CPCNET: CROSS-DEPTH PYRAMID CONSTRUCTION FOR UNDERWATER OBJECT DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting pretrained vision transformers to underwater object detection requires constructing a multiscale pyramid from encoder features that share a common spatial grid. How features from different encoder depths are combined during this construction determines the representations supplied to the detector. We propose CPCNet, an underwater object detector built around a Cross-depth Pyramid Constructor (CPC). CPC jointly aggregates four projected DINOv3 states on their native grid to form a shared central representation. This center feeds a finer branch and a progressively coarser chain, with depth-specific bypasses providing direct encoder inputs to the branches. In single-run evaluations, CPCNet achieves 69.66, 65.27, and 52.18 AP on DUO, RUOD, and UTDAC, respectively. Across all three underwater datasets, CPC achieves higher AP than approximately parameter-matched independent routing and higher AP with fewer operations than source-complete parallel construction.

1 INTRODUCTION

As illustrated in Figure 1, underwater organisms often resemble the surrounding seabed in color and texture, while indistinct boundaries and overlapping targets further complicate foreground–background discrimination (Wille et al., 2026). Detection under these conditions requires representations that distinguish targets from their surroundings across object scales. Pretrained vision transformers such as DINOv3 provide a foundation for learning these representations (Siméoni et al., 2025). However, their encoder states share a common spatial grid, whereas a multiscale detection neck requires features at different resolutions. Adapting these states therefore involves constructing a detection pyramid as well as selecting its feature sources.

Most ViT-based detectors follow a ViTDet-style paradigm: they construct multiscale features from the final encoder feature map, so all scales share the same final representation (Li et al., 2022; Fang

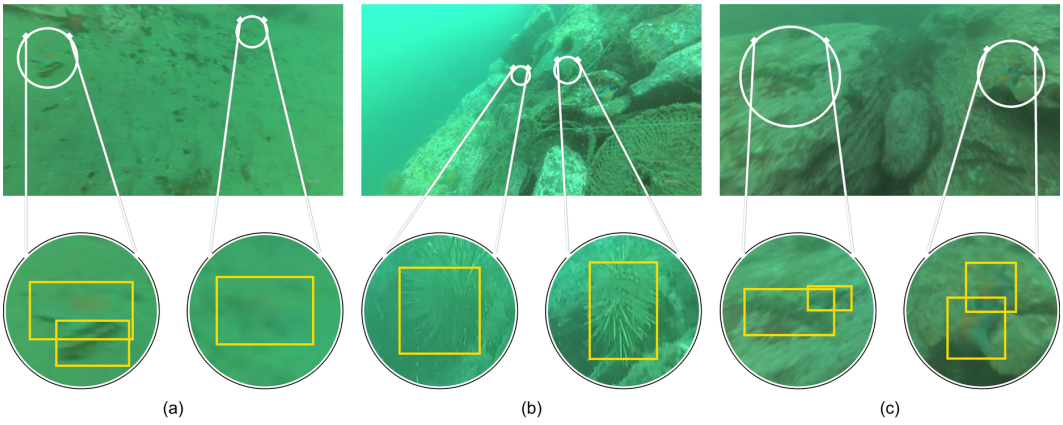

Figure 1: Visual challenges in underwater object detection. (a) Target–background color similarity (DUO), (b) structural and textural similarity (UTDAC), and (c) overlapping targets (RUOD).

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.

et al., 2023a). This approach demonstrates that a single final representation can support multiscale detection. Intermediate encoder states offer additional feature sources, but their availability alone does not determine how they should contribute to the pyramid. Whether they provide further benefits depends on how they are processed and combined during scale construction.

Existing studies explore multilevel information through hierarchical feature refinement (Zhao et al., 2025; Wang et al., 2026), intermediate-state resampling (Ali et al., 2021), and interactions between transformer and convolutional features (Chen et al., 2023; Xia et al., 2024; Huang et al., 2025). These approaches provide different ways to bring features from multiple depths into detection. For a constructor operating on completed, same-grid encoder states, two choices remain closely connected: which states each scale receives, and where those states interact. Assigning individual states to separate scales, mixing several states independently at each scale, and forming a shared representation before scale generation organize the same encoder sources differently. Their relative value for underwater detection calls for comparison within a common detection framework.

When multiple scales use the same encoder states, aggregating them before scale generation provides a shared representation for the branches. Supplementing this representation with direct depth-specific inputs allows each branch to access selected states without passing through central mixing. These connections combine shared information with direct encoder features while providing shorter paths for feature delivery and gradient propagation.

Following this design, we propose CPCNet, an underwater object detector built around a Cross-depth Pyramid Constructor (CPC). CPC projects four DINOv3 states to a common channel width and aggregates them on their native grid into a shared center. Stage propagation carries this representation into a finer branch and progressively coarser stages, while depth-specific bypasses supply direct encoder inputs to the branch mixers. We compare this construction with final-state, independently routed, and source-complete parallel alternatives, then use single-state and component controls to examine how source diversity, central mixing, and bypasses contribute to detection.

This paper makes the following contributions:

- We propose CPC, which combines shared cross-depth aggregation, stage propagation, and direct depth-specific inputs to construct a multiscale detection pyramid from same-grid encoder states.
- We compare encoder-source selection and construction paths within a common DINOv3-based detection framework, examining the contributions of independent refinement, multi-level mixing, and joint construction.
- We evaluate CPCNet on DUO, RUOD, and UTDAC and investigate its construction through retrained controls, branch-response analysis, and detection comparisons.

2 Related Work

Pyramid Construction from Final Encoder Features. The strong representations learned by pretrained vision transformers have encouraged their adaptation to object detection, which requires features that support predictions across object scales. ViTDet constructs a simple feature pyramid through parallel resampling of the final encoder representation (Li et al., 2022). EVA adopts this detection architecture for large-scale pretrained visual representations (Fang et al., 2023a). Both generate multiple resolutions from a single final-state source. CPC instead aggregates selected encoder states before scale generation and retains direct depth-specific inputs to the branches.

Multilevel Feature Adaptation. CIDNet and RHCNet refine hierarchical convolutional features through interference mining and hierarchical calibration, respectively (Zhao et al., 2025; Wang et al., 2026). Their backbone stages provide different resolutions, whereas plain-transformer states share a grid and require scale construction. XCiT resamples states from different encoder depths into separate scale paths before FPN fusion (Ali et al., 2021). DPT reassembles tokens from multiple ViT layers and progressively fuses them for dense prediction (Ranftl et al., 2021); its reassembly-and-fusion order differs from CPC's central aggregation followed by stage construction. ViT-Adapter and ViT-CoMer exchange information between transformer and convolutional features during encoding (Chen et al., 2023; Xia et al., 2024), while DEIMv2's Spatial Tuning Adapter combines multiple DINOv3

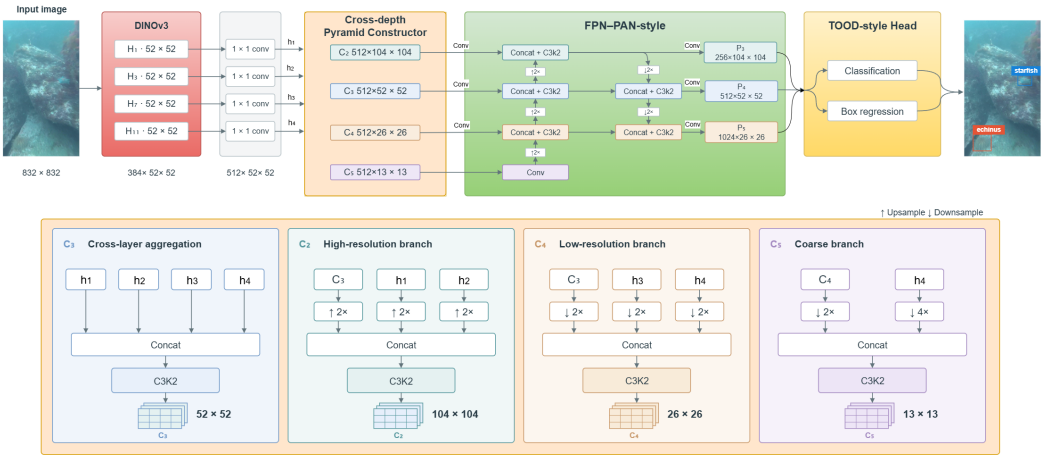

Figure 2: Overview of CPCNet. Four selected DINOv3 states feed CPC, FPN–PAN, and a TOOD-style head. Constructor and PAN downsampling share the same D weights and BN statistics.

block outputs with a lightweight convolutional image branch through Bi-Fusion (Huang et al., 2025). CPC operates after encoding, jointly aggregating selected states on their native grid before generating finer and progressively coarser branches.

Feature Propagation and Shortcuts. Spatial Residual introduces SkipCut connections for underwater detection, routing channel subsets through different processing paths and providing shortcuts for gradient propagation (Zhou et al., 2025). CPC's bypasses connect selected projected encoder states directly to branch mixers alongside propagated features from the shared center. These connections preserve direct feature access and provide shorter gradient paths to the selected states.

