Frozen-Detector Cluster-Level Late Fusion for Positive-Unlabeled Marine Object Detection

Bo-An Chen[1], Pin-Hao Shen[1] and Yi-Yu Hsu[1,*]

[1]Miin Wu School of Computing, National Cheng Kung University, No. 1, University Road, Tainan City 70101, Taiwan (R.O.C.)

Abstract

This paper describes the system submitted by team HSU Lab to FathomNetCLEF 2026, a 32-category marine object detection task with positive-unlabeled annotations and a distribution shift between the official training image pools and MBARI M3 released-test imagery. In this setting, fine-tuning a detector on the official training split was unreliable in our experiments because unlabeled objects can act as false negatives. Direct score calibration on the training split also did not preserve detection ranking on the released-test imagery. We therefore use cluster-level late fusion over frozen MBARI/FathomNet YOLO-family detectors. The pipeline maps detector labels to the 32 task categories, merges overlapping same-class detections into cluster anchors, computes a conservative base score from detector scores, bounded reliability reweighting, box geometry, and an empirical detector-combination factor, and then applies a PU-aware cluster-level fusion scorer to re-rank the anchors. During training, matched annotations provide positive evidence, while ambiguous or unmatched anchors are not treated as hard negatives. Our system obtains public/private COCO-style bounding-box mAP@[0.50:0.95] scores of 0.3210/0.3042, ranking first on the final leaderboard. Ablations show that detector diversity, PU-aware label handling, the empirical detector-combination factor, duplicate-aware training, and adding the narrow-domain VME detector only at inference and at its best tested single inference scale are useful. These results support cluster-level late fusion over frozen detectors as a practical strategy for marine object detection under missing annotations and distribution shift. Our code is publicly available on GitHub at https://github.com/hsucomputinglab/fathomnetclef2026-frozen-detector-cluster-fusion.

Keywords

marine object detection, positive-unlabeled learning, cluster-level late fusion, distribution shift, FathomNet

1. Introduction

Marine imaging supports biodiversity monitoring, but marine object detection differs from closed-set terrestrial benchmarks. Deep-sea frames often contain small, partially occluded organisms under light attenuation, backscatter or turbidity, motion blur, and high taxonomic diversity with visually similar morphologies [1, 2]. FathomNet was created to support this use case by standardizing and aggregating expert-curated underwater imagery at scale [3]. FathomNetCLEF 2026 [2, 4, 5] defines a 32-category marine object detection task evaluated with COCO-style bounding-box mAP@[0.50:0.95] [6, 7].

The task label space is intentionally consolidated. The official task maps FathomNet labels to 32 groupings assembled for FathomVerse [8], combining morphological and taxonomic categories [2], consistent with FathomVerse's focus on visually distinct deep-sea morphological groups [8]. This is realistic for triaging large marine image archives, but it makes label-space alignment a central requirement because reused detectors have native label spaces that may be finer, broader, or only partially overlapping with the 32 task categories.

Two practical properties shape the modeling problem. One is dataset heterogeneity, because the training split mixes expedition image pools, whereas the released-test split comes from MBARI M3 frame grabs [9]. This creates an in-the-wild distribution shift [10], which Sec. 2 documents. The other is annotation incompleteness under a positive-unlabeled protocol, where a released box marks a verified organism but unannotated regions cannot be treated as reliable background or evidence that a category is absent [2]. This missing-label setting is studied in positive-unlabeled object detection [11]

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany

*Corresponding author.

kevink0725n@gmail.com (B. Chen); chenpinhao910827@gmail.com (P. Shen); yiyuhsu@gs.ncku.edu.tw (Y. Hsu)

0009-0002-2195-3786 (B. Chen); 0009-0002-0382-1392 (P. Shen); 0000-0003-4286-3181 (Y. Hsu)

and motivates the non-negative PU loss used in our validity head training [12]. These properties lead to a simple design. We use cluster-level late fusion over frozen MBARI/FathomNet YOLO-family detectors. We do not fine-tune a new detector on the official training split. Instead, we keep the detector checkpoints fixed and train only a PU-aware cluster-level fusion scorer that combines their detections. This avoids changing detector representations that already encode marine-domain visual structure, and it avoids treating unlabeled regions as reliable background under missing annotations.

Our system follows four steps. First, frozen MBARI/FathomNet YOLO-family detectors produce candidate boxes. Second, their native labels are aligned to the 32 task categories [13, 14, 15, 16]. Third, overlapping same-class detections are merged into cluster anchors. Fourth, a PU-aware cluster-level fusion scorer ranks each anchor using bounded reliability reweighting from observed official-training matches, an empirical detector-combination factor from observed cluster-to-annotation matches, and learned validity/switch signals trained from matched annotations, ambiguous-pair ignore labels, and pairwise ranking signals between competing anchors. Our system reaches public/private COCO-style bounding-box mAP@[0.50:0.95] scores of 0.3210/0.3042. The ablations identify complementary frozen detectors as the largest early contributor. The remaining improvements come from PU-aware label handling, the empirical detector-combination factor, duplicate-aware training, and using the narrow-domain VME detector only at inference and at its best tested single scale.

2. Task setup and data characteristics

2.1. Task, data, and evaluation

The official training split contains 6,463 images and 22,225 released boxes. The released-test JSON contains 1,425 images whose annotations are hidden from participants [2]. Leaderboard scoring evaluates submissions against the hidden test annotations using COCO-style bounding-box mAP@[0.50:0.95] [6, 7]. The target categories are not purely species-level because they combine morphological and taxonomic groupings. The task therefore requires not only localization but also, for methods that reuse pretrained marine detectors, label-space alignment.

2.2. Distribution shift between official training and released-test data

Table 1 and Figure 1 summarize the dataset shift relevant to our design. The official training split combines several image pools with different resolutions, annotation densities, object scales, and visual appearance. In contrast, the released-test JSON points to MBARI M3 frame grabs [9], based on the released `coco_url` patterns, and the released-test images have a different resolution mix and visual style. We use these observations as dataset diagnostics rather than as a formal statistical test. Together, these diagnostics suggest that score adjustments learned only from official training annotations may not preserve detector ranking on the released-test images.

Table 1
Split composition and simple image statistics. Source/pool names are inferred from each image's released `coco_url`. Resolution lists exact sizes for single-resolution pools and dominant sizes for the released-test pool, which contains a few near-variant frame sizes. Luma and saturation are per-image means averaged over all released images in each source/pool after long-side 512 resizing (luma in $[0, 255]$, saturation in $[0, 1]$). Ann./img is the mean number of released boxes per image. Med. box area (%) is the median annotation area $100\, w_{\text{box}} h_{\text{box}} / (w_{\text{image}} h_{\text{image}})$. It is unavailable for the released-test split because test annotations are hidden.

Source / pool	Split	Images	Resolution	Ann./img	Med. box area (%)	Luma / sat.
SOI / Falkor expeditions	train	3,048	3840×2160	5.25	0.433	81.3 / 0.530
HURL / D2-EX2301	train	1,730	1920×1080	1.95	0.212	95.3 / 0.426
NOAA OER / EX1702	train	1,685	1920×1080	1.69	7.981	82.4 / 0.354
MBARI M3 frame grabs	test	1,425	mostly 1920×1080 / 720×486	–	–	92.7 / 0.301

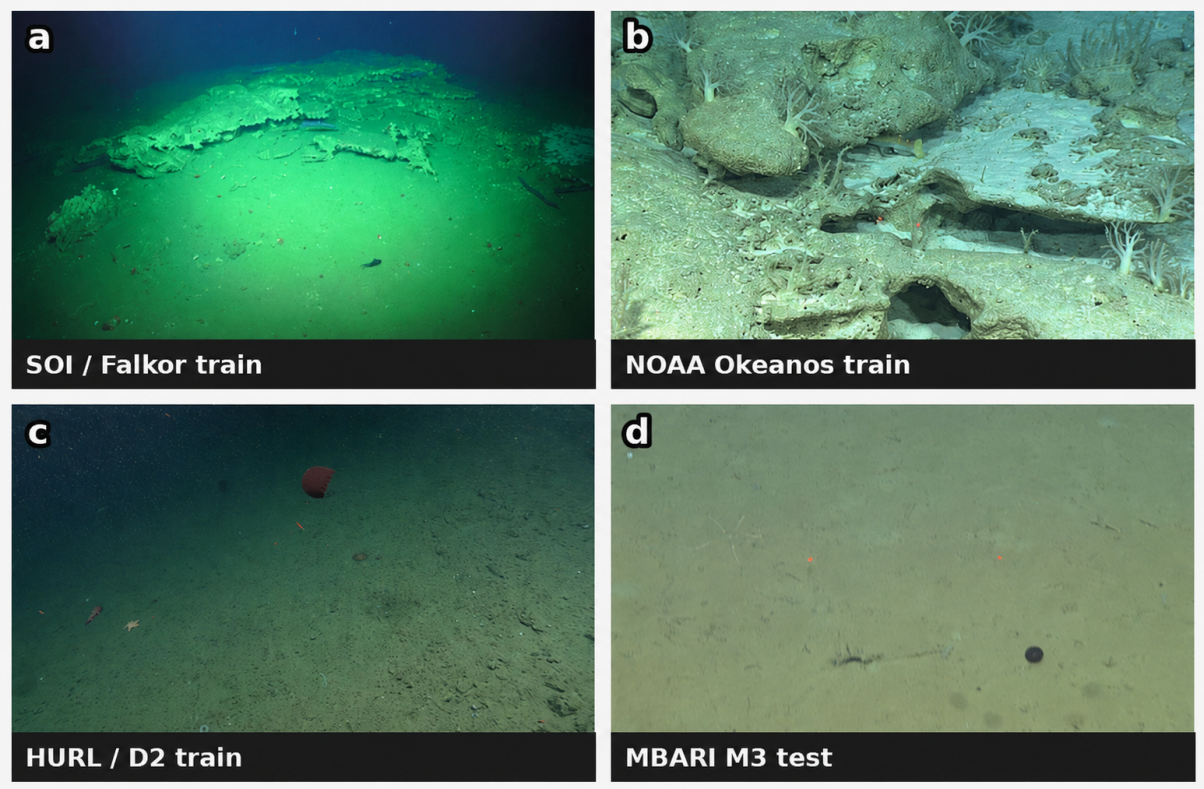

Figure 1: (**a**–**c**) Representative frames from the three official training sources and (**d**) the MBARI M3 released-test pool. The examples show pool-dependent differences in visual appearance and scene composition.

