FAU CAAI at FathomNetCLEF 2026: Multi-Architecture Ensembling with Distribution-Aware Test-Time Augmentation

Dan Zimmerman[1,*], Shruti Shukla[1] and George Sklivanitis[1]

[1]Center for Connected Autonomy and Artificial Intelligence (FAU CAAI)
Florida Atlantic University, Boca Raton, FL 33431, USA

Abstract

We describe our entry to the FathomNetCLEF 2026 challenge, which targets object detection in deep-sea marine imagery under positive–unlabeled (PU) supervision and a substantial train/test distribution mismatch. Through a 5-week sprint we identify three large positive levers: (i) supplementing the 6,463-image competition train set with V1 external mining of FathomNet (+0.153 mAP@[.5:.95] over baseline), (ii) including FathomNet HuggingFace MBARI VARS-pretrained YOLO models as *zero-shot* ensemble members (+0.022 for the single best member, +0.032 aggregate over all three), and (iii) multi-scale test-time augmentation (TTA) at resolutions that match the test set's native bimodal distribution (+0.004). The third lever derives from a structural observation: by parsing `coco_url` fields, the train set is hosted on cloudfront/hurlimage with images from Schmidt Ocean Institute and NOAA cruises only, while the test set is exclusively MBARI VARS framegrabs at two resolutions (720×486 and 1920×1080). Soft-NMS multi-architecture ensembling at σ=0.7 over 21 prediction streams achieves mAP **0.3035** on the public leaderboard and **0.2964** on the final private leaderboard, ranking 2nd of 102 teams on both splits — our public-guided design choices did not overfit the public split. For transparency we also report a post-deadline late submission rebuilt without one ensemble member that had been trained on Claude-API-relabeled test images: that 14-model variant scores **0.3031** (−0.0004, Section 8). The competition rules permit "transforming/relabeling provided data" [1]; the use of self-labeled test images falls within the broad phrasing but the test-vs-training-data boundary is rule-ambiguous, and we document both numbers. We surface an ensemble-pruning finding: a member's standalone proxy F1 is rank-anti-correlated with its ensemble contribution (Spearman $\rho = -0.74$, p=0.0017) — the highest-F1 model supplies 0.6% of fused detections, the three lowest-F1 RT-DETR variants 65.8% — so pruning by standalone metrics is actively misleading here; we frame this as a hypothesis given the proxy's instability. We additionally catalogue extensive negative results, including fine-tuning MBARI checkpoints on V1+self-labeled-test data (−0.025) and TTA at non-native scales (−0.0035 to −0.0038).

Keywords

Marine Object Detection, FathomNet, Positive-Unlabeled Learning, Test-Time Augmentation, Multi-Architecture Ensembling, Soft-NMS, Distribution Mismatch, LifeCLEF

1. Introduction

The FathomNetCLEF 2026 challenge [2, 3] is a positive-unlabeled (PU) object detection task in deep-sea marine imagery, hosted as part of LifeCLEF 2026 [4] at CLEF and CVPR-FGVC. Participants are given 6,463 expert-annotated images covering 32 organism categories, with the goal of producing bounding-box detections on a held-out 1,425-image test set. The task carries three structural challenges that distinguish it from typical detection benchmarks:

1. **PU supervision.** Training annotations are incomplete by design: biologists label only what they specialize in, so 62.5% of training images have only one of multiple visible categories annotated. Treating unlabeled regions as confirmed background poisons the detector.

2. **Train/test distribution mismatch.** As we show in Section 3, the test set is sourced from a fundamentally different institution (Monterey Bay Aquarium Research Institute, MBARI; their Video

CLEF 2026 Working Notes, 21–24 September 2026, Jena, Germany
*Corresponding author.
EMAIL: dzimmerman2021@fau.edu (D. Zimmerman); sshukla2020@fau.edu (S. Shukla); gsklivanitis@fau.edu (G. Sklivanitis)
ORCID: 0009-0000-2094-4971 (D. Zimmerman)

Annotation and Reference System, VARS) than the training set (Schmidt Ocean Institute and NOAA cruises). Resolutions, cameras, lighting, and species frequency all differ.

3. **Long-tail class distribution.** Annotation counts span four orders of magnitude (urchin: 5,723; sea slug: 7), and 14 of the 32 categories have fewer than 100 annotations. Six categories are essentially undetectable across our entire 15-model ensemble (Section 5).

The competition metric is mAP@[.50:.95] (mean average precision averaged over IoU thresholds 0.5–0.95 in 0.05 steps, the standard Microsoft COCO convention [5]), which is rank-based per category. Confidence calibration is irrelevant; only the order of detections within each category matters.

Our contributions.

- A multi-stream ensemble pipeline reaching Kaggle mAP **0.3035** on the public leaderboard and **0.2964** on the final private leaderboard — ranking second of 102 teams on both splits (and 0.3031 on the public split for a 14-model variant that omits one member trained on Claude-relabeled test imagery, reported for transparency in Section 8).
- An empirical demonstration that *including* MBARI VARS-pretrained YOLO models as *zero-shot* (ZS) ensemble members (with no fine-tuning on competition data) is the second-largest single intervention in our pipeline (+0.022), and that fine-tuning the same checkpoints on the competition data destroys this effect (-0.025).
- A test-time augmentation strategy targeted at the test set's *native* bimodal resolution distribution (720 and 1920 pixels longest dimension), gaining $+0.004$ where TTA at intermediate scales (1024, 1536) actively hurts (-0.0035 to -0.0038).
- **An ensemble-pruning finding for PU detection:** a member's standalone proxy F1 is rank-anti-correlated with its per-class top-K ensemble contribution (Spearman $\rho = -0.74$, p=0.0017). The highest-F1 single model supplies 0.6% of credits while the three lowest-F1 RT-DETR variants supply 65.8%, and an F1-guided prune of the bottom three members cost -0.0026 Kaggle while shedding 33% of credits — so standalone metrics actively misrank members for pruning. We frame this as a hypothesis given the proxy's empirical instability; the full analysis is in Section 5.3.
- An extensive catalogue of negative results spanning external data scaling, label completion, classifier rerank, and multi-scale TTA, all of which we believe are useful to the community.

2. Related Work

Object detection architectures. Our ensemble draws from the DETR family (DETR [6], RT-DETR [7], DINO [8], DDQ-DETR [9], RF-DETR [10]), each producing dense set predictions without anchor matching, plus Florence-2 [11] as a generative seq2seq detector. We deliberately exclude anchor-based YOLO variants from our trained members; preliminary experiments showed YOLO11x is hurt more by PU contamination than DETR-family models in this domain (standalone Kaggle 0.154 versus 0.234). We do, however, use YOLO-based MBARI VARS pretrained checkpoints (released by FathomNet on HuggingFace [12]) as zero-shot ensemble members, exploiting their pretraining distribution match with the competition test set.