3 Method

3.1 Overall Architecture

CPCNet connects a DINOv3 backbone to CPC, an FPN–PAN neck, and a TOOD-style head (Figure 2). At input resolution 832×832, CPC converts four 52×52 encoder maps into stages $C_2, \ldots, C_5$ at strides $8, 16, 32, 64$. The neck produces the three detection levels at strides $8, 16, 32$. The encoder is fully fine-tuned without changing its block structure. Let $x \in \mathbb{R}^{B\times 3\times 832\times 832}$ be an input batch of B images; tensor dimensions include batch, channels, height, and width.

3.2 Shared Cross-Depth Aggregation

Let E_{ℓ_j} be the hidden state at zero-based block index $\ell_j \in \{1, 3, 7, 11\}$. Removing the class and register tokens and restoring the patch grid gives

$$H_{\ell_j} = \mathcal{R}\big(E_{\ell_j}(x)\big) \in \mathbb{R}^{B\times 384\times 52\times 52}, \qquad j = 1, \ldots, 4. \tag{1}$$

Independent lateral projections produce $h_j = L_j(H_{\ell_j}) \in \mathbb{R}^{B\times 512\times 52\times 52}$. Uppercase H_1, H_3, H_7, H_{11} identify raw states by zero-based encoder block; lowercase $h_1, \ldots, h_4$ enumerate their projected counterparts. Encoder depth denotes the selected block, rather than scene depth or a guaranteed ordering of semantic quality.

Since the projected states share a grid, they can interact before a branch changes resolution. Channel concatenation $[\cdot]$ followed by a nonlinear mixer produces the central stage:

$$C_3 = \Psi_3([h_1, h_2, h_3, h_4]). \tag{2}$$

The mixer maps 2,048 concatenated channels to a 512-channel center. Each scale branch therefore receives a learned combination of all four selected states rather than a separately assigned depth. Branch-specific inputs can also reach a mixer without first passing through this shared transformation.

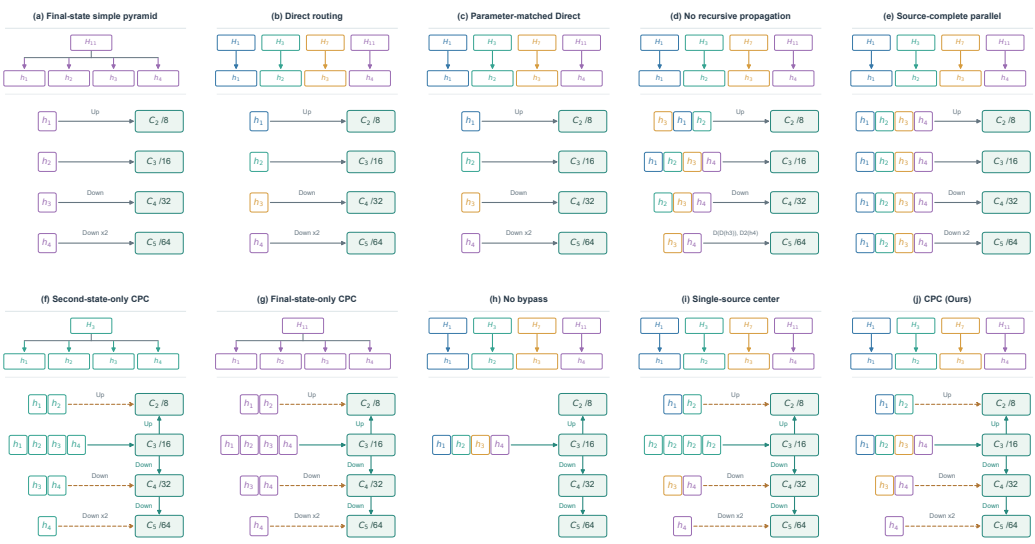

Figure 3: Construction variants and encoder-source coverage before FPN–PAN fusion.

These bypasses let the branch combine the common representation with selected depth-specific cues, providing additional inputs for distinguishing visually similar targets and backgrounds.

3.3 Scale Construction with Depth-Specific Bypasses

Let U denote nearest-neighbor $2\times$ upsampling and D a 3×3 stride-2 Conv–BN–SiLU block. All calls to D in the constructor and the two bottom-up PAN calls use the same convolution weights, BN parameters, and BN running statistics. An independent two-stage path Q, with separate weights, reduces h_4 by a factor of four. The remaining stages are constructed as

$$\begin{aligned} C_2 &= \Psi_2([U(C_3), U(h_1), U(h_2)]), \\ C_4 &= \Psi_4([D(C_3), D(h_3), D(h_4)]), \\ C_5 &= \Psi_5([D(C_4), Q(h_4)]). \end{aligned} \tag{3}$$

Each Ψ_i is a separate C3k2 mixer with expansion 0.5, two C3k blocks, and Conv–BN–SiLU processing (Appendix D). All constructed stages have 512 channels. The fine stage C_2 combines the upsampled center with h_1 and h_2. The coarse stage C_4 combines the downsampled center with h_3 and h_4, and C_5 combines the next downsampled stage with $Q(h_4)$. For an 832×832 input, C_2, C_3, C_4, C_5 have spatial sizes of $104, 52, 26, 13$ pixels per side, corresponding to strides $8, 16, 32, 64$, respectively.

The dependencies $C_3 \to C_2$, $C_3 \to C_4$, and $C_4 \to C_5$ carry the shared representation into each scale. The bypasses add designated projected states to these propagated inputs. Thus, every constructed stage has a path from all four projected taps before FPN fusion. Source-complete parallel construction provides the same coverage through independent stages, as compared in Figure 3. Reachability specifies available computational paths; the experiments measure the resulting configurations.

3.4 Multiscale Fusion and Detection

FPN–PAN applies top-down fusion followed by bottom-up aggregation to $C_2, \ldots, C_5$ (Lin et al., 2017; Liu et al., 2018). Output projections produce P_3, P_4, and P_5 at strides $8, 16, 32$ with $256, 512, 1024$ channels. At 832×832 input resolution, their spatial sizes are 104×104, 52×52, and 26×26. This fusion acts on already constructed resolutions, whereas CPC determines how encoder depths enter those resolutions. All constructor controls retain this downstream topology.

A TOOD-style head uses task-specific channel decomposition, classification alignment, and a 16-bin distribution for each box side (Feng et al., 2021; Li et al., 2020). The encoder, CPC, neck, and head

are trained jointly with

$$\mathcal{L} = 7.5\mathcal{L}_{\text{box}} + 0.5\mathcal{L}_{\text{cls}} + 1.5\mathcal{L}_{\text{DFL}}, \tag{4}$$

where the terms are the box, classification, and distribution-focal losses. These weights match the Ultralytics project defaults and the saved main-run training configuration; they are fixed across the DUO constructor comparisons. Inference uses NMS and no separate objectness output. Appendix D records the complete fusion path, operator sharing, and loss weights.

4 Experiments

4.1 Experimental Setup

Separate models are trained on DUO, RUOD, and UTDAC (Liu et al., 2021; Zhou et al., 2025; Wang et al., 2026) using 832×832 inputs, batch size 8, at most 200 epochs, and SGD with initial learning rate 0.01, momentum 0.937, and weight decay 5×10^{-4}. AMP is disabled. Each configuration contributes one checkpoint selected by its highest training-time validation AP. Final reporting uses this same split, so the results are validation-selected observations rather than estimates from an untouched test set. In particular, the RUOD configuration resolves its local validation entry to the annotated 'test' file listed in Appendix I; we do not interpret it as a blind official-test evaluation.

Re-evaluation uses rectangular preprocessing and pycocotools (Lin et al., 2014); the DUO, RUOD, and UTDAC exports use batch size 8, whereas the separately reported COCO export uses batch size 1. All primary AP columns use standard COCO maxDets$= 100$ with caps $[1, 10, 100]$, after retaining up to 1,000 post-NMS predictions. Object-area thresholds are 32^2 and 96^2 original-image pixels. Appendix F gives dataset splits and training details.

4.2 Main Detection Results

Table 1 compares CPCNet with published underwater detectors from 2022–2026 and earlier DUO/UTDAC references, including our server-side RHCNet reproduction. CPCNet reaches 52.18 AP on UTDAC compared with 49.50 for reproduced RHCNet, while the DUO comparison favors RHCNet by 0.24 points. Because these systems use different backbones, inputs, and evaluation protocols, the table provides context rather than a protocol-controlled ranking. The contribution of pyramid construction is examined next under a common detector framework.

RHCNet results use the best checkpoint selected by validation AP for each dataset, with the same UTDAC checkpoint used for the qualitative comparison in Appendix G.1. RHCNet retains its native aspect-ratio-preserving evaluation, and the other published values follow the sources in Appendix H. Architectures, pretraining, input sizes, and evaluation protocols differ across these systems. The comparison therefore describes complete detectors under their respective settings.

4.3 Construction Comparisons and Ablations

We compare encoder-source selection and construction paths on DUO using the configurations in Figure 3. All share the same DINOv3-S/16 pretrained source, output strides and channels, 512-channel FPN–PAN, TOOD-style head, and training recipe. Table 2 reports the results. Starting from final-state construction, we examine how intermediate states are processed and combined, then assess the contributions of source diversity and individual components within joint construction.