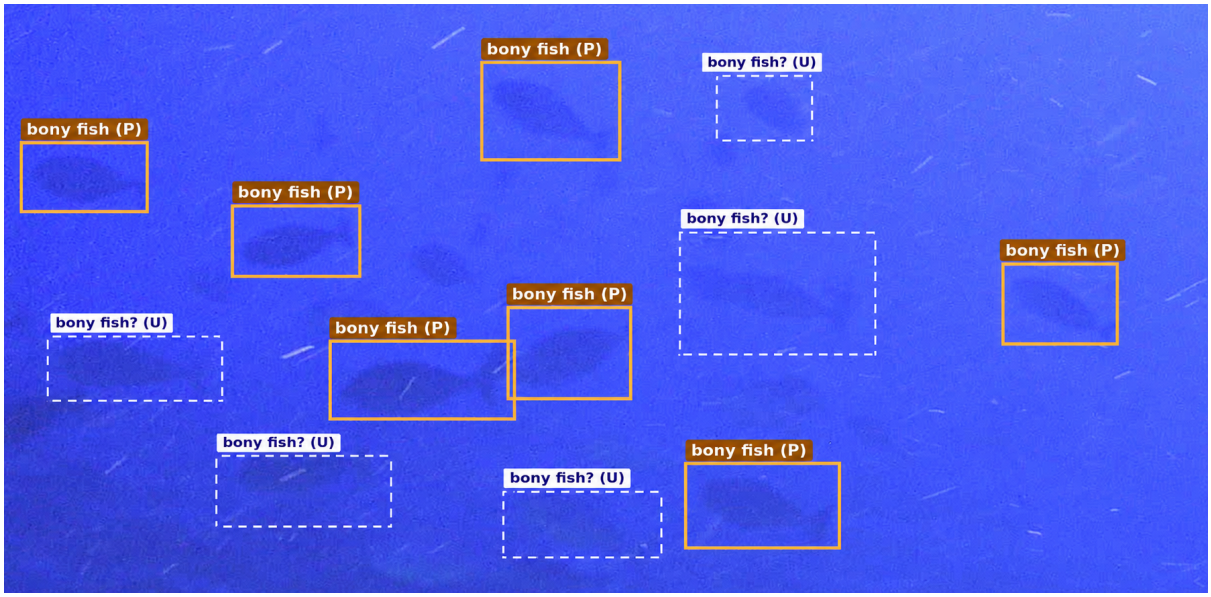

Figure 2: Positive-unlabeled annotations in an official training image. Solid yellow boxes labeled *bony fish (P)* denote released positive annotations. Dashed white boxes labeled *bony fish? (U)* indicate visually plausible fish-like regions without released boxes and are shown only for visualization and are not used as training labels. The example illustrates why unannotated regions are treated as unlabeled.

2.3. Sparse positive-unlabeled annotations

The annotation structure imposes a second, separate constraint beyond the distribution shift. Official training boxes are sparse and unevenly distributed. The train split averages 3.44 boxes per image, has a median of 2, and contains up to 106 boxes in a single frame. Table 1 also shows pool-dependent

annotation density from 1.69 to 5.25 boxes per image. Category frequencies are also concentrated, with urchin, bony fish, sea fan, and brittle star accounting for approximately two thirds of all training boxes. These are released positive boxes, not exhaustive labels. Figure 2 illustrates how released bony-fish boxes can coexist with visually plausible fish-like regions that have no released box.

Since the official task defines the training set as positive-unlabeled [2], unannotated regions may still contain objects. We therefore treat unmatched candidate detections as unlabeled rather than reliable negatives, because supervised detector losses can otherwise turn missing annotations into false-negative training signals [11].

3. Methodology

3.1. Overview

Our method follows the positive-unlabeled principle that the absence of a released annotation is not evidence that an image region is background. Official boxes show objects that have been verified, but they do not prove that the rest of the image is background. For this reason, we keep all MBARI/FathomNet YOLO-family detector weights frozen. We use the official training split to fit only a small PU-aware cluster-level fusion scorer over cluster anchors formed from frozen-detector outputs. This scorer serves as the learned decision layer for those anchors.

Figure 3 shows the training and inference flows of our system. During training, the three broad-domain detectors process official training images. Their detections are mapped to the 32 task categories, merged into same-class cluster anchors, and converted into production base scores and feature vectors. Official positive boxes provide positive evidence for training the fusion scorer. Unmatched anchors and ambiguous pairs are not treated as confirmed background. They are kept unlabeled or ignored when their labels are not reliable.

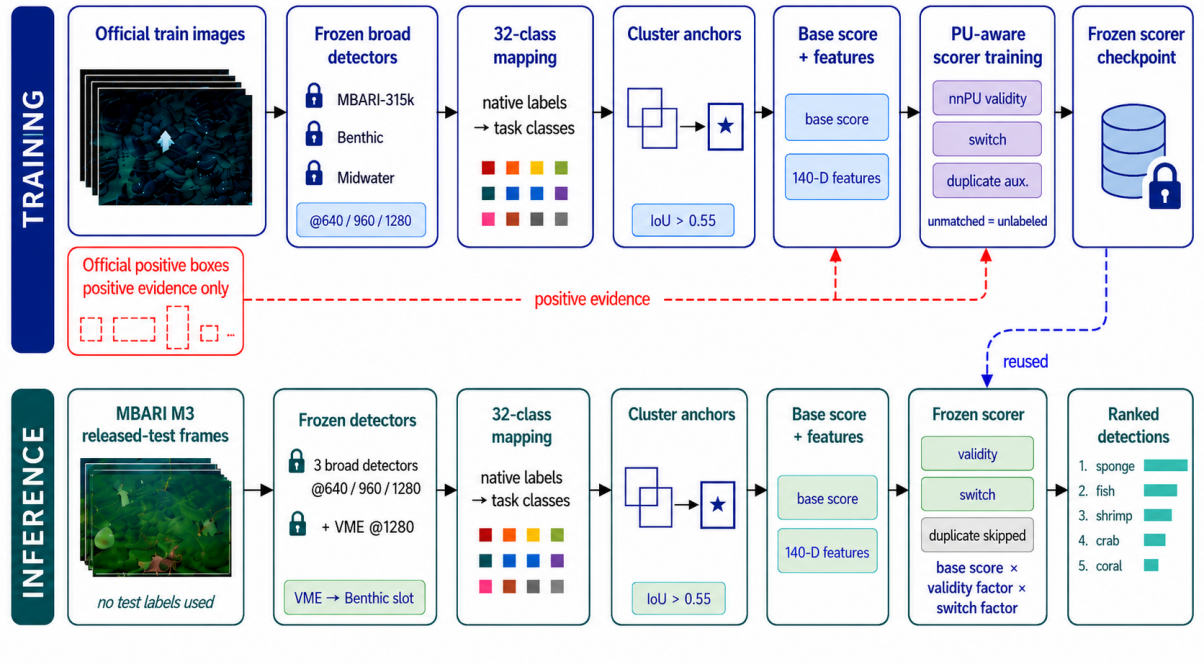

Figure 3: End-to-end training and inference flows. Detector weights remain frozen. During training, official training images and released positive boxes are used to train only the PU-aware scorer block. Both flows apply label mapping, same-class clustering, base-score construction, and feature extraction. VME is added only at inference, and its retained detections are merged into the benthic slot. Final inference queries the validity and switch heads. The duplicate head is used only during training.

At inference, the frozen detectors process MBARI M3 released-test images. The same label mapping, same-class clustering, base-score construction, and feature extraction steps are applied. The narrow-

domain VME detector is added only at inference, and its retained outputs are merged into the benthic detector slot. The frozen scorer then uses only the validity and switch heads to adjust each production base score and rank the final detections. Released-test labels are not used to train the scorer.