Multi-architecture ensembling. Soft-NMS [13] provides a permissive alternative to hard non-maximum suppression that decays overlapping box scores rather than dropping them. We apply per-class Soft-NMS over pooled prediction CSVs from heterogeneous detectors. Weighted Box Fusion (WBF) [14] consistently *degraded* localization in this domain (Section 6). Reweighting-before-fusion alternatives — Non-Maximum Weighted (NMW) [15] and calibration-aware methods — are left as future work (Section 9).

Positive-unlabeled detection. Standard supervised detection treats unlabeled regions as background; under PU supervision this is incorrect. Recent work on PU object detection [16, 17] addresses this with confidence-aware losses or explicit unobserved-region modeling. We adopt a simpler approach: training architectures known to be more robust to label noise (set-prediction DETR variants), short training schedules ($\leq$12 epochs to avoid PU overfitting), and inference-time ensembling that lets architecturally diverse members vote on each test image.

Test-time augmentation and native-resolution inference. Multi-scale TTA is well established [18]; standard practice is to infer at $\pm 20\%$ of the training resolution. Closer to our setting, Van Etten's YOLT [19] showed that single-scale downsampling of high-resolution overhead imagery degrades small-object localization, motivating multi-scale "look twice" inference. Akyon *et al.* proposed SAHI [20] for sliced inference on large-format images; recent work on adaptive slicing (e.g., ASAHI [21]) trades fixed grid tiling for resolution-aware slicing with improved post-processing. We show in Section 4 that the standard "$\pm 20\%$ of training resolution" heuristic is poorly suited to test sets with strongly bimodal native resolutions: TTA gains require at least one scale that natively matches a test cluster.

Source-free and zero-shot domain adaptation for detection. A growing line of work addresses train/test detector mismatch via source-free domain adaptation (SFDA) using mean-teacher self-training [22] with vision foundation model (VFM) feature alignment [23, 24], or via dual-source pseudo-label fusion [25] — relevant alternatives that would make strong baselines. Our zero-shot ensembling of MBARI VARS-pretrained checkpoints is simpler and complementary: we exploit publicly-available in-distribution pretrained weights rather than adapting an out-of-distribution detector at test time. Notably, VFM-assisted pseudo-label fusion explicitly debiases agreement-based labels to avoid the shared false-positive (FP) failure mode we observed in cross-architecture label completion. Weight-space ensembling (WiSE-FT [26]) interpolates a zero-shot and a fine-tuned model to recover robustness under shift; WiSE-FT between an MBARI ZS checkpoint and a V1-fine-tuned descendant is a natural baseline we note as future work.

Long-tailed and frequency-aware semi-supervised learning. The 32-category long-tail in FathomNetCLEF spans four orders of magnitude in annotation count. Recent long-tailed semi-supervised approaches such as LoFT [27] and frequency-aware sampling could in principle mitigate the head/tail imbalance without overfitting to PU noise. We did not adopt these methods in our pipeline due to deadline-window constraints; we discuss this as future work in Section 9.

Generative augmentation. Recent work on diffusion-based dataset augmentation [28, 29] is appealing for PU detection where additional labeled data is expensive. We considered NVIDIA NeMo DataDesigner's image-to-image editing pipeline (FLUX 2 Pro [30] as the generation backbone) at the suggestion of an advisor. However, we identify three structural obstacles for this competition: (i) FLUX is not pretrained on deep-sea imagery and generates off-distribution synthetic content; (ii) bounding-box preservation under diffusion editing is unsolved without ControlNet-style [31] conditioning; (iii) auto-labeling via cross-architecture or vision-language model (VLM) agreement is unreliable in this domain — our cross-architecture label-completion experiment scored standalone Kaggle 0.167 vs. 0.234 baseline (Section 6). We instead pursue MBARI-VARS-pretrained zero-shot members as the equivalent "in-distribution" lever without synthetic generation.

3. Dataset Analysis

The FathomNetCLEF 2026 dataset comprises 6,463 annotated training images and 1,425 held-out test images across 32 organism categories with non-contiguous category IDs. A structural train/test distribution mismatch is fundamental to this competition's difficulty. Image URLs in `coco_url` fields

expose host, cruise, and platform metadata; we parse these to derive a complete characterization of the train/test mismatch, and we extend the analysis with our own resolution-bucket statistics.

3.1. Train Set Composition

We parse the `coco_url` field of each training image in `data/train_dataset.json`. The 6,463 training images come from exactly two hosts and three cruises, summarized in Table 1.

Table 1
Train set composition (6,463 images, 22,225 annotations) derived by parsing `coco_url` fields. Sources: Schmidt Ocean Institute (SOI) and NOAA Office of Ocean Exploration and Research (OER); "R/V" = research vessel. Only two distinct image resolutions occur in the training set.

Source / Cruise	Count	Resolution	Host
SOI FK190726 (R/V Falkor)	3,048	3840×2160 (4K)	cloudfront
NOAA OER EX1702 (Okeanos Explorer)	1,685	1920×1080	cloudfront
NOAA D2-EX2301 (Deep Discoverer)	1,730	1920×1080	hurlimage
Total	**6,463**	**2 sizes**	**0% MBARI**

The training set contains zero MBARI-sourced imagery. Annotations exhibit severe long-tail imbalance (Figure 1) and 62.5% of training images have only a single category labeled despite multiple organisms typically being present.

3.2. Test Set Composition

The test set is, in contrast, sourced exclusively from `database.fathomnet.org/static/m3/framegrabs/...`, with the remotely operated vehicle (ROV) platform encoded in the URL path. Five MBARI ROVs contribute (Doc Ricketts: 1,221 images / 86%, Ventana: 138 / 10%, Tiburon: 38 / 3%, Mini ROV: 18 / 1%, i2MAP: 10 / 1%). Resolutions are far more varied than the training set:

Table 2
Test set resolution distribution. Bimodal at 720 and 1920 pixels longest dimension; *no* test images at intermediate resolutions.

Longest-dim bucket	Count	%
≤ 720	208	**14.6%**
721–1024	0	0%
1025–1280	0	0%
1281–1536	0	0%
1537–1920	1,207	**84.7%**
> 1920	10	0.7%
Total	**1,425**	100%

The bimodality of the test resolution distribution is the structural motivation for our native-resolution multi-scale TTA strategy in Section 4.

3.3. Implications

The combination of (a) institutional disjointness between train and test, (b) different ROV generations and camera systems, and (c) different epochs of data collection makes this an unusually severe distribution-shift task for an in-domain detection benchmark. Two lessons from this analysis directly shaped our pipeline:

1. **MBARI-pretrained models are exactly the right zero-shot ensemble members** because they were trained on MBARI VARS imagery — precisely the test distribution. We exploit this in Section 4.