The final-state simple pyramid applies four independent projections to block 11 and constructs multiple resolutions before the common neck and head, following ViTDet's construction principle (Li et al., 2022). Direct routing instead assigns four distinct encoder states to separate scale paths:

$$C_2^{\text{dir}} = U(h_1), \quad C_3^{\text{dir}} = h_2, \quad C_4^{\text{dir}} = D(h_3), \quad C_5^{\text{dir}} = Q(h_4). \tag{5}$$

Both configurations have 52.56M parameters and require 322.07 GFLOPs. The final-state simple pyramid achieves 67.94 AP, slightly above Direct routing at 67.20 AP, showing that the final encoder representation remains a useful source for multiscale detection. Whether intermediate states can provide additional benefits through refinement or interaction remains open.

We first examine refinement within each independent path. Parameter-matched Direct adds a separate nonlinear refiner at each routed level, reaching 68.46 AP, an increase of 1.26 points over Direct

Table 1: Comparison with published and reproduced underwater detectors. A dash denotes an unreported value.

Method	Venue/year	Backbone	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
DUO									
GFL	NeurIPS'20	R-50	32.20	78.52	58.66	79.33	46.54	61.63	55.62
YOLOv7	CVPR'23	CSPDarknet53	37.60	64.81	62.35	83.54	46.64	63.79	61.70
GCC-Net	PR'24	Swin-T	36.74	300.79	69.10	**87.80**	–	–	–
CIDNet	KBS'25	R-50	82.50	324.48	68.83	86.56	**56.52**	70.63	67.34
RHCNet	CVPR'26	R-50	70.41	494.12	**69.90**	86.30	54.40	**71.70**	**68.40**
CPCNet (Ours)	–	DINOv3-S/16	63.59	409.66	69.66	87.44	56.40	71.48	68.35
RUOD									
AMSP-UOD	AAAI'24	AMSP-VConv + YOLO	–	–	65.20	86.10	–	–	–
BSR5-YOLO(E-FPN)-L	TPAMI'25	Spatial-residual CNN	36.97	75.37	65.10	87.30	20.60	**50.30**	71.10
CPCNet (Ours)	–	DINOv3-S/16	63.60	409.70	**65.27**	**87.50**	**21.86**	50.27	**71.24**
UTDAC									
GFL	NeurIPS'20	R-50	32.20	78.52	46.80	72.42	20.45	42.22	51.61
YOLOv7	CVPR'23	CSPDarknet53	37.60	64.81	47.82	81.22	21.27	43.61	52.14
CIDNet	KBS'25	R-50	82.50	324.48	49.57	85.38	23.11	45.74	56.08
RHCNet	CVPR'26	R-50	70.41	494.12	49.50	85.50	24.90	44.50	55.40
CPCNet (Ours)	–	DINOv3-S/16	63.59	409.66	**52.18**	**86.18**	**26.76**	**48.23**	**56.95**

Table 2: Construction variants and ablations on DUO with DINOv3-S/16.

Configuration	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
(a) Final-state simple pyramid	52.56	322.07	67.94	86.34	53.58	70.20	66.16
(b) Direct routing	52.56	322.07	67.20	86.66	52.16	68.63	66.46
(c) Parameter-matched Direct	63.57	402.26	68.46	86.29	53.49	70.31	66.95
(d) No recursive propagation	63.59	409.66	66.97	86.99	52.41	68.79	66.02
(e) Source-complete parallel	63.58	423.93	69.34	86.61	56.32	71.14	68.14
(f) Second-state-only CPC	63.59	409.66	68.86	86.52	**56.51**	71.10	66.97
(g) Final-state-only CPC	63.59	409.66	68.82	86.26	53.65	70.79	67.32
(h) No bypass	62.28	387.15	66.66	86.76	52.47	68.18	66.14
(i) Single-source center	63.59	409.66	66.55	86.36	47.49	68.05	66.34
(j) CPC (Ours)	63.59	409.66	**69.66**	**87.44**	56.40	**71.48**	**68.35**

routing. This improvement accompanies increased capacity, while each scale still receives only one projected encoder state before FPN fusion. Independent refinement thus improves the use of individual states, leaving open whether combining multiple states within each scale can provide further gains.

The configurations in Figure 3(d,e) introduce such combinations without propagating constructed stages. No recursive propagation mixes subsets of the four selected states and achieves 66.97 AP. Source-complete parallel construction supplies all four states to every stage through independent resampling and mixing paths, reaching 69.34 AP. These results support providing each stage with all four selected states rather than only a subset, although mixer widths and resampling paths also differ.

With all four states available to every stage, we next examine how their fusion is organized across scales. Source-complete parallel construction learns a separate mixture of the same four projected states at each stage. An alternative is to aggregate these states before scale generation, allowing the resulting cross-depth representation to be reused as the stages are constructed. CPC adopts this organization through a shared center and stage propagation. Its depth-specific bypasses additionally supply selected encoder states directly to the branches, allowing each branch to combine the propagated representation with inputs that have not passed through central mixing (Figure 3(j)). These direct connections form shortcuts from the selected encoder states to the branch mixers, providing shorter paths for both feature delivery and gradient propagation. CPC achieves 69.66 AP, exceeding source-

complete parallel construction by 0.32 points and Parameter-matched Direct by 1.20 points, with higher AP than the latter in all three object-area ranges. Parameter-matched Direct has $63{,}574{,}543$ parameters, only $18{,}960$ (0.030%) fewer than CPCNet. Joint construction therefore improves on independent single-source refinement at nearly equal parameter counts.

Within this construction, we next examine whether combining distinct encoder depths contributes to its performance. Second-state-only CPC and Final-state-only CPC obtain all four projected inputs from block 3 and block 11, respectively, while retaining CPC's topology, parameter count, and operation count. They achieve 68.86 and 68.82 AP, compared with 69.66 for CPC. The final-state-only result reinforces the observation in (a) that a single final representation can support multiscale detection, while the comparison within CPC shows an additional benefit from combining distinct states. This benefit varies across object sizes: Second-state-only CPC has slightly higher small-object AP, at 56.51 versus 56.40. These source comparisons lead to the central mixing and bypass paths through which distinct states enter the branches.

Two retrained variants examine how these states enter the construction. Single-source center repeats h_2 in the retained center mixer while preserving propagation and bypasses, reaching 66.55 AP. Its comparison with CPC supports a four-depth center over this single-source center at the same nominal parameter count. No bypass retains the four-depth center and propagation but removes direct branch inputs, reaching 66.66 AP with 62.28M parameters and 387.15 GFLOPs. This comparison favors the full construction, while also changing mixer input widths and shared-operator calls (Appendix E). Spatial Residual motivates shortcuts for gradient propagation (Zhou et al., 2025); CPC provides shorter structural paths, but their optimization effect is not isolated here. We next examine the branch responses and detections learned by the two bypass configurations.

4.4 Analysis of Depth-Specific Bypasses

The AP difference motivates examining branch features alongside final detections. We fit separate linear probes on frozen C_2, C_4, and C_5 features from the two models using the same training samples. On held-out images, probe AUROC improves at C_4 but decreases at C_2 and C_5 (Appendix G.2). Thus, higher detector AP does not correspond to uniformly better branch separability. Figure 4 shows selected local response differences; the common display scale does not calibrate probabilities across independently fitted probes.

In the selected C_2 and C_5 examples, CPC detects targets missed by No bypass, while the C_4 example has less background probe response around a starfish detected by both models. These positive examples illustrate local behavior rather than an average gain at every branch. The full-validation high-IoU detection-recall curve provides a complementary detector-level comparison: at confidence 0.50 and IoU 0.90, CPC matches 59.91% of annotations versus 45.63% for No bypass. More targets therefore survive the joint confidence, category, and overlap requirements. This result reflects the complete detection pipeline, not box regression alone.

4.5 Results on RUOD, UTDAC, and COCO

Following the DUO analysis, we extend the construction comparison to RUOD and UTDAC, training each configuration separately within the same detection framework. Table 3 shows that CPC leads Parameter-matched Direct and source-complete parallel construction in AP and all reported object-area ranges on both datasets. Its margins over parallel construction remain below half an AP point, while the ordering between Direct and parallel changes on RUOD. The benefit of joint construction thus extends across these benchmarks, with margins that depend on the dataset.

Beyond underwater imagery, CPCNet trained separately on COCO2017 obtains 51.86 AP, 69.14 AP_{50}, 34.59 AP_S, 57.77 AP_M, and 67.83 AP_L with 63.65M parameters and 410.21 GFLOPs under the same evaluation implementation. This in-domain result shows that the construction also supports detection on terrestrial imagery. Appendix A compares CPCNet with major-conference detectors; their different backbones, training schedules, and evaluation settings make this a contextual comparison. We next examine the computational cost of the evaluated construction variants.