3.2. Frozen Detectors and Label Mapping

The final inference detector bank contains four frozen MBARI/FathomNet YOLO-family detectors. Their public checkpoints are listed in Table 2. During scorer training, we build the cluster cache with the three broad-domain detectors, MBARI-315k, benthic, and midwater. These three detectors are run at scales {640, 960, 1280}. The narrow-domain VME detector is added only at inference and is run with its best tested single-scale setting, {1280}. This final inference setting produces about 157k raw detections on the released-test split.

The reused detectors do not share the task's 32-category label space. We therefore map detector outputs to task categories before clustering. For the benthic, midwater, and VME detectors, this is a training-free direct source-label-to-target-label mapping. For MBARI-315k, which predicts 499 species-level classes, we use a training-free taxonomy projection in which species scores assigned to the same task category are pooled into one task-category score, while the original bounding box is kept unchanged. This lets us reuse the pretrained detector without retraining its detection head.

Table 2
Frozen MBARI/FathomNet YOLO-family detectors reused by the detector stage. The table documents the public Hugging Face repository and weight file for each reused checkpoint, and separates native label inventories from the task-label alignment used before clustering. The narrow-domain VME detector is restricted to its best tested single inference scale and is used only at inference. Sec. 5 evaluates this scale choice.

Detector	Public checkpoint	Native labels; task-label alignment	Scales
MBARI-315k	FathomNet/MBARI-315k-yolov8 [13]; weight: mbari_315k_yolov8.pt	native: 499 species; taxonomy projection → 32/32 task classes	640, 960, 1280
Benthic	FathomNet/2025-MBARI-Benthic-Supercategory-Object-Detector [14]; weight: best.pt	native: 29 labels; direct mapping → 20/32 task classes	640, 960, 1280
Midwater	FathomNet/2025-MBARI-Midwater-Supercategory-Object-Detector [15]; weight: best.pt	native: 21 labels; direct mapping → 10/32 task classes	640, 960, 1280
VME	FathomNet/vulnerable-marine-ecosystems [16]; weight: best.pt	native: 4 labels; direct mapping → 3/32 task classes	1280

The VME detector predicts coral, crinoid, sponge, and fish. We drop the coral output because the task contains several coral-related categories, and the VME coral label is too coarse to choose among them reliably. The retained VME detections for crinoid, sponge, and fish are added only at inference. In the fusion stage, they are counted as evidence in the benthic detector slot, not as a fourth detector slot. As a result, the trained scorer receives the same three detector slots during both training and inference, MBARI-315k, benthic, and midwater.

Label-mapping provenance and validation. We built the detector-to-task label correspondences above once, offline, and training-free, using no official training labels. We drafted them with the assistance of a large language model from each detector's published label list and the task-category definitions, and then validated them in two ways. First, empirically: of the 32 task categories, 30 receive correctly matched MBARI-315k detections on official-training data; only sea slug and hydroid do not, and the audit attributes this to detector difficulty rather than a mapping error. Second, by a per-entry taxonomic audit of the 392 MBARI-315k concept assignments (Appendix B), which confirmed about 93% of them, identified 16 clear corrections (11 re-classifications in Table 21 and 5 non-target taxa in

Table 22), and flagged a small set of borderline higher-rank concepts for expert review. We release the full tables and audit so that domain experts can correct any residual errors. Crucially, the pipeline does not require a perfect mapping: the bounded reliability reweighting and the PU-aware scorer are fit to official-training matches and automatically down-weight unreliable detector-class routes, so systematic mapping errors are mitigated empirically rather than assumed absent. The reported results use the mapping as submitted; we release the audited corrections separately and did not apply them to the submitted system.

For sea slug and hydroid, the two categories with no matched MBARI-315k detections, the audit found the correspondences taxonomically correct; the absence of matches reflects detector difficulty—very small or rare training samples and thin colonial morphology that resists tight bounding boxes—rather than a mapping error.

3.3. Cluster Anchor Construction

The fusion stage first turns raw detections into cluster anchors. Each anchor is a same-class detection cluster represented by a score-weighted fused bounding box, and it is the unit scored by the downstream base-score computation and fusion scorer. Within each image, detections are processed in descending score order. The highest-scoring unused detection becomes a seed, and lower-scoring detections of the same class are merged into it when their IoU with the seed is greater than 0.55. The cluster stores the fused box, the supporting detector slots, the supporting scales, and the detector scores of the detections merged into the cluster.

Only detections of the same class are merged. Detections of different classes remain as separate anchors, even when their boxes overlap. This is important because a nearby different-class anchor may be a useful competitor. For example, the same image region may be detected as both sponge and amphipod. The learned fusion scorer later uses the strongest overlapping different-class anchor to compute competitor features, rather than forcing a class decision during clustering. Figure 4 illustrates this rule. Same-class sponge detections are merged, while a nearby different-class amphipod detection remains separate and can later serve as a competing anchor.

This clustering step only defines candidate anchors. It does not decide the final ranking. Ranking is handled by the production base score (Sec. 3.4) and the PU-aware cluster-level fusion scorer (Sec. 3.5).

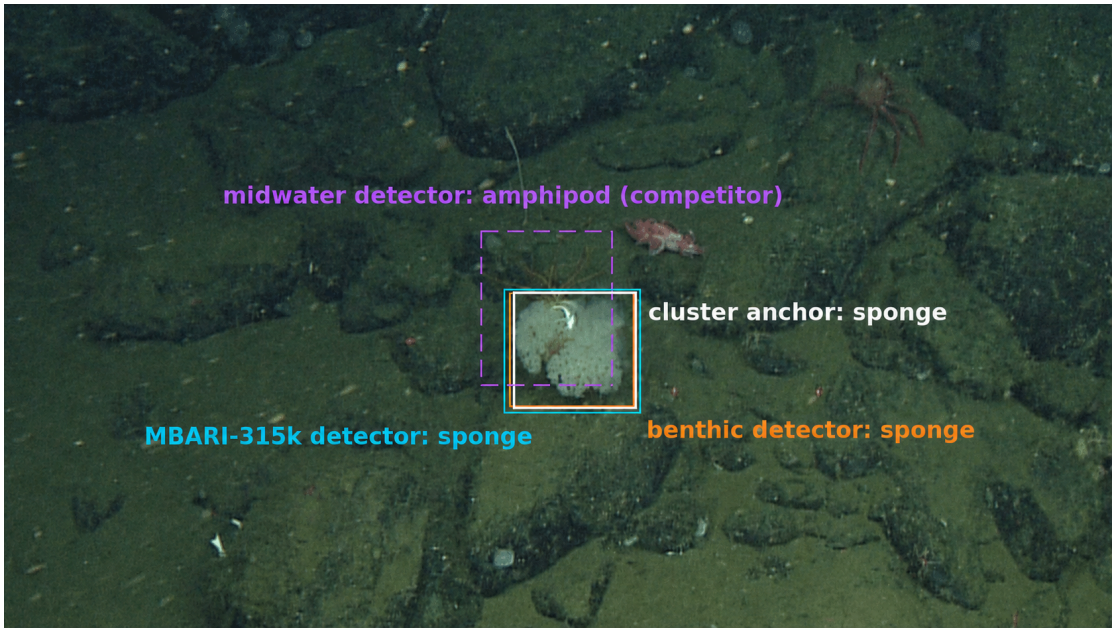

Figure 4: Same-class cluster-anchor construction on an official training image. Overlapping sponge detections from different detector sources are merged into a single sponge cluster anchor. The purple amphipod detection is a different-class neighbor. It is not merged during clustering, but remains available as a competing anchor.

3.4. Base Score and Empirical Detector-Combination Factor

Before the learned scorer is applied, each cluster anchor receives a conservative base score used by the production inference pipeline. We refer to this scalar as the production base score. A detector slot is one scoring source (MBARI-315k, benthic, or midwater). Retained VME detections are counted in the benthic detector slot at inference. The base score summarizes four signals. First, each detector slot contributes its strongest detector score for the anchor. Second, detector scores are adjusted by bounded reliability reweighting estimated from observed same-class matches on the official training split. Bounded reliability reweighting is not probability calibration. It is a bounded detector-category reliability factor based on observed positive evidence. Third, a small geometry multiplier down-weights boxes whose size or aspect ratio is unusual for the predicted class. Fourth, an empirical detector-combination factor adjusts the score according to which detector slots support the anchor.

The empirical detector-combination factor asks which detector slots supported the anchor. For each support pattern, such as MBARI-315k only, benthic only, or MBARI-315k plus benthic, we measure how often anchors with that pattern match a same-class official training annotation at IoU ≥ 0.5. Because the training labels are positive-unlabeled, this value should not be read as the true precision of the support pattern. Nonmatched anchors may still contain real objects. We therefore treat the match rate as a conservative empirical factor rather than as a precision estimate. We smooth it toward the global observed match rate and clip it to a bounded range. The resulting values are smoothed observed-match factors, not calibrated probabilities of correctness.