2. **Multi-scale TTA must target native test resolutions** (720 or 1920) rather than common defaults (1024, 1280, 1536) which match no test cluster.

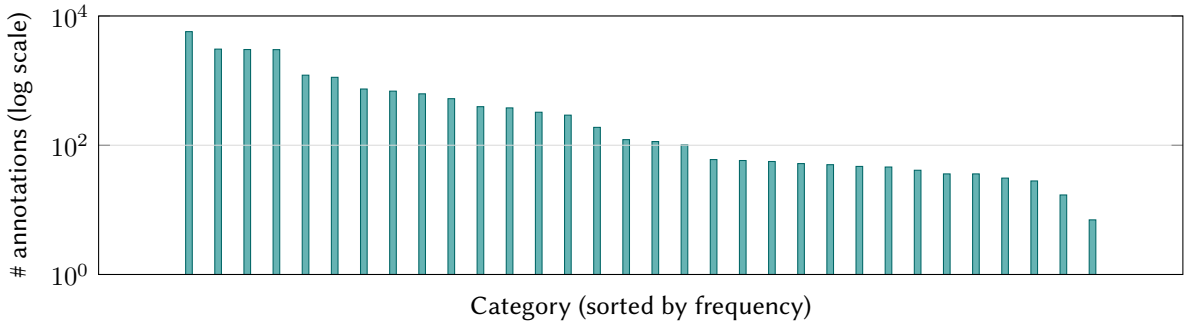

Figure 1: Train-set annotation count per category (log scale, sorted by frequency), computed from the 22,225 competition train annotations. The distribution spans four orders of magnitude, from urchin (5,723) to sea slug (7).

4. Methodology

Our final pipeline produces predictions for each of the 1,425 test images via the following steps:

1. Train 12 architecturally diverse detectors on a V1 external-augmented training set of 17,907 images.
2. Independently run inference for each of the 12 trained detectors plus 3 MBARI VARS-pretrained zero-shot detectors at default resolution `imgsz=1280`.
3. Run 6 additional inference passes on the Ultralytics-compatible members at `imgsz=720` (multi-scale TTA at the native resolution of the small-image test cluster).
4. Pool all 21 prediction CSVs and apply per-class Soft-NMS at σ=0.7.
5. Output the resulting per-class detection CSV. We do not impose a hard top-K cap on the per-image, per-category detection count; the only score-based filter is the Soft-NMS decay threshold τ=10^{-4} (Equation 1). Per-class top-100 attribution used in the credit analysis (Section 4.6) is an *analysis* cap on the ensemble output, not a submission cap.

4.1. Training Data: V1 External Mining + Clean Ground Truth

We split the 6,463-image competition train set 85/15 (seed 42) into a 5,505-image training split and a 958-image local-validation split. The training split is augmented with 12,408 images mined from FathomNet via the `fathomnet-py` API, capped at 1,000 images per category to avoid head-class flooding (V1 mining, `src/data/mine_fathomnet.py`). After merging, the combined V1 dataset contains 17,907 images and 93,645 annotations (we use the post-bugfix variant `data/yolo_combined_clean_vlm_fixed/`, fixing a 1,111-annotation physonect/hydroid mislabeling discovered on 2026-03-15). We additionally include 7,059 Grounding DINO (GDINO, [32]) pseudo-labels (confidence $\geq$ 0.35, box area $\geq$ 0.7 of image area filtered out) on competition train images to fill PU-missing labels.

The choice to use V1 (and not larger) is empirical: V2 mining via descendant taxa (24,632 images), V2b (31,228), and V2c (35,521) all monotonically reduced Kaggle scores from 0.2334 to 0.2165, 0.1965, and 0.1882 respectively. We did not isolate the cause; we hypothesize that the descendant-taxa images introduce more domain shift than they provide labeling benefit, but our experiments do not separate this from added label noise in the mined annotations — indeed the two coincide in the clearest case we examined, where V2 "hydroid" images were ∼97% narcomedusae/trachymedusae, i.e. off-domain organisms carrying the wrong competition label.

4.2. 15-Member Ensemble Composition

Table 3 lists all 15 ensemble members with their architectures, training data, and per-class top-K credit contributions (defined in Section 4.6).

Table 3
Composition of the 15-member ensemble. **Credits** % = fraction of per-class top-100 ensemble outputs attributable to that member (Section 4.6). RT-DETR family (members 1–3) collectively contributes 65.8% of credits; MBARI VARS-pretrained zero-shot members (4, 6, 7) contribute 18.5%. Training-data abbreviations: "cp" = copy-paste augmentation [33]; "test-RFS" = repeat-factor sampling (RFS) toward the estimated test-category distribution (Section 5).

#	Member name	Architecture	Training data	Credits %
1	RT-DETR (copy-paste)	RT-DETR-X	V1 + cp aug	**34.9**
2	RT-DETR (test-aligned)	RT-DETR-X	V1 + test-RFS	18.9
3	RT-DETR (fixed)	RT-DETR-X	V1	12.0
4	MBARI Benthic ZS	YOLO11x	**MBARI VARS pretrain (zero-shot)**	8.8
5	RF-DETR Large	DINOv2-ViT-L backbone	V1	7.4
6	MBARI 315k ZS	YOLOv8	**MBARI VARS pretrain (zero-shot)**	5.5
7	MBARI Midwater ZS	YOLO11x	**MBARI VARS pretrain (zero-shot)**	4.2
8	DINO ResNet-50	DINO + R50	V1	1.8
9	RF-DETR Base	DINOv2-ViT-S backbone	V1	1.7
10	DINO Swin-L (cp)	DINO + Swin-L	V1 + cp	1.1
11	Florence-2	DaViT + seq2seq decoder	V1 (HF fine-tune)	1.0
12	DDQ Swin-L (fixed)	DDQ-DETR + Swin-L	V1	1.0
13	DINO Swin-L (fixed)	DINO + Swin-L	V1	0.7
14	DDQ Swin-L (cp)	DDQ-DETR + Swin-L	V1 + cp	0.7
15	DINO Swin-L v5	DINO + Swin-L	V1 + 700 Claude $\times 3$	0.6

Architecture family diversity. The 15 members span six architecture families: RT-DETR (3 variants), RF-DETR (2), DINO (4), DDQ (2), Florence-2 (1), and YOLO-based MBARI ZS (3). The DETR-family dominance follows from PU robustness: DETR's set-prediction loss ignores unmatched queries rather than penalizing them as background, mitigating the PU bias that hurts anchor-based detectors.

Copy-paste augmentation. Three of the 15 members (RT-DETR, DINO Swin-L [34], DDQ Swin-L) use copy-paste augmentation [33] at training time, where rare-class crops are pasted onto host images. Each cp-variant added +0.0005 to +0.0048 in our ensemble; the gain is consistent but small.