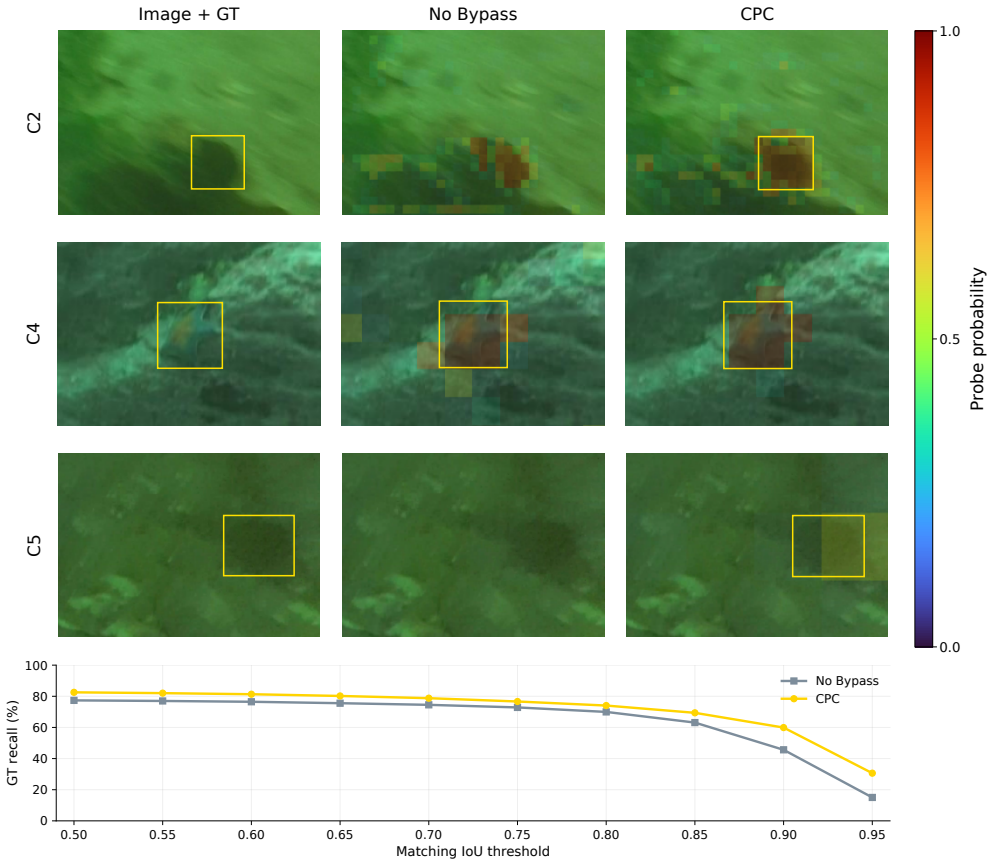

Figure 4: Selected branch-probe responses and full-validation high-IoU detection recall on DUO. Overlays use separately fitted probe probabilities; boxes are full-detector predictions. Recall covers all 1,111 validation images at confidence 0.50.

Table 3: Construction comparisons on RUOD and UTDAC with DINOv3-S/16.

Configuration	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
RUOD							
Parameter-matched Direct	63.58	402.31	65.08	87.33	19.27	50.21	70.74
Source-complete parallel	63.59	423.97	64.88	87.05	20.65	49.67	70.64
CPC	63.60	409.70	**65.27**	**87.50**	**21.86**	**50.27**	**71.24**
UTDAC							
Parameter-matched Direct	63.57	402.26	46.65	80.05	19.55	42.69	51.28
Source-complete parallel	63.58	423.93	51.74	85.47	26.11	48.06	56.44
CPC	63.59	409.66	**52.18**	**86.18**	**26.76**	**48.23**	**56.95**

4.6 Computational Cost

Controlled comparisons show that CPC balances detection accuracy and computation. On DUO, it outperforms source-complete parallel construction with 409.66 GFLOPs versus 423.93 and gains 1.20 AP over Parameter-matched Direct with 1.84% more computation. At 832×832, CPC has 11.03M more parameters and 87.59 more GFLOPs than the evaluated final-state simple pyramid. This is a measured whole-detector difference; stacked variants and inference latency were not measured.

The same cost ordering holds on RUOD and UTDAC, supporting joint construction within the shared detection framework. Comparisons with complete detectors additionally depend on input resolution and architecture; Appendix F provides the profiling details. We next examine how these detection differences appear in individual underwater scenes.

4.7 Qualitative Results and Failure Analysis

Figure 5 compares all ten evaluated configurations on three DUO scenes, showing all predictions with confidence ≥ 0.50. Simple Pyramid and Parameter-matched Direct miss the upper starfish, right-hand sea cucumber, and blurred urchin in the respective scenes; both also produce false positives. Source-complete parallel recovers all seven annotations in the second scene but adds a sea-cucumber prediction, and it misses targets in the first and third scenes. CPCNet matches all annotations without additional predictions in these selected examples, illustrating the detection differences associated with the quantitative comparison. Appendix G specifies case selection and matching, and Appendix G.1 provides the UTDAC comparison with reproduced RHCNet.

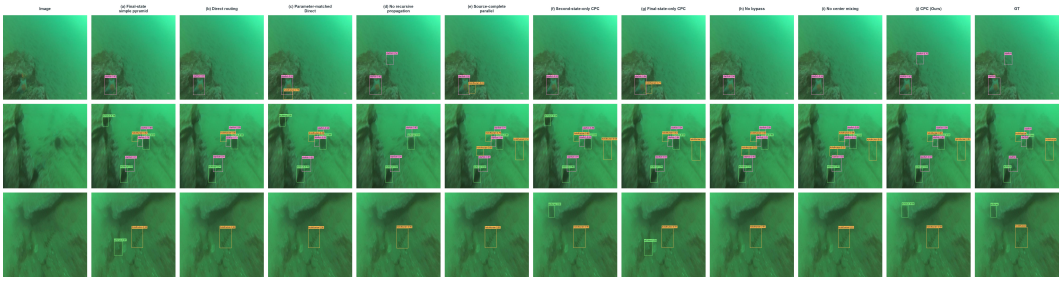

Figure 5: Selected detection successes on the DUO validation set. Columns show the image, configurations (a)–(j) from Figure 3, and ground truth (GT).

Beyond these successful cases, CPCNet remains sensitive to limited target resolution and degraded visual cues. At confidence and matching IoU thresholds of 0.50, recall is 33.69% for targets with input short sides below 16 pixels, versus 93.26% for those measuring at least 64 pixels. The controlled degradations reduce AP by 30.67 points under Gaussian blur with an input-equivalent standard deviation of 3 pixels and by 19.36 points at 30% contrast retention (Appendix G.3). At 30% brightness retention, AP decreases by only 3.37 points, substantially less than under contrast reduction at the same retention factor. Under these tested conditions, reduced contrast is more damaging than uniform dimming, while blur produces the largest decline. Together with the low recall for very small targets, these results point to difficulty retaining detection when object structure becomes poorly resolved or less distinguishable from its surroundings. Lowering the confidence threshold recovers more targets but increases unmatched detections, leaving a trade-off between missed targets and false positives.

5 Conclusion

We presented CPCNet, which adapts same-grid DINOv3 features to underwater object detection through joint cross-depth pyramid construction. Its shared center, stage propagation, and depth-specific bypasses combine information across encoder depths while preserving direct inputs to individual scales. Across DUO, RUOD, and UTDAC, this organization improves detection accuracy over Parameter-matched Direct and requires fewer operations than source-complete parallel construction, providing a useful accuracy–computation trade-off among the evaluated alternatives. The DUO controls support the complete CPC topology in the tested comparisons; because the controls also change widths, operator calls, and gradient routes, they do not isolate individual component effects. These gains, however, do not eliminate difficulties with very small targets or sensitivity to blur and reduced contrast, as shown by the failure analysis and controlled degradation tests. Future work will investigate how to preserve fine spatial detail for small targets and improve detection when visual cues are degraded.

AI USE STATEMENT

Generative AI tools assisted with literature retrieval and discovery, manuscript restructuring, English-language editing, LaTeX drafting, programmatic figure preparation, experimental-design feedback, evaluation-code assistance, and interpretation of recorded results. Experimental measurements were obtained by executing model evaluations and analysis code.

ETHICS STATEMENT

This work uses existing underwater object-detection benchmarks and does not involve human subjects or personal data. Dataset access and redistribution remain subject to the licenses of DUO, RUOD, and UTDAC.

REPRODUCIBILITY STATEMENT

Section 3 defines the constructor. Appendix D records downstream fusion and operator sharing; Appendix E specifies the controls; and Appendix F records splits and training. Appendix G specifies detection preprocessing, confidence filtering, matching, and sample selection. Code, checkpoint hashes, and visualization manifests will be released subject to benchmark licenses.

REFERENCES

Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Herve Jegou. XCiT: Cross-covariance image transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20014–20027, 2021.

Long Chen, Wei Miao, Xin Gao, Yunzhi Zhuge, Hongming Xu, Yaxin Li, and Qi Xu. Spatial-frequency spiking neural network for underwater object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20217–20225, 2026.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *International Conference on Learning Representations*, 2023.

Linhui Dai, Hong Liu, Pinhao Song, and Mengyuan Liu. A gated cross-domain collaborative network for underwater object detection. *Pattern Recognition*, 149:110222, 2024.

Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19358–19369, 2023a. doi: 10.1109/CVPR52729.2023.01855.