We define $\mathcal{S}(a)$ as the detector slots that support anchor a, and $\tilde{p}_{a,t}$ as the strongest detector score in slot t after applying the bounded detector-category reliability factor. We combine slot evidence with a noisy-OR score [17],

$$q_a = 1 - \prod_{t \in \mathcal{S}(a)} (1 - \tilde{p}_{a,t}).$$

The production base score is then

$$b_a = \mathrm{clip}(q_a g_a c_a,\ 10^{-4},\ 0.99),$$

where g_a is the geometry multiplier and c_a is the empirical detector-combination factor. The base score is the starting point for the learned scorer and is not the final submitted score.

For reproducibility, Table 3 lists the fixed constants used by the base score.

Table 3
Constants used by the production base score.

Item	Setting
Anchor match	Same task class and box IoU ≥ 0.5.
Detector-category reliability	For each detector slot and task class, matched supported anchors divided by supported anchors.
Reliability smoothing	Detector-category observed match rates use smoothing strength 20.
Reliability range	The bounded reliability factor is clipped to $[0.65, 1.25]$.
Geometry statistics	Class statistics use log normalized box area and log aspect ratio from official training boxes.
Geometry range	The geometry multiplier lies in $[0.9, 1.0]$. Classes without enough boxes use 1.0.
Support-pattern smoothing	Detector support-pattern rates use smoothing strength 100 and exponent 0.5.
Support-pattern range	The support-pattern factor is clipped to $[0.20, 1.10]$.
Unseen support pattern	The implementation falls back to factor 1.0.
VME handling	Retained VME detections use the benthic support slot and are not a fourth detector slot.

3.5. PU-Aware Cluster-Level Fusion Scorer

The PU-aware cluster-level fusion scorer is the component shown as *PU-aware scorer training* during training and as *Frozen scorer* during inference in Figure 3. It is not an object detector. detector weights remain fixed, no new boxes are generated, and the module only re-ranks cluster anchors produced by the frozen detectors.

Each anchor is represented by a 140-dimensional feature vector. The vector has three parts, comprising 66 anchor features, 66 competitor features, and 8 relation features. The anchor features describe the current cluster anchor, including score statistics, detector support, scale support, and box geometry. The competitor features describe the strongest overlapping anchor from a different class. The relation features describe how the two anchors interact, including their box IoU and score difference. The competitor is the highest-scoring different-class anchor with box IoU ≥ 0.45. If no such competitor exists, the competitor part uses the implementation's default no-competitor encoding.

Table 4 summarizes the three feature groups and their dimensions; Table 19 in Appendix A gives the full per-feature enumeration, and the feature builder is released in `anchor_features.py` for exact reproduction.

Table 4
Feature-vector blocks. The anchor part is $66 = 15 + 9 + 32 + 6 + 4$; the competitor part repeats the same 66 features on the competitor anchor; the relation part is 8, for 140 in total. Full enumeration in Table 19.

Block	Dim	Contents
Cluster score & geometry	15	base, raw, and reweighted noisy-OR; detector, scale, and cluster counts; detector & scale bonuses; geometry multiplier; score mean and std (raw and reweighted); log area; log aspect
Per-detector support	9	raw support, reweighted support, and reliability factor for each of the three broad detectors
Class identity	32	task-category one-hot
Image context	6	log width, log height, luma mean, luma std, mean$(G - R)$, mean$(B - G)$
Image regime	4	k-means (k=4) regime one-hot over the image-context features
Competitor	66	the same 15+9+32+6+4 block computed on the competitor anchor
Relation	8	constant bias, anchor–competitor IoU, base-score difference, reweighted noisy-OR difference, same coarse-family indicator, detector-set Jaccard, area ratio, center-distance ratio

The scorer has three heads during training. The validity head estimates whether the current anchor is likely to correspond to a real object. It is trained with a non-negative PU loss [12], so unmatched anchors are treated as unlabeled rather than hard negatives. The switch head estimates whether an overlapping different-class competitor should take priority over the current anchor. It is trained only on pairs whose label can be defined from reliable positive evidence. Ambiguous pairs are ignored. The auxiliary duplicate head marks same-class near-duplicates of a better neighboring anchor. This duplicate head is used only as a training signal and is not queried in the final inference pipeline.

The final scorer is trained from a cache built with the three broad-domain detectors, MBARI-315k, benthic, and midwater. The narrow-domain VME detector is not included in this training cache. At inference, retained VME outputs are merged into the benthic detector slot. This keeps the scorer input format the same during training and inference. The scorer always receives the three slots, MBARI-315k, benthic, and midwater.

Training losses. These losses train only this learned scorer over precomputed cluster anchors. They are not detector losses. For the validity head, anchors matched to official positive boxes provide positive examples, while unmatched anchors are unlabeled. We use the non-negative PU risk estimator with positive prior $\pi = \min(0.45, \max(0.15, r_+/0.7))$, where r_+ is the observed positive fraction in the training anchors. The 0.7 denominator rescales r_+ upward for the positive-unlabeled setting, where

the matched fraction under-estimates the true prior because some positives are unannotated; it corresponds to an assumed annotation completeness of ≈70%, and a sensitivity sweep (Table 17) confirms it is near-optimal. The switch and duplicate heads use weighted binary cross-entropy only for defined targets, and targets with value −1 are ignored. Pairwise ranking loss is applied only to anchor pairs whose ordering is defined by reliable positive evidence. We also use a small score-consistency regularizer, $\mathscr{L}_{\text{score}}$, which penalizes large relative changes between the training-time adjusted score and the production base score. The released scorer is trained with

$$\mathscr{L} = \mathscr{L}_{\text{valid}} + 0.35\,\mathscr{L}_{\text{switch}} + 0.20\,\mathscr{L}_{\text{dup}} + 0.25\,\mathscr{L}_{\text{rank}} + 0.05\,\mathscr{L}_{\text{score}}.$$

This loss is PU-aware because observed positives provide positive evidence, while unlabeled anchors are not forced to be negatives.

At inference, we use only the validity and switch heads. The duplicate head is skipped. Its value remains in the training recipe through duplicate supervision, duplicate-related training-time score terms, and same-class duplicate ranking pairs that shape the training signals used by the scorer. Sec. 6.2 measures the joint effect end-to-end.

3.6. Inference-Time Score Combiner

At inference, the submitted score starts from the production base score b_a. The validity head can raise or lower this score around its neutral point 0.5. The switch head can lower the score when a competing different-class anchor should take priority. The duplicate head is not queried at inference, and the final submission does not use targeted NMS.

For anchor a, the final score is

$$F_a = \text{clip}\big(b_a \cdot (1 + s_v(p_v - 0.5)) \cdot (1 - s_s\, g(p_s)),\ 10^{-4}, 0.99\big),$$

where p_v and p_s are the validity and switch probabilities from the frozen scorer. We set s_v=0.55 and s_s=0.10. The switch gate is $g(p)$=clip$((p-0.10)/0.90, 0, 1)$. The final score is used for ranking submitted detections and should not be interpreted as a calibrated probability of correctness.

4. Baseline failures and staged ablation study

Having defined the full pipeline, we now use public-leaderboard submissions to explain why each major component was added. Public mAP was the feedback available during system construction, so the staged analysis is based on public mAP.

The evidence is organized in three parts. First, direct baselines show why detector fine-tuning and detector-only ranking were not sufficient. Second, the training-stage construction path builds the frozen fusion-scorer checkpoint. Third, the inference-stage path shows how the frozen checkpoint is used in the final submitted system.

Each row in the staged construction paths is a separate submitted run. The deltas should therefore be read as staged development evidence, not as independent causal attributions. Full-pipeline component ablations in Sec. 6.3 provide stronger evidence for selected PU-related training components.

4.1. Direct baseline failures

Table 5 lists standalone detector-only baselines before cluster-level late fusion.

The direct YOLO fine-tuning run scores only 0.030 public mAP, consistent with the two constraints documented in Sec. 2. Missing annotations can create false-negative training signals, and released-test images differ from official training image pools. Direct detector-score calibration fitted on the official training split also underperforms. Even the strongest detector-only baseline, a frozen 3-detector MBARI ensemble with multi-scale test-time augmentation, reaches only 0.280 public mAP. These results motivate cluster-level late fusion over frozen MBARI/FathomNet YOLO-family detectors in the rest of the paper.

Table 5
Standalone direct detector-only baselines before cluster-level late fusion.

Approach	Public mAP	Observation
Direct YOLO [18] fine-tuning on the official training split	0.030	Consistent with false-negative signals from missing annotations and distribution shift.
Frozen MBARI/FathomNet YOLO-family inference + direct detector-score calibration	0.231	Official-training calibration does not preserve released-test ranking.
Frozen 3-detector MBARI ensemble (no score adjustment)	0.247	Unweighted score summation does not model class-specific detector reliability.
Single frozen MBARI-315k detector (direct detector-only ranking)	0.256	Marine-domain prior only, without cluster-level late fusion or multi-detector support.
Frozen 3-detector MBARI ensemble + multi-scale test-time augmentation	0.280	Strongest detector-only baseline, still below cluster-level late fusion.

4.2. Training Stage for Building the Fusion Scorer

Table 6 starts from a single MBARI-315k detector whose 499-class outputs are aligned to the 32 task categories by taxonomy projection and run at $\{640, 960, 1280\}$. This baseline scores 0.2861 public mAP. Steps 1–6 then build the training-stage construction path and end with a frozen step-6 fusion-scorer checkpoint at 0.3155 public mAP.