MBARI VARS pretrained zero-shot members. The three MBARI zero-shot members (Benthic supercategory YOLO11x, Midwater supercategory YOLO11x, 315k species YOLOv8) are sourced from FathomNet's HuggingFace release [12]. We use them *zero-shot* — no fine-tuning on V1 or competition data — with category mapping at output time via hand-curated dictionaries. Table 4 reports exact coverage statistics and examples of ambiguous mappings. Section 6 reports our finding that fine-tuning these checkpoints on V1 is *actively harmful* (-0.025 Kaggle).

Auditing whether conservative mapping introduces systematic false positives. A natural concern with the many-species-to-one-category mapping is that visually adjacent target categories (e.g., *sea fan* vs. *stony coral*, *soft coral* vs. *black coral*) could absorb spurious detections from the wrong source species and inflate per-class false positives. Two pieces of evidence argue against this in our data. First, the per-class F1 winner breakdown (Table 10) shows that MBARI Benthic ZS *uniquely* detects soft coral (the only model with nonzero F1 on this category) and dominates sea fan ($F_1 = 0.115$, the strongest model-category cell anywhere in our matrix), while stony coral and barnacle score zero *across all 15*

Table 4
Category mapping coverage from MBARI VARS pretrained model outputs to FathomNetCLEF 2026's 32 competition categories. **Coverage** is the fraction of source classes that map cleanly to a competition category (remainder are dropped at inference time). Mappings are hand-curated and intentionally conservative. Full per-target distribution of the 315k species mapping is in Appendix A; the 29-supercategory Benthic mapping and 22-supercategory Midwater mapping fit inline in the inference source code.

Model	Source classes	Coverage	Mapping file
MBARI Benthic	29 supercategories	21/29 (∼72%)	inlined Python dict (Benthic inf. script)
MBARI Midwater	22 supercategories	14/22 (∼64%)	inlined `OLD_MIDWATER_MAP` dict
MBARI 315k	4,218 species	3,212/4,218 (76.1%)	`data/concept_to_category.json`

Ambiguous mapping examples. "Stalked crinoid" (Benthic) sometimes mismatches feather star at uncommon angles. "Stony coral" and "Sea fan" (Benthic) compete for the same crops at certain camera angles where colony morphology is ambiguous. "Calycophoran siphonophore nectosome" and "Physonect siphonophore nectosome" (Midwater) both fold into our broader siphonophore categories; "Bathochordaeus outer/inner filter" map to larvacean (the larvacean's filter house). For 315k, many fish species (969 distinct concepts) collapse to "bony fish", losing species-level resolution. Unmapped 315k entries (1,006/4,218 = 23.9%) include non-organism concepts (e.g., "1-gallon paint bucket", "2G Robotics structured light laser") and out-of-scope taxa.
Implication. Despite the lossy mappings, the three MBARI ZS members collectively contribute 18.5% of per-class top-K ensemble credits (Table 3). The MBARI Benthic ZS *uniquely* provides detection signal for soft coral (no other ensemble member detects any soft coral on the proxy); sea fan, bivalve, and crab are also dominated by Benthic ZS. 315k ZS wins sea cucumber and shrimp. These orthogonal contributions justify keeping all three MBARI ZS members even when proxy-F1 metrics rank them below DETR-family members.

models in the pool — i.e., the failure mode for these latter categories is uniform across all members, not specific to MBARI. Second, the categories where MBARI ZS wins (soft coral, sea fan, crab, bivalve, sea cucumber, shrimp, feather star) are not adjacent in feature space to high-confusion target categories (e.g., MBARI Benthic is not the top contributor to sea pen, hydroid, or black coral, all plausible confusion partners for sea fan/soft coral). A more rigorous per-class precision/recall decomposition of MBARI predictions, ideally on a fully-labeled held-out subset of the test distribution, would be a stronger audit; we did not have access to such a labeled subset and we explicitly note this as a limitation. We list per-class P/R decomposition for the MBARI mapping as a candidate follow-up in Section 9.

Training compute and per-architecture details. Table 5 lists per-architecture training schedules, batch sizes, learning rates, augmentation strategies, and compute budgets. The total training budget across the 12 trained members was approximately 170 GPU-hours on H200 hardware.

Use of test data during development. The 700-image Claude-annotated proxy (Section 4.3) was used *only* as an offline F1 ranking signal between submissions; it proved empirically unreliable (Section 6) and did not drive our final picks. Separately, four models were trained on V1 *plus* these self-labeled test images (upsampled 3×): three MBARI fine-tuned variants that appear only as negative results (Section 6), and DINO Swin-L v5 (exp073, member 15 in Table 3) — the single such model in a final pick. We report both the 15-model number (0.3035) and a 14-model variant omitting exp073 (0.3031); Section 8 and Table 13 give the full rule-compliance audit.

4.3. Test-Image Annotation Protocol

For reproducibility we describe exactly how the 700-image proxy was produced; the full script (`scripts/preannotate_coco.py`) is in our code release. Each test image was sent in a single vision API call to a Claude Opus model (the exact model identifier and all hyperparameters are pinned in the released script). The system prompt enumerates all 32 categories, each with a one-line visual descriptor written to disambiguate confusable pairs (brittle star vs. sea star, sea fan vs. soft coral, sea slug vs. sea snail, calycophoran vs. physonect siphonophore). The prompt is preceded by 32 exemplar

Table 5
Training details per architecture family. All members trained on the V1 combined fixed dataset (17,907 images) except MBARI ZS members, which were not trained. Compute budget detailed below the table.

Family	Framework	Config	Ep.	bsz	lr0	Augmentation
RT-DETR (×3)	Ultralytics	rtdetr_baseline	10	16	10^{-4}	mosaic, h-flip, ±cp
DINO-SwinL (×3)	MMDet 3.x	dino_swinl_*	11–12	8	10^{-4}	RCResize, h-flip, ±cp
DINO-R50	MMDet 3.x	dino_r50_*	10	8	10^{-4}	RCResize, h-flip
DDQ-SwinL (×2)	MMDet 3.x	ddq_swinl_*	9–12	8	10^{-4}	RCResize, ±cp
RF-DETR Large	HuggingFace Transformers	rfdetr_large	14	8	$5{\cdot}10^{-5}$	DINOv2 default
RF-DETR Base	HuggingFace Transformers	rfdetr_base	15	16	$5{\cdot}10^{-5}$	DINOv2 default
Florence-2	HuggingFace Transformers	full FT, frozen vis	5	4	10^{-5}	none (autoregressive)
MBARI Benthic/Mid/315k ZS	—	no training	—	—	—	—