Yuxin Fang, Shusheng Yang, Shijie Wang, Yixiao Ge, Ying Shan, and Xinggang Wang. Unleashing vanilla vision transformer with masked image modeling for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6244–6253, 2023b.

Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R. Scott, and Weilin Huang. TOOD: Task-aligned one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3490–3499, 2021.

Shihua Huang, Yongjie Hou, Longfei Liu, Xuanlong Yu, and Xi Shen. Real-time object detection meets DINOv3. *arXiv preprint arXiv:2509.20787*, 2025. URL https://arxiv.org/abs/2509.20787.

Xiang Li, Wenhai Wang, Lijun Wu, Shuo Chen, Xiaolin Hu, Jun Li, Jinhui Tang, and Jian Yang. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21002–21012, 2020.

Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. In *European Conference on Computer Vision*, pp. 280–296, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pp. 740–755, 2014.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

Chongwei Liu, Haojie Li, Shuchang Wang, Ming Zhu, Dong Wang, Xin Fan, and Zhihui Wang. A dataset and benchmark of underwater object detection for robot picking. In *IEEE International Conference on Multimedia and Expo Workshops*, pp. 1–6, 2021.

Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

Yansong Peng, Hebei Li, Peixi Wu, Yueyi Zhang, Xiaoyan Sun, and Feng Wu. D-FINE: Redefine regression task in DETRs as fine-grained distribution refinement. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.13842.

Siyuan Qiao, Liang-Chieh Chen, and Alan Yuille. DetectoRS: Detecting objects with recursive feature pyramid and switchable atrous convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10213–10224, 2021. URL https://arxiv.org/abs/2006.02334.

René Ranftl, Aleksei Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 12179–12188, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Ranftl_Vision_Transformers_for_Dense_Prediction_ICCV_2021_paper.html.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. DINOv3. *arXiv preprint arXiv:2508.10104*, 2025. URL https://arxiv.org/abs/2508.10104.

Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790, 2020. URL https://arxiv.org/abs/1911.09070.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. FCOS: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9627–9636, 2019.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and Guiguang Ding. YOLOv10: Real-time end-to-end object detection. In *Advances in Neural Information Processing Systems*, volume 37, pp. 107984–108011, 2024a.

Bowen Wang, Zhi Wang, Wenhui Guo, and Yanjiang Wang. A dual-branch joint learning network for underwater object detection. *Knowledge-Based Systems*, 293:111672, 2024b.

Yueying Wang, Yiteng Guo, Weidong Zhang, Jie Wen, Liquan Shen, Huaicheng Yan, and Xin Xu. RHCNet: Residual-guided hierarchical calibration network for robust underwater object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4393–4402, 2026.

Melanie Wille, Tobias Fischer, and Scarlett Raine. Are all marine species created equal? Performance disparities in underwater object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 4556–4565, 2026.

Chunlong Xia, Xinliang Wang, Feng Lv, Xin Hao, and Yifeng Shi. ViT-CoMer: Vision transformer with convolutional multi-scale feature interaction for dense predictions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5493–5502, 2024.

Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412, 2018. URL https://arxiv.org/abs/1707.06484.

Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M. Ni, and Heung-Yeung Shum. DINO: DETR with improved DeNoising anchor boxes for end-to-end object detection. In *International Conference on Learning Representations*, 2023a.

Shilong Zhang, Xinjiang Wang, Jiaqi Wang, Jiangmiao Pang, Chengqi Lyu, Wenwei Zhang, Ping Luo, and Kai Chen. Dense distinct query for end-to-end object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7329–7338, 2023b.

Gaoli Zhao, Kefei Zhang, Liangzhi Wang, Wenyi Zhao, and Weidong Zhang. CIDNet: Cross-scale interference mining detection network for underwater object detection. *Knowledge-Based Systems*, 324:113902, 2025.

Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. DETRs beat YOLOs on real-time object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16965–16974, 2024.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6881–6890, 2021. URL https://arxiv.org/abs/2012.15840.

Jingchun Zhou, Zongxin He, Kin-Man Lam, Yudong Wang, Weishi Zhang, Chunle Guo, and Chongyi Li. AMSP-UOD: When vortex convolution and stochastic perturbation meet underwater object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7659–7667, 2024.

Jingchun Zhou, Zongxin He, Dehuan Zhang, Siyuan Liu, Xianping Fu, and Xuelong Li. Spatial residual for underwater object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4996–5013, 2025.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021.

A COCO2017 Detection Results

CPCNet is trained on 118,287 COCO2017 images using the main SGD recipe (seed 0, batch size 8, 200 epochs). The resumed run selects epoch 179 by training-time AP on the 5,000-image val2017 split using YOLO-format labels. The checkpoint is re-evaluated at size 832 with rectangular preprocessing, batch size 1, FP32, confidence floor 0.001, NMS IoU 0.70, and up to 1,000 detections before COCO evaluation with maxDets= $[1, 10, 100]$. Official annotations yield 51.86 AP, 69.14 AP_{50}, and $34.59/57.77/67.83$ $AP_{S/M/L}$; the model has 63.65M parameters and 410.21 GFLOPs. Image and category IDs are matched by filename and category name, retaining official instance areas and crowd flags.

Table 4 compares major-conference detectors from 2019–2025 under their published settings. FCOS improved 1x and Deformable DETR use author-released implementations that differ from the original paper configurations. FCOS uses 5,000-image minival; the others use val2017. Input sizes, pretraining, schedules, and evaluation implementations vary.

B Encoder Size and Pretraining Source

Table 5 compares encoder sizes and pretraining sources within CPC. DINOv3-S/16 has the highest AP, L/16 the highest AP_{50}, and B/16 the highest AP_L. These runs have different optimization histories, so they do not isolate encoder size under a common schedule.

Table 4: COCO2017 detection results for CPCNet and reference detectors.

Detector	Venue	Backbone	AP	Configuration and source
FCOS-imprv	ICCV'19	ResNet-50	38.7	Author model zoo; improved 1x, 90k steps, $800/1333$ (Tian et al., 2019)
Deformable DETR	ICLR'21	ResNet-50	44.5	Author model zoo; 4 scales, 50 epochs, $800/1333$ (Zhu et al., 2021)
DINO	ICLR'23	ResNet-50	49.0	Table 1; 4 scales, 12 epochs, $800/1333$ (Zhang et al., 2023a)
DDQ-DETR	CVPR'23	ResNet-50	52.1	Author model zoo; 5 scales, 12 epochs, $800/1333$ (Zhang et al., 2023b)
RT-DETR	CVPR'24	ResNet-50	<u>53.1</u>	Table 1; 72 epochs, 640^2 (Zhao et al., 2024)
YOLOv10-B	NeurIPS'24	CSP-based	52.5	Author model zoo; B variant, 640^2 (Wang et al., 2024a)
D-FINE-L	ICLR'25	HGNetv2-B4	**54.0**	Author model zoo; L variant, 80 epochs, 640^2 (Peng et al., 2025)
CPCNet (Ours)	–	DINOv3-S/16	51.86	This work; 200 epochs, 832 rectangular evaluation

The resumed L/16 retry uses initial learning rate 0.001, versus 0.01 for S/16 and B/16. The supervised ViT-S/16 run also resumes training, initialized from timm's ImageNet-1K model (vit_small_patch16_224.augreg_in1k). It retains CPC, the 512-channel FPN–PAN, and the TOOD-style head, reaching 65.44 AP. This comparison changes both pretrained source and encoder implementation; saved configurations do not recover every pre-resume optimization step.

Table 5: Encoder-size and pretraining-source comparisons on DUO.

Encoder	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑	Params ↓ (M)	GFLOPs ↓
DINOv3-S/16	**69.66**	<u>87.44</u>	**56.40**	**71.48**	<u>68.35</u>	63.59	409.66
DINOv3-B/16	<u>68.98</u>	87.16	47.31	<u>70.31</u>	**68.39**	128.74	760.76
DINOv3-L/16 (retry)	68.06	**88.06**	43.77	69.02	68.05	346.83	1941.08
Supervised ViT-S/16	65.44	84.41	<u>48.00</u>	66.86	64.83	63.66	525.89

C Related Feature Aggregation Architectures

Table 6 compares where and how features are aggregated. Cross-depth interaction denotes explicit aggregation of selected encoder outputs, each of which already incorporates preceding blocks. CPC combines this aggregation with spatial resampling, stage propagation, and depth-specific bypasses.

Table 6: Architectural comparison of feature aggregation and scale construction methods.

Design	Established operation	Distinction from the present constructor
FPN / PANet / BiFPN	Exchange among existing multiresolution stages (Lin et al., 2017; Liu et al., 2018; Tan et al., 2020).	Our constructor generates multiresolution stages from same-grid taps before FPN–PAN fusion.
DLA	Nonlinear hierarchical and iterative aggregation across layers and resolutions (Yu et al., 2018).	CPC aggregates completed same-grid encoder outputs after the backbone, before spatial resampling.