The direct detector-score calibration baseline in Table 5 differs from bounded reliability reweighting in Table 6. The former ranks detector boxes directly on the released-test set, while Step 3 applies bounded reliability reweighting only inside cluster-anchor scoring, before the empirical detector-combination factor and the PU-aware cluster-level fusion scorer.

The largest gain comes from adding the benthic and midwater detectors inside same-class clustering. This increases public mAP from 0.2861 to 0.3054 (+0.0193), indicating that complementary frozen detectors are the strongest early contributor in this construction path. Bounded reliability reweighting then improves the path to 0.3116. The empirical detector-combination factor is a small local regression in this staged path, but the full-pipeline ablation in Sec. 6.3 shows that removing it before retraining the released scorer is harmful. We therefore treat Table 6 as a construction path, not as a strictly additive decomposition of independent effects.

The step-6 checkpoint is frozen after the duplicate-aware training recipe is added. All later rows in Table 7 reuse this checkpoint and change only inference-time components.

Table 6
Training-stage construction path. Each row is a separate submitted run, but the deltas should be read as staged development steps rather than independent causal attributions.

Step	Configuration / component added	Public mAP	Δ
1	Single MBARI-315k detector (taxonomy projection, scales $\{640, 960, 1280\}$)	0.2861	n/a
2	+ Benthic and Midwater frozen detectors inside same-class clustering (no bounded reliability reweighting)	0.3054	+0.0193
3	+ Bounded reliability reweighting from observed official-training matches	0.3116	+0.0062
4	+ Empirical detector-combination factor for the cluster base score (Sec. 3.4)	0.3108	−0.0008
5	+ PU-aware cluster-level fusion scorer (validity + switch heads, pairwise ranking loss)	0.3129	+0.0021
6	+ Duplicate-aware training recipe (auxiliary duplicate head + duplicate-aware score/ranking terms)	0.3155	+0.0026

4.3. Inference Stage with the Frozen Scorer

Starting from the frozen step-6 scorer, Table 7 changes only inference-time components, namely the detection cache, switch parameters, and inference architecture. Adding the narrow-domain VME detector gives a small gain. The switch-parameter setting gives the largest pre-final increase in this construction path. Dual-head inference without targeted NMS is nearly neutral, but it simplifies the production path by querying only the validity and switch heads.

Table 7
Inference-stage construction path from the frozen step-6 scorer. Rows change inference-time components only.

Step	Configuration / component added	Public mAP	Δ
7	+ Add narrow-domain VME detections to the benthic support slot, scales {640, 1280}	0.3162	+0.0007
8	+ Uniform 3-scale {640, 960, 1280} for all four detectors	0.3167	+0.0005
9	+ Set tuned switch parameters `switch_scale=0.10` and `switch_mask_mode=all`	0.3186	+0.0019
10	+ Dual-head inference (skip duplicate head) without targeted NMS	0.3187	+0.0001
11	+ VME at best tested single inference scale {1280} while broad-domain detectors remain at {640, 960, 1280}	0.3210	+0.0023

The final inference-stage step restricts the narrow-domain VME detector to its best tested single inference scale, {1280}, while keeping the three broad-domain detectors at {640, 960, 1280}. This reaches the final public mAP of 0.3210. Sec. 5 explains this scale choice, and Sec. 6 analyzes the switch parameters and duplicate-aware training recipe.

5. Multi-detector × multi-scale extension

5.1. Detector Diversity and Scale Behavior

We next study scale behavior after detector diversity is fixed. Table 8 forces all four detectors to use the same scale set and varies the number of scales from one to five. The best tested uniform setting is the three-scale set {640, 960, 1280}, which scores 0.3167 public mAP. Adding more scales does not help, and the four-scale and five-scale settings both remain below the three-scale setting.

This result shows that scale diversity is useful, but only up to a point. The one-scale to three-scale gain is similar in size to the gain from adding complementary detectors in Table 6. However, the later downturn shows that more scales are not automatically better. One plausible reason is that the bounded reliability reweighting tables and fusion scorer were built around the original tested scale vocabulary, not the expanded four- or five-scale sets.

Table 8
Uniform N-scale curve across all four detectors. Performance peaks at the best tested 3-scale set.

N	Configuration	Public mAP	Marginal Δ
1	{640} only	0.2952	n/a
2	{640, 1280}	0.3131	+0.0179
3	{640, 960, 1280} (best tested)	0.3167	+0.0036
4	{512, 640, 960, 1280}	0.3149	−0.0018
5	{512, 640, 960, 1280, 1920}	0.3150	+0.0001

5.2. Narrow-domain versus broad-domain scale behavior

The final system does not use the same scale set for every detector. The three broad-domain detectors use {640, 960, 1280}, while the narrow-domain VME detector uses only {1280}. Table 9, Table 10, and

Table 11 explain this non-uniform choice.

The VME scale controls in Table 9 and Table 10 are evaluated under the step-9 predecessor in Table 7, with the four-detector uniform 3-scale cache, three-head scorer, tuned switch parameters, and targeted NMS, which scores 0.3186 public mAP. Thus the 0.3208 VME @ {1280} row isolates only the VME scale assignment under that predecessor. The final 0.3210 result in Table 7 then adds the later inference simplification from Sec. 3.5, namely dual-head inference without targeted NMS.

Table 9 and Table 10 vary only the VME scale assignment while keeping the three broad-domain detectors at three scales. Under the step-9 predecessor, every single-scale VME setting beats the three-scale VME baseline, and VME @ {1280} is the best tested VME setting. Two-scale VME settings are also worse than VME @ {1280}. In contrast, Table 11 restricts each broad-domain detector to a single scale and shows that all six single-scale broad-domain controls regress. This contrast supports the final scale design. Broad-domain detectors benefit from multi-scale inference, while the narrow-domain VME add-on is most useful at its best tested single inference scale.

Table 9
VME single-scale variants under the step-9 three-head scorer + targeted NMS predecessor. Every single-scale VME run beats the 3-scale baseline by +0.0010 to +0.0022 mAP.

Variant	VME scales	Public mAP	Δ vs 3-scale baseline
VME at uniform 3-scale baseline	{640, 960, 1280}	0.3186	n/a
VME @ 1920 only	{1920}	0.3196	+0.0010
VME @ 512 only	{512}	0.3200	+0.0014
VME @ 640 only	{640}	0.3202	+0.0016
VME @ 960 only	{960}	0.3204	+0.0018
VME @ 1280 only	{1280}	0.3208	+0.0022

Table 10
Two-scale VME variants under the same step-9 predecessor. Deltas are reported against the best tested VME setting, VME @ {1280}.

Variant	Public mAP	Δ vs VME @ {1280}
VME @ {960, 1280}	0.3204	−0.0004
VME @ {640, 1280}	0.3188	−0.0020

Table 11
Single-scale broad-domain detector controls. Each row restricts one broad-domain detector while leaving the rest of the step-9 predecessor unchanged.

Restricted detector	@ {960}	@ {1280}
MBARI-315k	0.3099 (−0.0087)	0.3123 (−0.0063)
Benthic	0.3145 (−0.0041)	0.3141 (−0.0045)
Midwater	0.3183 (−0.0003)	0.3182 (−0.0004)

Mechanism. We interpret the VME scale result as a redundancy effect. VME contributes only three mapped classes, crinoid, sponge, and fish. These classes overlap strongly with the benthic detector, and VME detections are merged into the benthic detector slot rather than treated as a new detector identity. Multi-scale VME can therefore add extra boxes around objects already covered by benthic detections. If these boxes merge into the same cluster, they add little new independent evidence because same-slot support is max-pooled. If they do not merge, they can create extra same-class anchors.

The final inference path does not explicitly suppress these same-class VME duplicates. The switch head compares different-class competitors, the duplicate head is skipped, and targeted NMS is disabled. Running VME at its best tested single inference scale preserves the observed VME gain while reduc-

ing this duplicate structure. The broad-domain detectors behave differently because their multi-scale outputs are part of the scorer's training distribution and cover a wider set of task categories.

5.3. Expanded scale sets regress

The right side of the scale curve in Table 8 comes from expanding the tested scale set beyond {640, 960, 1280}. Uniform settings that add 512 and/or 1920 regress by about 0.0017–0.0018 public mAP relative to the best tested three-scale set. One possible reason is that the bounded reliability reweighting tables and the fusion scorer were fitted around the original scale vocabulary, so detections from additional scales may be scored less reliably.

5.4. Single-class detector add-on

We also tested whether a narrow single-class add-on would help. As a negative control, we added the public FathomNet MegaFishDetector, a YOLOv5-based generic fish detector [19, 20], and mapped its detections to the bony fish class. This regressed to 0.3149 public mAP (−0.0006). Restricting VME to emit only one of its three retained classes also lost 0.0013–0.0026 public mAP. These runs suggest that the useful VME signal comes from multi-class narrow-domain coverage, not from simply increasing detections for one class.