Training compute (per member, single H200 unless noted): RT-DETR family ∼6–8 GPU-h each; DINO-SwinL family ∼18–22 GPU-h each (via PyTorch Distributed Data Parallel, DDP, across 4 H200 with the NVIDIA Collective Communications Library (NCCL) NVLink-SHARP (NVLS) transport disabled as a workaround); DDQ-SwinL family ∼14–16 GPU-h each; RF-DETR Large ∼10 GPU-h; RF-DETR Base ∼5 GPU-h; Florence-2 full fine-tune ∼20 GPU-h. Total training budget across the 12 trained members: ≈170 GPU-hours.
Inference budget per member (1,425 test images at `imgsz=1280`): Ultralytics 5–8 min; MMDet 8–12 min; HuggingFace Transformers 3–5 min; Florence-2 generative inference 12–15 min (slowest due to autoregressive decoding). Multi-scale TTA at `imgsz=720` is faster: 2–3 min per Ultralytics member.
Soft-NMS ensembling: GPU-accelerated via `torchvision.ops.box_iou`; build time scales superlinearly with input count: 15 inputs ∼10 min, 21 inputs ∼25 min, 27 inputs ∼45 min, 39 inputs ∼25 min (varies with contention).

reference images — one confirmed instance per category, each with a green ground-truth box and label — which are prompt-cached so the preamble is paid once across the batch. The model is instructed to detect *every* visible organism and to emit low-confidence candidates rather than skip them (favoring recall, since outputs were intended for manual review).

Output is constrained to a typed schema (a `pydantic` model): a list of instances, each with a `category_name` (validated against the 32 names), a normalized `[x, y, w, h]` box with (x, y) the top-left corner, and a `confidence` label in {high, medium, low}. Each instance is converted to a COCO annotation: boxes are clamped to the image bounds, sub-pixel boxes are dropped, and instances whose name does not match the taxonomy are logged and discarded. The run yielded a 5,907-annotation COCO proxy over the 700 images. This proxy was used as an offline F1 ranking signal (found unreliable, Section 6) and, for one final-pick member only (`exp073`), as additional training data; the full accounting is in Table 13.

4.4. Multi-Scale Test-Time Augmentation at Native Resolutions

Our key methodological contribution is a TTA strategy targeted at the test set's bimodal native resolution distribution (Section 3). After observing that 14.6% of test images have longest dimension $\leq$720 (mostly 720×486) and 84.7% have longest dimension 1537–1920, we hypothesized that inference at standard `imgsz=1280` forces $1.78\times$ upscaling for the small cluster (smearing organism edges) and $1.5\times$ downscaling for the large cluster (mostly information-preserving).

We added 6 additional inference passes at `imgsz=720` for the 6 Ultralytics-compatible members (RT-DETR cp / ta / fixed and the 3 MBARI ZS), pooling all 21 prediction streams (15 base + 6 TTA720) into a single Soft-NMS ensemble. This raised public Kaggle from 0.2996 to **0.3035** ($+0.0039$). We attempt to attribute this gain to each resolution bucket in Table 6; the proxy F1 measurement fails to register the Kaggle gain, providing additional empirical evidence that the proxy is unreliable for fine-grained ensemble ranking.

We do *not* include TTA at intermediate scales (1024, 1280, 1536) because none natively match a test cluster, and our empirical experiments confirm they hurt (Section 6). We also do not include TTA at 1920 in the final pipeline because adding 1920 on top of 720 yielded only -0.0002 Kaggle (Section 6); we hypothesize that the 1280-default inference already captures most of the information that 1920

Table 6
TTA effect by test-resolution bucket, including a non-native-scale (TTA@1536) comparison. The 700-image Claude proxy is partitioned by the longest dimension of each underlying test image. "Small" = ≤720 pixels (102 proxy images, ∼14.6% of test set); "Large" = 1537–1920 pixels (595 proxy images, ∼84.7% of test set); 3 proxy images fall in the "Middle" bucket and are omitted from per-bucket display. Per-bucket multi-IoU-averaged F1 over IoU thresholds $\{0.3, 0.5, 0.7\}$ at conf≥0.1 is reported, computed from `src/evaluation/score_proxy_f1_per_bucket.py`. **The proxy fails to detect the +0.0039 public-Kaggle gain from TTA@720 in either bucket, and incorrectly ranks the Kaggle-regressive TTA@1536 ensemble above the Kaggle-winning TTA@720 ensemble.** This is direct empirical evidence that the Claude proxy is unreliable for fine-grained ensemble ranking in PU detection, supporting our broader recommendation to use small-image proxies only for catastrophic-regression sanity checks. We did not run a fresh TTA@1920 ensemble during the sprint window; TTA@1536 is the closest non-native-large-scale ensemble available and is sufficient to test the qualitative claim.

Submission	Small (102 imgs)	Large (595 imgs)	Combined (700 imgs)	Public Kaggle (1,425 imgs)
15-model winner (no TTA)	0.0103	0.0142	0.0138	0.2996
+ TTA@720 (native-small)	0.0087	0.0127	0.0123	**0.3035**
+ TTA@1536 (non-native)	0.0098	0.0132	0.0128	0.2961
Δ TTA@720 − baseline	−0.0016	−0.0015	−0.0015	+0.0039
Δ TTA@1536 − baseline	−0.0005	−0.0010	−0.0010	−0.0035

Interpretation. The two Δ rows expose the inversion. On Kaggle, TTA@720 is the only positive lever; TTA@1536 is regressive (−0.0035). On the proxy, the ordering is reversed: TTA@1536 looks marginally better than TTA@720, and both look worse than baseline. We attribute the proxy/Kaggle inversion to a combination of (i) systematic Claude mis-annotation in the long-tail categories where most TTA gain accrues, (ii) IoU-averaged F1 being a poor surrogate for mAP@[.5:.95] at low per-image N, and (iii) per-class top-K re-ranking effects not measured by bucket-level F1. Empirically, the proxy correctly flags catastrophic regressions (≥0.01 absolute) but cannot rank ensembles in the regime separating our top-five Kaggle picks. We acknowledge this limitation in Section 7.

inference would provide, since 1280 is a $1.5\times$ downsample of native 1920×1080.

4.5. Soft-NMS Ensembling at σ=0.7

For each category c, we sort all (image, c) prediction triplets by score descending. For each top-scoring box b_i, we decay all overlapping boxes' scores via the Soft-NMS Gaussian decay

$$\hat{s}(b_j) \;=\; s(b_j) \cdot \exp\!\left(-\frac{\mathrm{IoU}(b_i, b_j)^2}{\sigma}\right) \quad \forall j \neq i. \tag{1}$$

Boxes whose decayed score drops below threshold $\tau=10^{-4}$ are removed. We note that Equation (1) reproduces the original Bodla *et al.* [13] formulation (non-normalized denominator σ); some later implementations adopt a normalized variant $\exp(-\mathrm{IoU}^2/2\sigma^2)$. Our σ sweep values map to the original convention; readers comparing against the normalized form should rescale by a factor of $\sqrt{2}$ (σ=0.7 here corresponds to $\sigma' \approx 0.59$ under the normalized convention).