Table 6 (continued)

Design	Established operation	Distinction from the present constructor
XCiT (Ali et al., 2021)	Per-depth resampling of same-grid states, followed by an independent FPN.	XCiT assigns encoder states to separate scale paths; CPC mixes the selected states before generating scale branches.
DPT (Ranftl et al., 2021)	Reassembly of tokens from multiple ViT layers followed by progressive feature fusion for dense prediction.	DPT progressively fuses reassembled features; CPC forms a shared same-grid center and then propagates constructed stages with direct depth-specific inputs.
DEIMv2 STA (Huang et al., 2025)	Resamples multiple ViT block outputs and fuses them with a lightweight convolutional image branch.	CPC mixes projected states before scale generation and uses stage propagation with depth-specific inputs, without an extra image branch.
MIMDet (Fang et al., 2023b)	Convolutional fine-scale features and a ViT-derived coarse representation feed FPN.	Its fine scales originate before ViT patch-grid processing; ours are constructed from same-grid encoder states.
SETR-MLA	Same-grid cross-tap aggregation and upsampling for segmentation (Zheng et al., 2021).	SETR-MLA aggregates same-grid outputs for segmentation; CPC propagates constructed stages to multiple detection strides.
ViTDet	Parallel resampling of the final transformer map (Li et al., 2022).	The CPC topology additionally uses intermediate taps and dependencies among constructed stages.
DINOv3 detector	Four-tap concatenation, global/window feature combination, and transformer detection (Siméoni et al., 2025).	We reuse a multi-depth source but use a central-stage recursive adapter for a dense multilevel head.
DetectoRS RFP	Feedback from the FPN into backbone stages, unrolled over repeated passes (Qiao et al., 2021).	Stage propagation in the CPC topology occurs after one encoder pass, with no feature feedback into the backbone.
ViT-Adapter / ViT-CoMer	Encoder-interleaved convolution–transformer interaction produces multiscale outputs for an independent FPN in detection (Chen et al., 2023; Xia et al., 2024).	CPC consumes completed encoder taps without an additional image-derived branch or interaction inside the encoder.

D Implementation Details

At 832×832, the four encoder maps have shape $B \times 384 \times 52 \times 52$. CPC produces stages of spatial sizes 104, 52, 26, and 13; detector outputs are $B \times 256 \times 104 \times 104$, $B \times 512 \times 52 \times 52$, and $B \times 1024 \times 26 \times 26$, at strides $[8, 16, 32]$. Table 7 lists the remaining configuration.

Table 7: Configuration of CPCNet with DINOv3-S/16.

Field	Value
Initialization	DINOv3 pretrained
Parameters (DUO)	63,593,503
Hidden width / blocks / heads	$384/12/6$
Patch size / register tokens	$16/4$
Selected zero-based blocks	$1, 3, 7, 11$
Lateral / fusion width	$384 \rightarrow 512$
Constructed-stage channels	512
Output channels	$256, 512, 1024$
`end2end` / `reg_max`	`False` / 16
Loss weights (box / cls / DFL)	$7.5/0.5/1.5$
DUO classes / prediction locations	$4/14{,}196$

Operators and Sharing Rules. Each mixer Ψ_i is an independent C3k2 block with two repeats and C3k enabled. A 1×1 projection forms two channel groups; one traverses two C3k blocks. The initial groups and block outputs are concatenated and projected by a second 1×1 convolution. Each C3k has a split branch and two residual bottlenecks, each containing two 3×3 convolutions; convolutions use BN and SiLU. Lateral projections are independent 1×1 Conv–BN–SiLU blocks. The stride-2

3×3 Conv–BN–SiLU operator D shares weights, BN parameters, and running statistics across constructor downsampling and both PAN calls. The two stride-2 blocks in Q are independent of each other and D. FPN input projections and FPN/PAN fusion blocks also have independent weights.

FPN–PAN Fusion. With lateral projections η_i and nonlinear fusion blocks Γ_i and Π_i, the FPN–PAN path is

$$\begin{aligned} T_5 &= \eta_5(C_5), & T_4 &= \Gamma_4([U(T_5), \eta_4(C_4)]), \\ T_3 &= \Gamma_3([U(T_4), \eta_3(C_3)]), & T_2 &= \Gamma_2([U(T_3), \eta_2(C_2)]), \\ V_4 &= \Pi_4([D(T_2), T_3]), & V_5 &= \Pi_5([D(V_4), T_4]). \end{aligned} \tag{6}$$

Here U uses nearest-neighbor interpolation to the target stage's spatial shape ($2\times$ for the square inputs above). Output projections give $P_3 = \rho_3(T_2)$, $P_4 = \rho_4(V_4)$, and $P_5 = \rho_5(V_5)$; training uses Equation 4.

E Construction Variants and Training Details

Final-State Simple Pyramid. Four independent lateral projections process block 11 before the resampling in Equation 5. FPN–PAN and output projections are retained, without parameter-matching refiners. This matches the parameter count of multi-depth direct routing and preserves task-loss gradients through the full encoder.

Routing, Propagation, and Encoder-Source Controls. Parameter-matched Direct adds independent C3k2 refiners to the four outputs of Equation 5. At fusion width 512, hidden widths $291, 307, 291, 274$ for C_2, C_3, C_4, C_5 closely match CPCNet's parameter budget while keeping every refiner active.

The nonrecursive implementation uses the same projected states $h_1, \ldots, h_4$ but replaces stage propagation as follows:

$$\begin{aligned} C_3^{\mathrm{nr}} &= \Psi_3([h_1, h_2, h_3, h_4]), \\ C_2^{\mathrm{nr}} &= \Psi_2([U(h_3), U(h_1), U(h_2)]), \\ C_4^{\mathrm{nr}} &= \Psi_4([D(h_2), D(h_3), D(h_4)]), \\ C_5^{\mathrm{nr}} &= \Psi_5([D(D(h_3)), Q(h_4)]). \end{aligned} \tag{7}$$

Equation 7 preserves modules and nominal model size but changes both stage dependencies and source access: C_2^{nr} lacks the pre-FPN path from h_4 available through CPC's mixed center. It therefore does not isolate propagation. The implementation caches $D(h_2)$, $D(h_3)$, and $D(h_4)$, reusing the cached $D(h_3)$ for $D(D(h_3))$. Both this control and CPC call D four times in construction and twice in PAN.

Second-state-only CPC extracts all four maps but passes only H_3 (zero-based block 3) through the four independent lateral projections. Subsequent blocks receive no detection task-loss gradient. Final-state-only CPC instead uses block 11, retaining gradients through the full encoder.

Source-Complete Parallel Construction. With $[f(h_j)]_{j=1}^4$ denoting channel concatenation, this control computes

$$\begin{aligned} C_2^{\mathrm{sc}} &= \widetilde{\Psi}_2([U(h_j)]_{j=1}^4), & C_3^{\mathrm{sc}} &= \widetilde{\Psi}_3([h_j]_{j=1}^4), \\ C_4^{\mathrm{sc}} &= \widetilde{\Psi}_4([D(h_j)]_{j=1}^4), & C_5^{\mathrm{sc}} &= \widetilde{\Psi}_5([Q(h_j)]_{j=1}^4). \end{aligned} \tag{8}$$

Every stage receives all four taps without constructed-stage propagation. Mixers output 512 channels with hidden widths $241, 256, 241, 225$ at C_2, C_3, C_4, C_5, giving 11,752 fewer parameters and 14.27 more GFLOPs than CPCNet. Shared D processes four sources and is called twice in PAN; the two-step Q shares weights and BN statistics across all four sources.

CPC's 0.32-point advantage accompanies differences in mixer widths, resampling paths, operator calls, and gradient routes; it does not isolate propagation.

Single-source center and No bypass. Single-source center retains all four lateral projections and the center mixer, but changes the center input to

$$C_3 = \Psi_3([h_2, h_2, h_2, h_2]). \tag{9}$$

The remaining stages follow Equation 3. Repeating h_2 preserves nominal parameters and operations but supplies only one source to the center.

No bypass instead constructs

$$\begin{aligned} C_3 &= \Psi_3([h_1, h_2, h_3, h_4]), & C_2 &= \widetilde{\Psi}_2(U(C_3)), \\ C_4 &= \widetilde{\Psi}_4(D(C_3)), & C_5 &= \widetilde{\Psi}_5(D(C_4)). \end{aligned} \tag{10}$$

All mixers output 512 channels with hidden width 256. Branch input widths change from $1536, 1536, 1024$ to $512, 512, 512$, while the center retains 2048 channels. Construction calls D twice rather than four times; both models add two PAN calls, so shared BN receives different training inputs. The unused Q remains counted but receives no task-loss gradient; CPC and Single-source center each call it once. Thus No bypass changes direct inputs, mixer capacity, computation, and normalization updates together.

Training Records and Checkpoint Selection. Final-state-only CPC and DUO source-complete parallel use seed 0, a 200-epoch budget, and patience 100. Their runs end at epochs 186/178, with validation optima at 86/78. Under the same schedule, UTDAC parallel ends at 189 (best 89); UTDAC Direct ends at 200 (best 109). RUOD Direct ends at 200 (best 111), and RUOD parallel uses seed 0 and ends at 196 (best 96), matching its saved metrics. Best checkpoints are re-evaluated with maxDets$= [1, 10, 100]$.