5.5. Retraining with the narrow-domain VME detector

A natural follow-up is to ask whether the fusion scorer should be retrained with VME detections in the training cluster cache. Table 12 tests this idea with single-scale and multi-scale VME training caches. Both retrained scorers regress relative to the final no-retrain method. The multi-scale VME retrain is less harmful than the single-scale VME retrain, but neither improves the public leaderboard result.

Table 12
Retraining with VME in the training cluster cache regresses for both single-scale and multi-scale assignments.

Variant	Train VME scales	Train anchors	Public mAP	Δ
Final method (no retrain)	n/a	248,674	0.3210	n/a
Single-scale VME retrain	{1280}	285,982	0.3129	−0.0081
Multi-scale VME retrain	{640, 960, 1280}	298,327	0.3166	−0.0044

The final pipeline therefore keeps the step-6 scorer trained on the three base detectors and adds VME only at inference. This avoids the training-cache drift observed when VME is included during training while still using VME as narrow-domain evidence for crinoid, sponge, and fish. This result supports the training-stage / inference-stage split used in Sec. 4.

6. Switch-parameter and training-recipe ablations

This section analyzes the learned scorer after the main construction path has been fixed. Sec. 6.1 tunes the switch-head inference parameters. Sec. 6.2 separates inference-time head usage from the duplicate-aware training recipe. Sec. 6.3 retrains the full pipeline after replacing one core recipe component at a time.

6.1. Switch-parameter tuning

We tune only two inference parameters for the switch head, `switch_scale`, which controls the strength of switch suppression, and `switch_mask_mode`, which controls whether the switch signal is applied only to ambiguous anchors or to all anchors with a defined competitor. In this parameter grid, the best tested setting is `switch_scale=0.10` with `switch_mask_mode=all`, which moves the matched step-8

predecessor from 0.3167 to 0.3186 public mAP. This is a parameter-tuning result over nonzero switch settings, not a complete switch-head ablation.

6.2. Dual-head inference and duplicate-aware training

After switch-parameter tuning, we tested whether the validity head, the duplicate head, and targeted NMS were needed under the same 4-detector 3-scale parameter configuration. Table 13 shows that the validity head has the clearest isolated inference-time effect. Disabling it reduces public mAP by 0.0023. In contrast, disabling the duplicate head at inference and removing targeted NMS are both nearly neutral. We therefore use dual-head inference without targeted NMS in the final submission. The production path queries only the validity and switch heads and skips duplicate-neighbor selection.

Table 13
Inference-time controls under the matched 4-detector 3-scale parameters. The table isolates the validity head, duplicate head, and targeted NMS. It does not independently isolate the switch head.

Variant	Public mAP	Δ vs 0.3186
Full pipeline (three-head scorer + targeted NMS)	0.3186	n/a
Validity head off (`valid_scale=0`)	0.3163	−0.0023
Duplicate head off (`duplicate_scale=0`)	0.3187	+0.0001
Targeted NMS off	0.3185	−0.0001
All heads off + targeted NMS off (base score only)	0.3165	−0.0021
Dual-head inference + no targeted NMS	0.3187	+0.0001

This inference simplification does not mean that duplicate-aware training is unnecessary. Table 14 retrains a dual-head-only model that removes the duplicate head, duplicate BCE loss, duplicate factor in the training-time score combiner, and same-class duplicate ranking pairs.

The retrained dual-head-only model loses 0.0049 public mAP. This shows that the duplicate-related terms are useful during training, even though the duplicate head is skipped at inference. We treat this as evidence for the duplicate-aware training recipe as a whole. We do not claim that each duplicate-related term is independently necessary, because we did not run the full factorial ablation.

Table 14
Duplicate-aware training-recipe ablation. The dual-head-only retrain removes the duplicate head and its associated duplicate BCE loss, the duplicate factor in the training-time score combiner, and the same-class duplicate pairs from the pairwise ranking loss simultaneously.

Training recipe	Inference architecture	Public mAP
Duplicate-aware recipe with validity, switch, and auxiliary duplicate heads, with duplicate-related training terms	Dual-head inference (skip duplicate head, no targeted NMS)	0.3187
Dual-head-only retrain with validity and switch heads, with all duplicate-related training terms removed	Dual-head inference	0.3138

Per-component factorial. Table 14 measures the duplicate-aware recipe as a whole at the 0.3187 stage. In the final configuration (VME at {1280}, dual-head inference; 0.3210 public mAP), a per-component factorial (Table 15) isolates the three duplicate terms. Removing all three costs 0.0020; the training-time duplicate factor is removable (±0.0000); and removing the duplicate BCE most affects the validity head (its AUC drops from 0.870 to 0.678) and public mAP (−0.0041). The single-term removals are non-monotonic at the leaderboard noise floor—removing all three costs less than removing the BCE alone—so the effect is recipe-level rather than attributable to any single term.

Table 15
Per-component factorial of the duplicate-aware recipe in the final configuration (public mAP 0.3210). Validity-head AUC is on the official-train validation split.

Variant	Public mAP	Δ	Validity AUC
Full duplicate-aware recipe	0.3210	—	0.870
− duplicate BCE loss	0.3169	−0.0041	0.678
− training-time duplicate factor	0.3210	±0.0000	0.870
− same-class duplicate rank pairs	0.3190	−0.0020	0.870
− all three	0.3190	−0.0020	0.870

6.3. Recipe-component ablations on the full pipeline

Finally, we test whether three core recipe components are important in the full pipeline. For each ablation, we retrain the fusion scorer with one component replaced and run the same final inference configuration with three broad-domain detectors at {640, 960, 1280}, VME at {1280}, tuned switch parameters, and dual-head inference without targeted NMS.

The first ablation replaces the non-negative PU loss [12] for the validity head with weighted BCE that treats unmatched anchors as hard negatives. The second ablation replaces the PU-aware ignore policy for ambiguous switch and duplicate pairs with forced hard labels. This expands switch-defined items from ∼ 3.3k to ∼ 200k and duplicate-defined items from ∼ 2.5k to ∼ 200k. The third ablation rebuilds the training dataset and base scores with all empirical detector-combination factor values forced to 1.0, removing the empirical factor from the base score. All other architecture choices, hyperparameters, image splits, and seeds are kept unchanged within each comparison.

Table 16
Recipe-component ablations on the full pipeline.

Variant	Public mAP	Δ vs final
Final method	0.3210	n/a
nnPU → BCE-hard-negative validity	0.3154	−0.0056
PU-aware ignore policy → forced labels	0.3183	−0.0027
empirical detector-combination factor → identity values	0.3155	−0.0055

Table 16 shows that all three replacements reduce public mAP. Replacing nnPU with BCE-hard-negative validity loses 0.0056 mAP, replacing the PU-aware ignore policy loses 0.0027 mAP, and removing the empirical detector-combination factor loses 0.0055 mAP. These results show that the final recipe is PU-aware by design and that, in these full-pipeline retrains, its PU-aware loss, ignore policy, and empirical detector-combination factor each have a measurable effect.

Sensitivity to the positive prior. We sweep the denominator that sets the nnPU prior π. Table 17 reports public mAP and the validity-head AUC (on the official-train validation split) for denominators $\{0.5, 0.6, 0.7, 0.85, 1.0\}$, i.e. $\pi \in \{0.29, 0.24, 0.21, 0.17, 0.15\}$ after the $[0.15, 0.45]$ clamp. Public mAP peaks at the production denominator 0.7 ($\pi \approx 0.21$) and falls by 0.003–0.004 on both sides, and the validity-head AUC peaks at the same point. The head is thus not brittle to the prior, and the 0.7 denominator ($\approx$ 70% annotation completeness) is empirically near-optimal.

7. Discussion

Frozen-detector fusion under PU labels and distribution shift. Positive-unlabeled annotations make task-specific detector fine-tuning risky, because unannotated positives can create false-negative gradients. The distribution shift between the official training image pools and MBARI M3 released-test images adds another risk, as fine-tuning on the official training split may damage pretrained marine-domain representations that transfer better to the released-test images. These two constraints motivate

Table 17
Sensitivity of the nnPU validity head to the prior denominator (which sets π). Public mAP peaks at the production value 0.7; the validity-head AUC is measured on the official-train validation split.

Denominator	π	Public mAP	Δ	Validity AUC
0.50	0.29	0.3170	−0.0040	0.731
0.60	0.24	0.3169	−0.0041	0.709
0.70 (ours)	0.21	**0.3210**	—	**0.870**
0.85	0.17	0.3178	−0.0032	0.854
1.00	0.15	0.3180	−0.0030	0.838

our main design of keeping the MBARI/FathomNet detectors frozen and training only the PU-aware cluster-level fusion scorer that combines their outputs. The full-pipeline ablations in Sec. 6.3 show that this PU-aware formulation is not just compatible with the task but important for the final method.

Duplicate-aware training recipe for the learned scorer. The duplicate head is not used at inference, but the duplicate-aware training recipe remains useful. The retrain in Sec. 6.2 removes duplicate supervision, the duplicate factor in the training-time score combiner, and same-class duplicate ranking pairs together, and loses 0.0049 public mAP. We therefore interpret the result as evidence for the duplicate-aware training recipe as a whole, not for any single duplicate-related term. Because the three heads use disjoint trunks, this gain is not shared-trunk regularization; instead the duplicate terms shape the validity and switch heads through two output-level couplings. The training-time score combiner multiplies the validity, switch, and duplicate factors into a single score, on which the score-consistency regularizer and the pairwise-ranking loss act; and the pairwise-ranking loss includes same-class duplicate pairs. Both constrain the combined score—and hence the validity and switch outputs—even though the duplicate head is dropped at inference. The per-component factorial (Table 15) is consistent with this output-level account: the effect is recipe-level, not attributable to a single term.