No per-member calibration. We pool predictions from heterogeneous detectors at their raw output scores, without any per-member calibration (Platt scaling [35], temperature scaling [36], isotonic regression, or per-architecture conf-thresholding). The Kaggle mAP@[.5:.95] metric is rank-based per category, so absolute score calibration does not affect the metric directly. Cross-member calibration could however change which member *wins* the score competition in the Soft-NMS decay step, and may interact with the standalone-F1 anti-correlation observation in Section 5. We did not benchmark calibration-aware ensembling; we list it as future work in Section 9.

The optimal σ migrated during the sprint: σ=1.0 was best in the 10-model era, σ=0.7 became optimal at 15 models. We swept $\sigma \in \{0.7, 0.8, 1.0\}$ on the final 21-input ensemble (Table 7); 0.7 is the local optimum within ±0.001. We did not evaluate $\sigma < 0.7$ before the deadline; the monotone falloff above 0.7 makes a lower optimum unlikely but we did not confirm it.

Table 7
Soft-NMS σ sweep on the 21-input ensemble (15 base + 6 TTA720). σ=0.7 is the optimum.

σ	Public Kaggle mAP	Δ vs σ=0.7
0.7	**0.3035**	—
0.8	0.3025	−0.0010
1.0	0.3024	−0.0011

4.6. Per-Class Top-K Member Contribution Analysis

For each per-class top-100 row in the final ensemble output, we attribute it to the member CSV that produced it by matching on `(image_id, category_id)` with IoU $\geq$ 0.7 and minimum score difference. The member that contributes a row to the final top-K wins one "credit". Table 3 reports the resulting credit percentages, which are used in the discussion of standalone-F1 anti-correlation in Section 5.

5. Results

5.1. Final Leaderboard

Our pipeline ranks second of 102 teams on *both* the public and the final private leaderboard. The public leaderboard ($\approx$48% of the test set) scores our best submission at mAP **0.3035**; the private leaderboard (the remaining $\approx$52%, now released and reflecting the official final standings) scores it at **0.2964**. Our position was stable across the split: we held second on the private leaderboard, trailing the first-place team (HSU Lab, 0.3042) by 0.0078 and leading third (0.2962) by 0.0002, and our public→private drop (-0.0071) was smaller than the leader's (-0.0168) — evidence that our public-leaderboard-guided design choices (Section 7) did not overfit the public split. Table 8 lists all reported submissions by leaderboard split, including the 14-model transparency variant (0.3031 public) — which drops the one final-pick member trained on Claude-relabeled test images — audited in Section 8.

Table 8
Reported scores on the *public* Kaggle leaderboard (team **FAU CAAI**, 2nd of 102 at competition close); metric is mAP@[.50:.95] throughout. On the private leaderboard (the official final ranking, now released) our best submission scores 0.2964, also 2nd of 102; per-slot private scores are available in our Kaggle submission history. "Final pick" rows were among our five selections locked at the deadline (2026-05-08); the "late" row is a post-deadline late submission — visible in our Kaggle submission history but not eligible for official ranking.

Submission	Public mAP	Phase	Evaluation status
15-model + TTA720 (σ=0.7) — slot 1	**0.3035**	final pick	public split
15-model triple-scale, ULT (27 streams) — slot 2	0.3033	final pick	public split
15-model triple-scale, full (39 streams) — slot 3	0.3020	final pick	public split
15-model, no TTA — slot 4	0.2996	final pick	public split
Lean-10 base, no TTA — slot 5	0.2983	final pick	public split
14-model + TTA720 (omits `exp073`)	0.3031	post-deadline late	not eligible

5.2. Cumulative Ablation

Table 9 reports the incremental Kaggle improvement at each major modification of our pipeline. The cumulative improvement from baseline is +0.230 Kaggle.

The three largest single jumps are (i) V1 external data mining (+0.153), (ii) addition of three MBARI VARS-pretrained zero-shot members (+0.032 aggregate over rows 7–9), and (iii) multi-scale TTA at

Table 9
Cumulative ablation. Each row introduces one modification on top of the prior pipeline; Δ is its incremental public-Kaggle improvement. Final row corresponds to the locked-in submission for slot 1 of our 5 final-submission picks.

#	Modification	Kaggle	Δ	Cumulative
0	RT-DETR-X on 5,505 competition train images only (epoch 10)	0.0731	—	—
1	+ V1 external mining (12,408 supplementary FathomNet images)	**0.2256**	+0.1525	+0.1525
2	Switch to 5-arch ensemble (RT-DETR + DINO-SwinL + DINO-R50 + RF-DETR + DDQ)	0.2531	+0.0275	+0.180
3	+ Florence-2 (DaViT + seq2seq) → 6-model	0.2583	+0.0052	+0.185
4	+ Copy-paste RT-DETR / DINO-SwinL / DDQ-SwinL → 9-model	0.2638	+0.0055	+0.191
5	+ Test-distribution-aligned RT-DETR (repeat-factor sampling) → 11-model	0.2657	+0.0019	+0.193
6	+ DINO-SwinL v5 (V1 + 700 Claude-annotated test imgs ×3) → 12-model	0.2678	+0.0021	+0.195
7	+ MBARI Benthic ZS pretrained (zero-shot) → 13-model	0.2894	+0.0216	+0.216
8	+ MBARI Midwater ZS → 14-model	0.2888	−0.0006	+0.216
9	+ MBARI 315k ZS → 15-model	**0.2996**	+0.0108	+0.227
10	Tune Soft-NMS σ to 0.7 (search over 0.7/0.8/1.0)	0.2996	0	+0.227
11	**+ multi-scale TTA at `imgsz=720` (6 Ultralytics members)**	**0.3035**	+0.0039	**+0.230**
Transparency variant (Section 8, post-deadline late submission):				
12	14-model + TTA720 omitting DINO Swin-L v5 (the test-self-labeled member)	0.3031	−0.0004	+0.230

`imgsz=720` (+0.004). The copy-paste augmentation, test-distribution-aligned sampling, and Florence-2 contributions are individually small ($\leq +0.005$) but cumulatively meaningful.