On RUOD, CPC, Parameter-matched Direct, and parallel obtain $65.27/65.08/64.88$ AP using batch size 8, rectangular preprocessing, FP32, size 832, confidence floor 0.001, and NMS IoU 0.70. CPC and Direct were re-inferred; parallel's export from the same protocol was rescored against the same annotations, reproducing all six AP metrics.

No bypass and Single-source center use seed 0, training batch size 8, and a 200-epoch budget. Logs end at epochs 195/178, with highest logged validation AP at 72/78; No bypass records resumed training. Both best checkpoints are re-evaluated on 1,111 DUO images with rectangular preprocessing, FP32, batch size one, NMS IoU 0.70, and up to 1,000 detections before COCO evaluation with maxDets$= 100$.

F Dataset Splits, Training, and Profiling

Table 8 lists the training and validation splits. The main training command below uses DUO; RUOD and UTDAC substitute their data configurations and class counts.

Table 8: Training and validation splits used in this study.

Dataset	Classes	Train	Validation
DUO	4	6,671	1,111
RUOD	10	9,800	4,200
UTDAC	4	5,168	1,293

```
yolo detect train \
  model=ultralytics/cfg/models/26/dino-tood-fuse512-localdir.yaml \
  data=data/DUO.yaml imgsz=832 batch=8 epochs=200 \
  patience=100 optimizer=SGD lr0=0.01 momentum=0.937 \
  weight_decay=0.0005 box=7.5 cls=0.5 dfl=1.5 \
  device=0 workers=8 amp=False deterministic=False \
  seed=0 name=dino_tood_fuse512_duo_e200
```

For computational profiling, DINOv3 parameter counts include all instantiated weights. GFLOPs measure an unfused batch-one forward pass at 832×832, counting a multiply–add as two FLOPs and excluding preprocessing and NMS. Reproduced RHCNet uses the same convention at 800×1344 (494.12 GFLOPs); published costs retain their source settings.

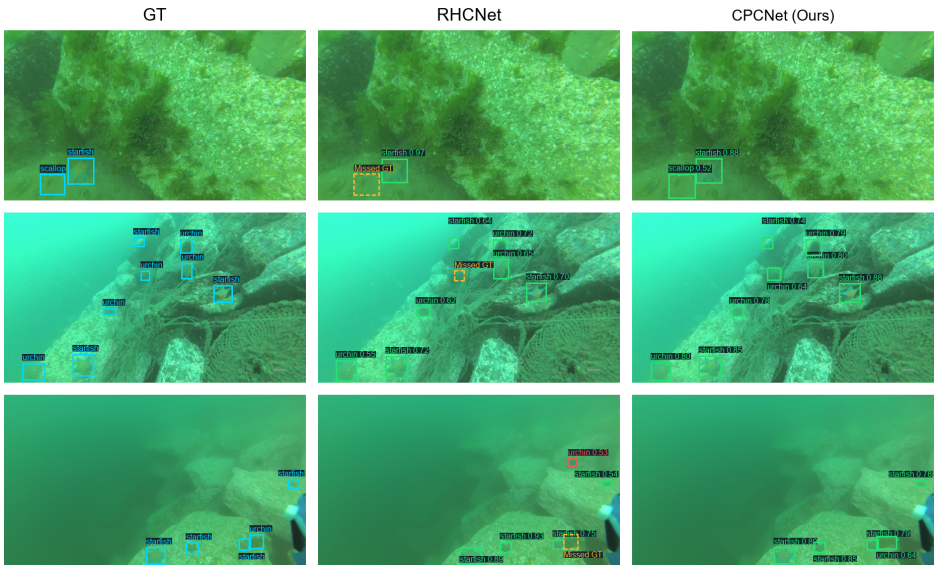

Figure 6: Selected CPCNet detection successes relative to reproduced RHCNet on UTDAC. Columns: ground truth, reproduced RHCNet, and CPCNet (Ours).

G Visualization and Diagnostic Protocols

DUO Case Selection and Matching. Figure 5 uses the ten trained checkpoints in Table 2. Full-validation predictions use size 832, rectangular preprocessing, FP32, NMS IoU 0.70, and at most 1,000 detections per image, filtered at confidence 0.50 for display. Annotations and predictions are mapped from original coordinates to square panels.

Predictions are greedily matched in descending confidence to unmatched same-category GT boxes at IoU ≥ 0.50. Unmatched predictions and annotations count as false positives and misses. Selection compares CPCNet with Simple Pyramid and Parameter-matched Direct: eligible images have no more CPCNet misses or false positives than either baseline, and fewer total errors than each. After excluding previously inspected scenes, 40 candidates from 116 eligible images were inspected. Ranking prioritizes complete CPCNet detection without false positives, then the minimum relative error reduction across the two baselines. All remaining configurations are added without reselection; image identities, prediction-source hashes, and matching counts accompany the artifacts.

For Simple Pyramid, Direct, and CPCNet, respectively, the counts (matched, missed, false positives) are $(1,1,0)/(1,1,1)/(2,0,0)$ in scene 1, $(6,1,1)/(6,1,1)/(7,0,0)$ in scene 2, and $(1,1,1)/(1,1,0)/(2,0,0)$ in scene 3. Source-complete parallel yields $(1,1,1)$, $(7,0,1)$, and $(1,1,0)$, respectively.

G.1 Qualitative Comparison on UTDAC

Figure 6 shows UTDAC predictions at confidence ≥ 0.50, matched to same-category annotations at IoU ≥ 0.50, using each detector's native preprocessing and NMS. Reproduced RHCNet (Wang et al., 2026) uses the same epoch-34 checkpoint as Tables 1 and 11; CPCNet uses its best UTDAC checkpoint. The three scenes were selected for complete CPCNet coverage without extra predictions and incomplete RHCNet coverage, showing low-contrast targets, textured surroundings, and adjacent organisms. RHCNet's matched/missed/unmatched counts are $1/1/0$, $7/1/0$, and $4/1/1$; CPCNet detects all annotations without additional predictions.

G.2 Feature Probes and Localization Recall

Figure 4 uses the best DUO checkpoints of CPC and independently trained No bypass. Frozen models run in evaluation mode with FP32, batch-one inputs letterboxed to 832×832. Hooks capture C_2, C_4, C_5 before FPN–PAN; channel vectors are normalized at each spatial cell.

Separate five-class logistic probes for each model and stage use the same 256 training images and sampled positions (seed 20260924). Foreground cells have centers inside boxes of exactly one category; background footprints intersect no annotation. Sampling retains up to eight cells per category and sixteen background cells per image. Probes use balanced class weights, inverse regularization $C = 10$, and at most 500 iterations. On 128 held-out validation images, macro one-versus-rest AUROC over the four object categories is $93.90/94.76/87.70\%$ for No bypass and $92.58/95.36/87.07\%$ for CPC at $C_2/C_4/C_5$. Average discrimination improves only at C_4.

Maps display selected-category probe probabilities on a common $[0, 1]$ scale, with nearest-neighbor resizing and opacity $0.30p$. These measure feature decodability, not detector confidence. Screening uses 120 candidates plus 140 images with at most four objects for C_2/C_4. CPC's target mean must exceed 0.4 and No bypass's response; target–background contrast, measured against the 95th percentile outside all GT boxes, must exceed 0.1 and improve on No bypass. CPC target-detection IoU must be at least 0.6, with an advantage of 0.1 for C_2/C_5 and no decrease for C_4. Thus the maps show selected positive cases. Each row uses the same crop across models: the first column includes all intersecting GT boxes, and the model columns show intersecting full-detector predictions at confidence ≥ 0.50, capped at 100 per image.

To complement these selected cases, localization recall uses all 1,111 DUO validation images and 10,517 annotations.

Predictions at confidence ≥ 0.50 are sorted and capped at 100 per image. Greedy matching to unmatched same-category annotations is recomputed at IoU thresholds $0.50{:}0.05{:}0.95$; recall is the matched fraction of all annotations. Boxes and recall use rectangular-preprocessing exports from the same checkpoints, whereas probes use square inputs. This fixed-confidence recall differs from COCO AP and AR.

G.3 Failure Analysis and Input Degradation

The failure analysis evaluates CPCNet's best DUO checkpoint on 1,111 images and 10,517 annotations through the main validation pipeline: size 832, rectangular preprocessing, batch size 8, FP32, confidence floor 0.001, NMS IoU 0.70, and at most 1,000 predictions before pycocotools evaluation with maxDets$= [1, 10, 100]$. The clean run reproduces 69.66 AP. All seven conditions retain identical image order, batches, and padded shapes.

Each degradation acts independently on decoded originals before resizing and padding, leaving annotations unchanged. Gaussian blur uses $\sigma = sL/832$, where L is the original longer side and $s \in \{1.5, 3\}$ the input-scale standard deviation. Brightness multiplies intensities by 0.6 or 0.3; contrast uses $\mu + c(I - \mu)$, with per-channel mean μ and $c \in \{0.6, 0.3\}$. Values are rounded and clipped to 8-bit without JPEG re-encoding. AP is $60.13/38.99$ for blur, $69.39/66.29$ for brightness, and $66.68/50.30$ for contrast. These perturbations measure input sensitivity, not physically calibrated turbidity or illumination.