The recipe is also PU-aware. Same-class duplicate pairs are mined from local same-class IoU structure plus official-positive `anchor_match`/`neighbor_match` evidence, so labels are only generated where positive supervision is reliable. The duplicate-aware label-generation pipeline therefore uses official ground truth only through positive matches, not by treating unmatched anchors as negatives.

Different scale behavior for narrow-domain add-on detectors. The most counterintuitive result is that the narrow-domain VME detector is most useful at its best tested single inference scale, while the broad-domain detectors benefit from three-scale inference. We interpret this as a redundancy effect. VME covers only three retained classes and is merged into the benthic detector slot, so multi-scale VME can create same-class duplicate anchors rather than new independent detector support. This suggests a useful design rule for similar ensembles. When a narrow detector is added to a broader detector that already covers the same classes, more scales may increase duplicate structure more than complementary evidence.

Negative results matter. Two of our most informative ablations are negative. Retraining with the narrow-domain VME detector (Sec. 5.5) and stripping the entire duplicate-aware training recipe (Sec. 6.2) both regress. Reporting these explicitly strengthens the justification of the chosen design. We also tried several output-only score adjustments, re-ranking, and recalibration extensions on the step-6 baseline. These did not improve the public leaderboard score, so we keep the paper focused on the failure modes that directly explain the final architecture.

Reusability and transfer. The method itself—cluster-level late fusion with a PU-aware scorer—is general, but several components are fitted to this specific detector set. The empirical detector-combination factor and the bounded reliability reweighting are estimated from official-training match rates, and the scorer is trained on a cluster cache built from these detectors at these scales. Re-targeting

the pipeline to a new detector set is therefore asymmetric in cost: the detector-combination factor and reliability reweighting re-estimate *automatically* from any new set's training match rates (no manual tuning), the scorer needs *one* retraining from the official positive-unlabeled labels, and only the label mapping is *manual*. The pipeline already shows that the framework accommodates a new detector through this machinery: the narrow-domain VME detector is a different-domain add-on integrated via the existing benthic slot and the same combination and reliability estimation, and it helps at its best single inference scale. What remains untested is transfer to a different category taxonomy and a substantially different detector set; we leave a full cross-taxonomy study to future work.

Inference cost. On a single NVIDIA RTX 5090, end-to-end inference costs 444 ms/image (10.5 min for the 1425-image released-test set; Table 18). The frozen-detector forward passes dominate (416 ms/image across the ten passes), while same-class clustering and the dual-head scorer add only 28 ms/image on CPU. Table 8 exposes a compute/accuracy knob for deployment: dropping the broad detectors from three scales to two removes the 960-px pass (≈ 117 ms/image, ≈ 28% of the detector cost) for a 0.0036 mAP reduction.

Table 18
Per-component inference cost on one NVIDIA RTX 5090 (conf 0.01, batch 8, 50-image timing sample).

Component	Scales	ms / image
MBARI-315k detector	{640, 960, 1280}	122
benthic detector	{640, 960, 1280}	124
midwater detector	{640, 960, 1280}	122
VME detector	{1280}	48
Detectors (stage 1)		416
Clustering + dual-head scorer (stage 2, CPU)		28
Total		444

8. Future Work

We also explored BioCLIP 2.5 [21] and DINOv3 [22] after the frozen-detector fusion pipeline was stable. Simple post-hoc uses of these models were not stable in our setup. Directly changing detector labels, directly editing final scores, or strongly reranking detections after inference often hurt public-leaderboard performance. This should be read as a design constraint rather than as evidence that these models are unsuitable for the task.

The direction remains worth revisiting because some remaining errors are class confusions and label-space mismatches. Similar objects can compete across related task categories, and the 32 task categories do not always match the detector labels. BioCLIP 2.5 and DINOv3 may provide crop-to-text similarity scores or visual features for each box. However, these signals should be passed to a learned PU-aware scorer, not used as direct label-replacement or output-score rules.

A future version could keep the frozen detector proposal bank and change only the scorer. One cautious option is to keep the detector label for each cluster anchor, add one or two alternative labels from BioCLIP 2.5 or DINOv3, and let the scorer choose among them. Another option is a small attention-based scorer that compares detector scores, BioCLIP 2.5 scores, DINOv3 features, and nearby competing anchors at the same time. DINOv3 should also be treated first as a feature source for existing detector boxes rather than as a standalone box generator. A more speculative direction is PU teacher-student adaptation, where pseudo-labels are used only when the fused detector, BioCLIP 2.5, and DINOv3 agree strongly. All of these directions should keep the same PU rule that missing labels are not negative labels, and pseudo-labels should be used only with strong agreement across sources.

9. Conclusion

We presented a cluster-level late-fusion method over frozen MBARI/FathomNet YOLO-family detectors for FathomNetCLEF 2026. The method reaches public/private COCO-style bounding-box mAP@[0.50:0.95] scores of 0.3210/0.3042, ranking first on the final leaderboard. Instead of fine-tuning a detector on positive-unlabeled annotations, we keep the detectors frozen and train only a PU-aware cluster-level fusion scorer. This learned scorer is designed to avoid the assumption that unlabeled anchors are negatives.

The ablation study supports three main findings. First, combining complementary frozen detectors gives the largest early gain over a single-detector baseline. Second, PU-aware label handling, the empirical detector-combination factor, and duplicate-aware training are important in the full recipe. Third, the narrow-domain VME detector is most useful when added only at inference and run at its best tested single inference scale. These results suggest that cluster-level late fusion is a practical strategy for marine object detection when annotations are incomplete and MBARI M3 released-test images differ from the official training image pools.

Acknowledgments

We thank the FathomNetCLEF 2026 organizers, the FathomNet and MBARI teams, LifeCLEF, FGVC13, and Kaggle for providing the competition, evaluation infrastructure, dataset, annotations, and public detector checkpoints used in this work. We also acknowledge the broader FathomNet data-contributor and annotation communities for making this challenge possible. We thank the Miin Wu School of Computing, National Cheng Kung University, for providing computational resources for this work.

Declaration on Generative AI

During the preparation of this work, the authors used OpenAI GPT-5.5 for grammar and spelling checks and for translation into English. The authors also used OpenAI GPT-5.5 to assist in drafting the detector-to-task label-space correspondences and to perform a first-pass taxonomic audit of them (Appendix B); all label assignments are released for expert verification. The pipeline schematic in Figure 3 was designed and hand-drafted by the authors; OpenAI GPT-5.5 was used to re-render the authors' draft in a cleaner visual style. The authors also used OpenAI GPT-5.5 image tools to assist with panel-label annotation in Figure 1, with dashed-box visualization in Figure 2, and with text-label edits in Figure 4. After using these tools/services, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] M. J. Er, J. Chen, Y. Zhang, W. Gao, Research challenges, recent advances, and popular datasets in deep learning-based underwater marine object detection: A review, Sensors 23 (2023) 1990. URL: https://www.mdpi.com/1424-8220/23/4/1990. doi:10.3390/s23041990.

[2] K. Barnard, L. Chrobak, FathomNetCLEF2026 @ LifeCLEF & CVPR-FGVC, Kaggle competition and ImageCLEF/LifeCLEF task page, 2026. URL: https://www.imageclef.org/FathomNetCLEF2026, unpublished. Official challenge citation points to Kaggle; task details were accessed from the ImageCLEF/LifeCLEF page on 2026-05-11.

[3] K. Katija, E. Orenstein, B. Schlining, L. Lundsten, K. Barnard, G. Sainz, O. Boulais, M. Cromwell, E. Butler, B. Woodward, K. L. C. Bell, FathomNet: A global image database for enabling artificial intelligence in the ocean, Scientific Reports 12 (2022) 15914. URL: https://www.nature.com/articles/s41598-022-19939-2. doi:10.1038/s41598-022-19939-2.

[4] L. Picek, L. Adam, S. Kahl, R. Bossy, L. Chrobak, H. Goëau, K. Papafitsoros, H. Klinck, W.-P. Vellinga, R. Planqué, T. Denton, K. Barnard, C. Nédellec, L. Deléger, M. Courtin, G. Martellucci,

I. Moummad, F. Vinatier, P. Bonnet, A. Joly, Overview of LifeCLEF 2026: AI challenges for biodiversity understanding and ecosystem management, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2026.

[5] L. Chrobak, K. Barnard, Overview of FathomNetCLEF 2026: Positive-unlabeled object detection in marine images, in: Working Notes of CLEF 2026 - Conference and Labs of the Evaluation Forum, 2026.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: Common objects in context, in: Computer Vision – ECCV 2014, volume 8693 of *Lecture Notes in Computer Science*, Springer International Publishing, 2014, pp. 740–755. URL: https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48. doi:10.1007/978-3-319-10602-1_48.