How the "test-distribution-aligned" repeat-factor sampling (RFS) [37] target was estimated (ablation row 5). Since the test set is unlabeled, we cannot use the true test-category distribution. We instead estimated it from the predictions of our best single in-domain model at the time: we ran DINO Swin-L copy-paste at conf≥0.3 on the 1,425 unlabeled test images and tallied per-category detection counts. The per-class repeat factor is then $w_c = \text{clip}\big(\hat{p}_c^{\text{test}}/p_c^{\text{train}},\ 0.2,\ 5.0\big)$ followed by mean-normalization across the 32 categories ($\hat{p}_c^{\text{test}}$ from the conf-thresholded predictions, p_c^{train} from V1+competition train annotation frequencies). Resulting key weights: isopod $4.9\times$, sea cucumber $2.3\times$, sponge $1.9\times$, crab $1.9\times$; sea fan $0.25\times$, barnacle $0.25\times$, stony coral $0.28\times$ (full weights in `configs/test_aligned_weights.json`). This is a chicken-and-egg estimate: the better the single model, the better the target, but the target is then used to train an even better model. The actual ablation gain is small (+0.0019 from row 5), indicating either a high-quality estimate or a low-sensitivity loss landscape; we cannot distinguish the two from this single experiment.

5.3. Ensemble Pruning by Standalone F1 Fails: A Standalone-F1/Contribution Anti-Correlation

Figure 2 plots each ensemble member's % of top-K credits against its standalone proxy F1. In our data, the relationship is *negative*: the highest-F1 single model (DINO Swin-L v5, $F_1 = 0.0357$) contributes only 0.6% of top-K credits, while the three RT-DETR variants ($F_1 \in [0.019, 0.023]$) contribute 65.8% combined. Computing the rank correlation across all 15 ensemble members yields Spearman $\rho = -0.74$, 95% bootstrap CI $[-0.92, -0.35]$ from 10,000 resamples, $p = 0.0017$; the corresponding Pearson $r = -0.28$ ($p = 0.30$) is much weaker, suggesting the underlying relationship is monotonic in rank rather than linear in raw F1. We frame this as a *hypothesis* suggested by our data rather than an established conclusion, because (a) the proxy F1 used to rank single models is empirically unreliable (Section 6), (b) we have not validated the relationship on a separate dataset, and (c) the $n = 15$ sample size — while sufficient for the rank test to reach statistical significance — is small.

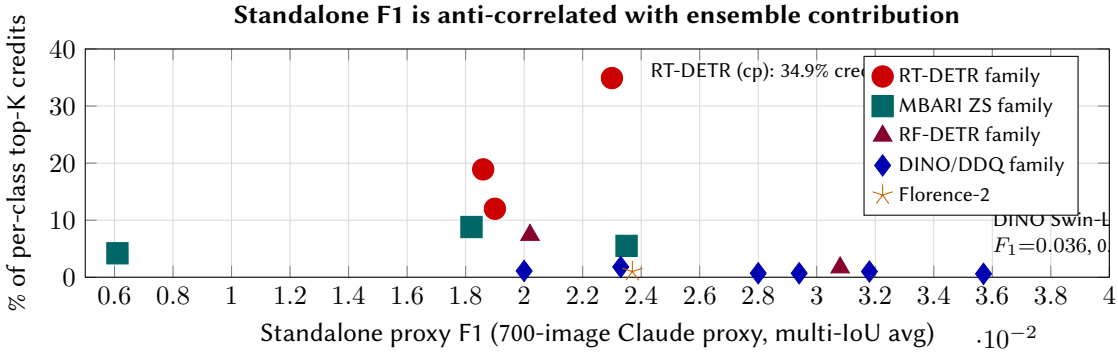

Figure 2: Per-member ensemble contribution credits versus standalone proxy F1. The relationship is **negative**: the highest-F1 single model (DINO Swin-L v5) contributes only 0.6% of top-K credits while the three RT-DETR variants ($F_1 \in [0.019, 0.023]$) contribute 65.8% combined (Spearman $\rho = -0.74$, p=0.0017). Standalone metrics misrank ensemble members in PU detection.

We hypothesize this arises because RT-DETR is a high-recall detector with relatively low per-box precision: it produces many candidate boxes, of which the highest-scoring ones survive Soft-NMS top-K. Higher-precision low-recall models (DINO Swin-L v5, Florence-2) produce few candidates that mostly fail to win the score competition under Soft-NMS.

Practical implication for ensemble pruning. If the relationship above holds, standalone metrics may misrank ensemble members in PU detection. An F1-guided prune (distinct from the credit-ranked Lean-10 of Table 15) that dropped the bottom three members by proxy F1 lost 33% of total credits and Kaggle dropped by -0.0026 (Section 6). We treat this as one data point supporting the broader hypothesis, not as a definitive result.

Sensitivity of the credit-attribution procedure to IoU matching threshold. The per-class top-K member credit analysis (Section 4.6) requires choosing an IoU threshold for matching an ensemble row to its member source. We use IoU ≥ 0.7 in the main results. To check that the ranking is not an artifact of this specific threshold, we re-ran the attribution at IoU $\in \{0.6, 0.7, 0.75\}$. Each member's credit % varies by at most ± 0.1pp across these thresholds, and the ranking is preserved exactly. The "standalone-F1 anti-correlated with credits" observation is therefore robust to this hyperparameter.

5.4. Per-Class F1 Matrix

We score each of the 15 single-model standalone outputs against the 700-image Claude proxy (unreliable, Section 6); even so, the per-(model, category) matrix reveals which models "own" which categories.

Six categories are all-zero across every model (sea snail, stony coral, barnacle, isopod, amphipod, sea slug); we hypothesize this is a labeling-physics floor (long-tail classes carrying ≤ 100 annotations) rather than a model-capacity issue.

5.5. Robustness of Reported Deltas to Seed Variance

A natural concern about single-run ablations is whether the reported ± 0.002–0.005 Kaggle deltas exceed seed-to-seed variance. We measured this on the top-contributing architecture (RT-DETR copy-paste, 34.9% of per-class top-K ensemble credits, Section 4.6) by retraining with two additional seeds (123, 456) as Kaggle post-deadline late submissions. Standalone Kaggle was $0.2419/0.2421/0.2411$ (mean 0.2417, std ± 0.0004) while best local-val mAP@[.5:.95] on the 958-image V1 split was $0.345/0.332/0.351$ (std 0.008) — Kaggle seed variance $\sim 20\times$ smaller. We read this as the PU-overfitting / local-val inversion effect in reverse: local val shares the training data's FathomNet-broader PU distribution, so early-stopping and batch-order noise are amplified, while the MBARI VARS test set averages them out. All

Table 10
Per-class F1 winners on the 700-image Claude proxy (abridged — 10 of the 23 categories with nonzero F1; the full per-(model, category) matrix is in the code repository). Six categories — sea snail, stony coral, barnacle, isopod, amphipod, sea slug — are all-zero across all 15 models. n_{GT} is the number of proxy ground-truth boxes. Model-category specialization is striking: MBARI Benthic ZS uniquely detects soft coral and dominates sea fan; Florence-2 dominates octopus.