Natural-image recall uses descending-confidence greedy matching to unmatched same-category annotations at IoU ≥ 0.50, confidence ≥ 0.50, and at most 100 predictions per image. Scaling target short sides by $832/L$ gives groups below 16, 16–32, 32–64, and at least 64 pixels: 947, 2,836, 4,731, and 2,003 annotations, with recall 33.69%, 77.12%, 90.91%, and 93.26%. These groups differ from COCO area ranges. Lowering confidence to 0.25 raises recall from 82.49% to 89.26% and unmatched predictions from 703 to 1,584, measured against existing annotations. The diagnostics characterize the full detector rather than individual constructor components.

H Extended Comparisons with Published Detectors

Reproduced RHCNet uses epoch-34 checkpoints tied for highest logged validation AP: 69.90 on DUO and 49.50 on UTDAC. These identify selected checkpoints, not prescribed training durations.

Profiling gives 70.412512M parameters and 494.12 GFLOPs at padded 800×1344, corresponding to resize setting $(1333, 800)$; evaluation shapes vary. Counts include explicit batched matrix multiplication but exclude custom sampling and interpolation.

DUO/UTDAC values follow Wang et al. (2026), except GCC-Net's original AutoAssign configuration (Dai et al., 2024). RUOD values follow Zhou et al. (2025), except the original AMSP-UOD and SFSNN-L reports (Zhou et al., 2024; Chen et al., 2026) and CPCNet.

Tables 9–11 retain source-specific architectures, pretraining, inputs, schedules, and evaluation settings. The numerical rows are transcribed from the cited sources; they are retained for provenance and context, rather than used as protocol-controlled rankings. SFSNN-L uses WaveletSNN-L in spike-based YOLOX; its Table 1 supplies five accuracy metrics, with mAP over IoU $0.50{:}0.05{:}0.95$ (Chen et al., 2026). RHCNet's DUO/UTDAC tables give YOLO backbone labels without all model-scale variants, label CSP-DDC as DJENet, and report CIDNet/DJL-Net values differing from the original DUO $68.8/65.6$ AP (Zhao et al., 2025; Wang et al., 2024b).

Table 9: Published and reproduced detection results on DUO.

Method	Venue	Backbone	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
Faster R-CNN	NeurIPS'15	R-50	41.53	206.68	61.32	81.84	49.12	46.21	62.83
FCOS	ICCV'19	R-50	32.00	196.81	61.54	82.22	49.77	63.56	62.75
GFL	NeurIPS'20	R-50	32.20	78.52	58.66	79.33	46.54	61.63	55.62
YOLOv7	CVPR'23	CSPDarknet53	37.60	64.81	62.35	83.54	46.64	63.79	61.70
GCC-Net	PR'24	Swin-T	36.74	300.79	69.10	**87.80**	–	–	–
DJL-Net	KBS'24	CSP-DDC	33.40	86.35	65.66	84.20	55.63	67.46	64.17
YOLOv10	NeurIPS'24	CSPDarknet	2.71	6.70	60.45	80.38	44.06	63.01	57.77
YOLOv11	2024	C3K2	2.60	6.50	63.46	82.92	52.15	65.10	61.42
CIDNet	KBS'25	R-50	82.50	324.48	68.83	86.56	**56.52**	70.63	67.34
RHCNet	CVPR'26	R-50	70.41	494.12	**69.90**	86.30	54.40	**71.70**	**68.40**
CPCNet (Ours)	–	DINOv3-S/16	63.59	409.66	<u>69.66</u>	<u>87.44</u>	<u>56.40</u>	<u>71.48</u>	<u>68.35</u>

Table 10: Detection results on RUOD across model families.

Method	Venue/year	System family	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
Faster R-CNN	NeurIPS'15	Two-stage CNN (R-50)	41.39	–	49.10	80.30	14.50	37.20	53.90
RetinaNet	ICCV'17	One-stage CNN (R-50)	36.52	–	48.00	77.80	13.20	34.00	53.20
FCOS	ICCV'19	Anchor-free CNN (R-50)	32.13	–	49.10	80.30	14.50	37.20	53.90
YOLOv5-L	2020	YOLO CNN	46.19	54.04	64.00	87.00	<u>22.00</u>	49.40	69.90
ATSS	CVPR'20	One-stage CNN (R-50)	32.13	–	53.90	82.70	16.20	39.10	59.30
DETR	ECCV'20	R-50 + transformer decoder	41.56	–	46.80	78.10	10.60	31.20	52.30
Deformable DETR	ICLR'21	R-50 + deformable decoder	40.10	–	57.80	85.50	19.60	43.00	63.50
TOOD	ICCV'21	Task-aligned CNN (R-50)	32.04	–	55.30	83.10	16.80	40.20	60.80
YOLOX-M	2021	YOLO CNN	25.33	36.88	62.80	87.20	**24.80**	49.20	68.30
RTMDet-L	2022	One-stage CNN	52.26	79.97	64.30	**87.50**	20.10	48.80	70.30
DINO-DETR	ICLR'23	R-50 + transformer decoder	47.56	–	57.80	85.50	19.60	43.00	63.50
DDQ-DETR	CVPR'23	R-50 + transformer decoder	48.34	–	60.80	85.30	20.20	45.80	66.40
YOLOv8-M	2023	YOLO CNN	25.86	39.45	64.80	87.20	18.00	**50.40**	70.90

Table 10 (continued)

Method	Venue/year	System family	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
YOLOv8-L	2023	YOLO CNN	43.64	82.57	65.20	87.00	20.20	49.80	**71.30**
RT-DETR-R101	CVPR'24	R-101 + transformer decoder	60.92	93.20	60.30	84.60	13.20	45.40	66.40
AMSP-UOD	AAAI'24	AMSP-VConv + YOLO	–	–	65.20	86.10	–	–	–
BSR5-DETR(1024)	TPAMI'25	Spatial-residual CNN + DETR	56.87	82.20	63.00	86.00	17.60	47.60	68.90
BSR5-YOLO(E-FPN)-L	TPAMI'25	Spatial-residual CNN + YOLO	36.97	75.37	65.10	87.30	20.60	50.30	71.10
SFSNN-L	AAAI'26	WaveletSNN-L	49.66	167.56	58.30	85.20	13.40	42.70	64.30
CPCNet (Ours)	–	Plain ViT + TOOD	63.60	409.70	**65.27**	**87.50**	21.86	50.27	71.24

Table 11: Published and reproduced detection results on UTDAC.

Method	Venue	Backbone	Params ↓ (M)	GFLOPs ↓	AP ↑	AP_{50} ↑	AP_S ↑	AP_M ↑	AP_L ↑
Faster R-CNN	NeurIPS'15	R-50	41.53	206.68	39.94	60.33	23.02	43.83	52.97
FCOS	ICCV'19	R-50	32.00	196.81	47.55	71.64	21.38	41.94	52.71
GFL	NeurIPS'20	R-50	32.20	78.52	46.80	72.42	20.45	42.22	51.61
YOLOv7	CVPR'23	CSPDarknet53	37.60	64.81	47.82	81.22	21.27	43.61	52.14
DJL-Net	KBS'24	CSP-DDC	33.40	86.35	49.08	84.63	23.19	45.73	55.62
YOLOv10	NeurIPS'24	CSPDarknet	2.71	6.70	47.93	81.32	22.94	44.22	52.36
YOLOv11	2024	C3K2	2.60	6.50	49.75	85.58	23.53	45.84	56.23
CIDNet	KBS'25	R-50	82.50	324.48	49.57	85.38	23.11	45.74	56.08
RHCNet	CVPR'26	R-50	70.41	494.12	49.50	85.50	24.90	44.50	55.40
CPCNet (Ours)	–	DINOv3-S/16	63.59	409.66	**52.18**	**86.18**	**26.76**	**48.23**	**56.95**

I Experimental Scope and Limitations

Each dataset uses a separate training run, with checkpoint selection and reporting on the same local validation split. Topology analysis centers on DUO; CPC, Parameter-matched Direct, and source-complete parallel are also compared on RUOD and UTDAC. There is no cross-dataset transfer test, seed average, or confidence interval. RUOD's `val` entry resolves to 4,200 images in `instances_test.json`, making its result validation-selected rather than an untouched test evaluation. COCO uses official val2017 annotations (Appendix A); the earlier YOLO-generated annotations remain audit artifacts. UTDAC lacks the final-state simple-pyramid control.

Source-complete parallel preserves CPC's four-source coverage at every stage. Its narrower gap than the nonrecursive control does not isolate source coverage, central mixing, or propagation, because mixer widths, resampling paths, operator calls, and gradient routes also differ (Appendix E). All controls retain FPN–PAN and share constructor downsampling with PAN; rankings under BiFPN or independent downsampling operators are unmeasured.

Same-device forward and end-to-end latency remain unmeasured. Published comparisons retain source settings, and encoder comparisons have unmatched optimization histories (Appendices H and B). Selected visual examples illustrate local behavior under the protocols in Appendix G.

These boundaries distinguish two levels of comparison. Within each underwater benchmark, the published-detector tables compare complete systems, so their AP differences also reflect backbone, training, and preprocessing choices. The constructor ablations instead hold the detector framework and evaluation protocol fixed to examine how the selected encoder features are organized.