[7] L. Chrobak, mAP50-95: Fathomnetclef 2026 evaluation notebook, https://www.kaggle.com/code/lauravchrobak/map50-95, 2026. Official evaluation notebook linked from the ImageCLEF FathomNetCLEF 2026 page. Accessed 2026-05-09.

[8] G. Patterson, J. Daniels, B. Woodward, K. Barnard, G. Sainz, L. Lundsten, K. Katija, FathomVerse: A community science dataset for ocean animal discovery, arXiv preprint arXiv:2412.01701, 2024. URL: https://arxiv.org/abs/2412.01701. doi:10.48550/arXiv.2412.01701.

[9] Monterey Bay Aquarium Research Institute, MBARI's video annotation and reference system (VARS), https://docs.mbari.org/vars/, n.d. Official documentation. Accessed 2026-05-09.

[10] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. Haque, S. M. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, P. Liang, WILDS: A benchmark of in-the-wild distribution shifts, in: Proceedings of the 38th International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 5637–5664. URL: https://proceedings.mlr.press/v139/koh21a.html.

[11] Y. Yang, K. J. Liang, L. Carin, Object detection as a positive-unlabeled problem, in: 31st British Machine Vision Conference (BMVC), 2020, pp. 1–14. URL: https://www.bmva-archive.org.uk/bmvc/2020/assets/papers/0264.pdf.

[12] R. Kiryo, G. Niu, M. C. du Plessis, M. Sugiyama, Positive-unlabeled learning with non-negative risk estimator, in: Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 1675–1685. URL: https://proceedings.neurips.cc/paper/2017/hash/7cce53cf90577442771720a370c3c723-Abstract.html.

[13] FathomNet, FathomNet/MBARI-315k-yolov8, https://huggingface.co/FathomNet/MBARI-315k-yolov8, n.d.. Hugging Face model repository; weight file: mbari_315k_yolov8.pt; accessed 2026-05-11.

[14] FathomNet, FathomNet/2025-MBARI-Benthic-Supercategory-Object-Detector, https://huggingface.co/FathomNet/2025-MBARI-Benthic-Supercategory-Object-Detector, n.d.. Hugging Face model repository; weight file: best.pt; accessed 2026-05-11.

[15] FathomNet, FathomNet/2025-MBARI-Midwater-Supercategory-Object-Detector, https://huggingface.co/FathomNet/2025-MBARI-Midwater-Supercategory-Object-Detector, n.d.. Hugging Face model repository; weight file: best.pt; accessed 2026-05-11.

[16] FathomNet, FathomNet/vulnerable-marine-ecosystems, https://huggingface.co/FathomNet/vulnerable-marine-ecosystems, n.d.. Hugging Face model repository; weight file: best.pt; accessed 2026-05-11.

[17] A. Oniśko, M. J. Druzdzel, H. Wasyluk, Learning Bayesian network parameters from small data sets: application of Noisy-OR gates, International Journal of Approximate Reasoning 27 (2001) 165–182. doi:10.1016/S0888-613X(01)00039-1.

[18] G. Jocher, J. Qiu, A. Chaurasia, Ultralytics YOLO, 2023. URL: https://github.com/ultralytics/ultralytics, official citation metadata from https://github.com/ultralytics/ultralytics/blob/main/CITATION.cff.

[19] FathomNet, FathomNet/megafishdetector, https://huggingface.co/FathomNet/megafishdetector, n.d. Hugging Face model repository for the public MegaFishDetector; accessed 2026-05-17.

[20] D. Yang, L. Cai, S. Jamieson, Y. Girdhar, Robot goes fishing: Rapid, high-resolution biological hotspot mapping in coral reefs with vision-guided autonomous underwater vehicles, arXiv preprint arXiv:2305.02330, 2023. URL: https://arxiv.org/abs/2305.02330. doi:10.48550/arXiv.2305.02330.

[21] J. Gu, S. Stevens, E. G. Campolongo, M. J. Thompson, N. Zhang, J. Wu, A. Kopanev, Z. Mai, A. E. White, J. Balhoff, W. M. Dahdul, D. Rubenstein, H. Lapp, T. Berger-Wolf, W.-L. Chao, Y. Su, BioCLIP 2.5 huge, Hugging Face model card, 2026. URL: https://huggingface.co/imageomics/bioclip-2.5-vith14.

[22] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, F. Massa, D. Haziza, L. Wehrstedt, J. Wang, T. Darcet, T. Moutakanni, L. Sentana, C. Roberts, A. Vedaldi, J. Tolan, J. Brandt, C. Couprie, J. Mairal, H. Jégou, P. Labatut, P. Bojanowski, DINOv3, arXiv preprint arXiv:2508.10104 (2025). URL: https://arxiv.org/abs/2508.10104. doi:10.48550/arXiv.2508.10104.

A. Full feature-vector enumeration

This appendix enumerates every feature in the 140-dimensional per-anchor vector summarized in Table 4: the anchor part (Table 19, 66 features), the competitor part (the same 66 features computed on the highest-scoring different-class anchor with box IoU ≥ 0.45), and the relation part (Table 20, 8 features). The feature builder is released in `anchor_features.py`.

Table 19
Anchor feature enumeration ($66 = 15 + 9 + 32 + 6 + 4$). The competitor part repeats these 66 features on the competitor anchor.

#	Feature	Definition
1	base cluster score	fused base score of the cluster anchor
2	raw noisy-OR	noisy-OR over raw per-detector scores
3	reweighted noisy-OR	noisy-OR after bounded reliability reweighting
4	detector count	distinct supporting detectors, divided by 3
5	scale count	distinct supporting inference scales, divided by 7
6	cluster size	raw detections merged, $\min(1, \text{size}/10)$
7	detector bonus	multi-detector bonus term
8	scale bonus	multi-scale bonus term
9	geometry	geometry multiplier
10	score mean (raw)	mean of raw detection scores in the cluster
11	score mean (reweighted)	mean of reweighted detection scores
12	score std (raw)	standard deviation of raw scores
13	score std (reweighted)	standard deviation of reweighted scores
14	log area	log normalized box area
15	log aspect	log box aspect ratio
16–24	per-detector support	for each detector in order (MBARI-315k, benthic, midwater): raw support, reweighted support, reliability factor
25–56	class one-hot	task-category one-hot (32 dimensions)
57–62	image context	log width, log height, luma mean, luma std, mean$(G - R)$, mean$(B - G)$
63–66	image regime	k-means (k=4) regime one-hot over features 57–62

B. MBARI-315k label-mapping audit

This appendix lists the corrections from the per-entry taxonomic audit of the 392 MBARI-315k concept-to-category assignments summarized in Table 2. We confirmed about 93%; Table 21 gives the 11 reclassifications and Table 22 the 5 non-target taxa. We flagged a further handful of borderline pelagic or

Table 20
Relation feature enumeration (8). Computed between the anchor and its competitor; all zero when no competitor exists.

#	Feature	Definition
1	bias	constant 1.0
2	IoU	anchor–competitor box IoU
3	base-score difference	anchor minus competitor base cluster score
4	noisy-OR difference	anchor minus competitor reweighted noisy-OR
5	same family	1 if both share a coarse taxonomic family (five hand-defined groups), else 0
6	detector Jaccard	Jaccard overlap of the two anchors' supporting-detector sets
7	area ratio	anchor/competitor cluster-area ratio, clipped to $[0.25, 4]$
8	center-distance ratio	center distance over the larger box diagonal, clipped to $[0, 4]$

higher-rank concepts for expert review. The audit is an LLM-assisted screen that we release for expert verification; we did not apply these corrections to the submitted system.

Table 21
Audit re-classifications (11): MBARI-315k concepts whose task-category assignment should change.

Concept	Mapping (current → corrected)	Reason
Gigantactis gargantua	anemone → bony fish	whipnose anglerfish
Branchiocerianthus imperator	anemone → hydroid	giant solitary hydroid
Heteropolypus ritteri	anemone → soft coral	mushroom soft coral (ex-Anthomastus)
Parastenella	sponge → sea fan	primnoid gorgonian octocoral
Calyptrophora	calycophoran siphonophore → sea fan	primnoid gorgonian octocoral
Craseoa	jelly → calycophoran siphonophore	Prayidae siphonophore
Praya dubia	physonect siphonophore → calycophoran siphonophore	Prayidae (calycophoran)
Prayidae	physonect siphonophore → calycophoran siphonophore	calycophoran family
Vogtia serrata	physonect siphonophore → calycophoran siphonophore	Hippopodiidae (calycophoran)
Thliptodon	sea snail → sea slug	shell-less gymnosome pteropod
Yoda	sea cucumber → benthic worm	acorn worm (Enteropneusta)

Table 22
Audit drops (5): concepts outside the 32-category space.

Concept	Current	Reason
Verum proximum	barnacle	not a recognized taxon (likely corrupt entry)
Smithsonius dorothea	jelly	bryozoan, not a cnidarian
Tuscarilla similis	calycophoran siphonophore	phaeodarian protist
Tuscaroridae	calycophoran siphonophore	phaeodarian protist family
Brachiopoda	bivalve	lamp shells (separate phylum)