.4	11.7	19.5	7.7
RT-DETR-l	100.0	46.0	100.0	51.5

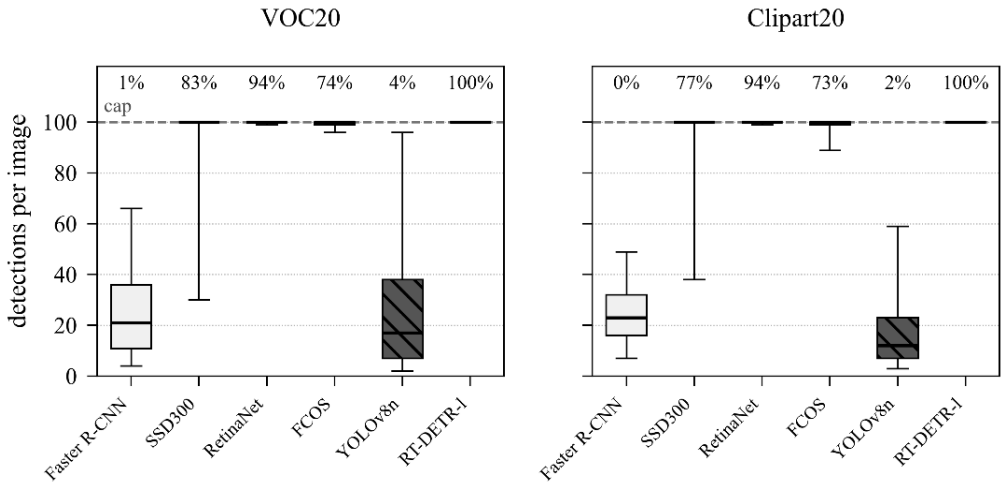

Figure 5. Per-image detection counts against the truncation limit (photo credit: original)

5. Conclusion

Six detector configurations were compared for cross-domain robustness under a common source training set, training budget, and evaluation protocol, with three random seeds each. In-domain average precision has limited predictive value for average precision on an unseen domain, the

DOI: 10.54254/2755-2721/2026.36995

Spearman coefficient being 0.257 for Clipart1k and 0.429 for Watercolor2k. The ranking changes under domain shift: SSD300 rises from fifth in-domain to first on both target domains, and the reordering differs between the two targets. Seed variability is larger on cross-domain views than in-domain for every configuration, by factors between 3.4 and 16.2, so differences of a few tenths of a point on a target domain from a single run cannot be treated as evidence of superiority.

Attribution requires care. Clipart1k and Watercolor2k contain 500 and 1000 images against 4952 in-domain, so finite-test-set sensitivity is a competing explanation for the larger observed seed variability that this design cannot rule out. The configurations differ in optimiser, learning rate, and schedule; although each derives from an official reference implementation through a rule declared in advance, the observed differences still contain an optimisation contribution. Warmup exposure differs by roughly a factor of twenty-five between the two frameworks, and the shared detection cap binds four of the six configurations. The fixed budget of 18 epochs is also shorter than several native schedules, so convergence may differ. The findings should accordingly be read as observations about specific configurations under a specific budget rather than as causal statements about architecture, and the domain shift examined is restricted to artistic style.

Future work will broaden the shift types by adding an evaluation set built from common image corruptions, add per-category average precision so that ranking preservation can be examined at class level, and vary the training budget to separate convergence effects from robustness effects.

References

[1] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems, 28*, 91–99.

[2] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single shot multibox detector. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016 (Lecture Notes in Computer Science, Vol. 9905*, pp. 21–37). Springer.

[3] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., & Chen, J. (2024). DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 16965–16974).

[4] Inoue, N., Furuta, R., Yamasaki, T., & Aizawa, K. (2018). Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5001–5009).

[5] Li, Y.-J., Dai, X., Ma, C.-Y., Liu, Y.-C., Chen, K., Wu, B., He, Z., Kitani, K., & Vajda, P. (2022). Cross-domain adaptive teacher for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 7581–7590).

[6] Danish, M. S., Khan, M. H., Munir, M. A., Sarfraz, M. S., & Ali, M. (2024). Improving single domain-generalized object detection: A focus on diversification and alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 17732–17742).

[7] Lee, W., Hong, D., Lim, H., & Myung, H. (2024). Object-aware domain generalization for object detection. *Proceedings of the AAAI Conference on Artificial Intelligence, 38*(4), 2947–2955.

[8] Vidit, V., Engilberge, M., & Salzmann, M. (2023). CLIP the gap: A single domain generalization approach for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3219–3229).

[9] Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2980–2988).

[10] Tian, Z., Shen, C., Chen, H., & He, T. (2019). FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (pp. 9627–9636).

[11] Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Machine Learning and Knowledge Extraction, 5*(4), 1680–1716.

[12] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338.

DOI: 10.54254/2755-2721/2026.36995

[13] Paniego, S., Sharma, V., & Cañas, J. M. (2022). Open source assessment of deep learning visual object detection. *Sensors, 22*(12), 4575.

[14] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., ... Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems, 32*, 8024–8035.