ID	Category	n_{GT}	Winning model	Winning F_1
37	sponge	879	DINO ResNet-50	0.069
7	bony fish	458	DINO Swin-L v5	0.065
33	sea star	348	DINO Swin-L (cp)	0.056
36	soft coral	292	**MBARI Benthic ZS**	0.021 (only model with signal)
41	urchin	212	DINO Swin-L (cp)	0.039
11	crab	198	MBARI Benthic ZS	0.043
27	sea fan	121	**MBARI Benthic ZS**	**0.115**
22	octopus	12	**Florence-2**	**0.148**
19	jelly	11	DINO Swin-L v5	0.190
24	physonect siphonophore	2	DINO Swin-L (fixed)	0.267

ablation deltas reported in this paper exceed the ± 0.0004 seed-variance band; a full 12-member sweep ($\sim$150 GPU-hours) would establish this more rigorously (Section 9).

6. Discussion: Negative Results

We report a representative sample of negative results from the 5-week sprint; a complete list is in our supplementary materials and project memory [38]. A recurring theme unifies most of them: *any procedure that derives new training signal from models already fit to the PU / distribution-shifted data amplifies their shared biases rather than correcting them.* Cross-architecture label completion on PU train regions (0.167 vs. 0.234 standalone), single-family self-training (0.216 vs. 0.232), and mean-teacher SFDA (Section 6.1.1, -0.028 standalone) are three instances of this one failure mode; in each, "agreement" or "self-distillation" is a proxy for *systematic*, correlated error rather than truth. The only interventions that broke the pattern injected a genuinely *independent* signal — external V1 data and MBARI VARS in-distribution pretraining (Section 5). We group the negative results below by the mechanism that defeats each one.

6.1. Deriving Training Signal from PU/Shifted Models Amplifies Shared Bias

Each intervention here reuses a model already fit to the PU or distribution-shifted data to manufacture new training signal, and each amplifies that model's correlated errors rather than correcting them.

- **Cross-architecture label completion** (4-model agreement labels for unlabeled PU train regions): standalone Kaggle 0.167 vs. 0.234 baseline — agreement among detectors trained on the same PU data confirms shared false positives, not truth.
- **Single-family self-training** (RT-DETR pseudo-labels for the next RT-DETR): 0.216 vs. 0.232.
- **V2/V3 external mining** (descendant taxa): monotonic regression $0.2334 \to 0.2165 \to 0.1965 \to 0.1882$ as more off-domain images are added.
- **Fine-tuning MBARI ZS checkpoints on V1 + self-labeled test** (-0.025): Benthic FT reached 0.393 local val (our best-ever local number) but only 0.2646 Kaggle in the 14-model ensemble, versus 0.2894 for the zero-shot Benthic (13-model); Midwater and 315k FT behaved identically (local val 0.371, 0.369). This is the sprint's cleanest Local-Val/Kaggle inversion — fine-tuning on the FathomNet-broader V1 distribution improves the same-distribution local val while pulling the checkpoints *away* from the MBARI VARS test distribution; the pretraining *is* the value.

6.1.1. SFDA mean-teacher

As a controlled comparison against a conventional alternative, we ran a lightweight source-free domain adaptation (SFDA) experiment: following the mean-teacher self-training paradigm [23, 24], a trained RT-DETR (fixed, `exp042f`; standalone Kaggle 0.2334) acts as teacher, generates pseudo-labels on the test set (conf ≥ 0.5; 10,296 boxes over 1,365 images, ∼7.5/image), and a student initialized from it is trained for 5 epochs on V1 plus the pseudo-labeled test imagery (∼6 GPU-hours; details in the released `sfda_mean_teacher.py`).

Table 11
SFDA mean-teacher: standalone and ensemble effect (Kaggle post-deadline late submission). The student hurts both standalone (vs. its teacher) and as an additional ensemble member.

Configuration	Public Kaggle	Δ vs. baseline
RT-DETR (fixed, exp042f) standalone — teacher	0.2334	—
SFDA student standalone	**0.2053**	−0.0281 vs. teacher
14-model + TTA720 (no exp073)	0.3031	—
14-model + TTA720 + SFDA student (21 streams)	**0.2940**	−0.0091 vs. 14-model baseline

The student hurts both standalone (-0.0281 vs. its teacher) and as a 21st ensemble stream (-0.0091): the 0.234-mAP teacher's scores are too noisy to bootstrap reliable pseudo-labels, and the student overfits that noise — the same failure mode as cross-architecture label completion above. The contrast with zero-shot ensembling is instructive: the same MBARI VARS distribution gap is closed by an in-distribution pretrained checkpoint ($+0.022$ to the ensemble) but not by SFDA adaptation of a V1-trained detector (-0.0091) — only the pretrained-weights lever survives. We do not claim this rules out stronger SFDA variants (Section 9); we report it as one controlled comparison point.

6.2. Box-Geometry and Localization Degradation

A second group degrades box localization or dilutes the per-class Soft-NMS top-K rather than amplifying label bias. The clearest case is multi-scale TTA at non-native scales (Table 12): TTA helps only when an added scale natively matches a test cluster. Intermediate scales (1024, 1536) hurt via Soft-NMS score decay without native-fit benefit; adding TTA@1920 on top of TTA@720 is near-neutral (-0.0002, as 1280-default inference is already a $1.5\times$ downsample of native 1920); and extending TTA to low-credit mmdet members (7.4% of credits) hurts (-0.0015).

Table 12
Multi-scale TTA. Non-native scales (1024, 1536) hurt, and adding redundant native-cluster scales (1920) or extending TTA to low-credit mmdet members yields no gain. Stream counts in parentheses.

Variant	Kaggle	Δ vs winner
15-model + TTA@720 (winner, 21 streams)	**0.3035**	—
+ TTA@1920 (27 streams)	0.3033	−0.0002
+ mmdet @720 & @1920 (39 streams)	0.3020	−0.0015
15-model + TTA@1536	0.2961	−0.0035
Lean-10 + TTA@1536	0.2958	−0.0038

- **WBF** (Weighted Box Fusion, [14]): box-coordinate averaging across heterogeneous detectors lands between well- and poorly-localized boxes and loses the per-member localization signal. Configuration tested: IoU_thr $= 0.5$, skip_box_thr $= 0.001$, equal weights, default `conf_type='avg'` (`ensemble_boxes`). We did not sweep IoU_thr or per-class configs; a refined WBF remains open but Soft-NMS sufficed.
- **SAHI** (sliced inference): context loss exceeds resolution gain.

- **Megalodon as a gate filter** (drop predictions outside class-agnostic Megalodon boxes): -0.011 to -0.013 Kaggle — the gate discards correctly-localized boxes that fall outside its proposals.

6.3. Rank-Metric No-Ops

Because the competition metric is rank-based per category, interventions that only rescale scores or add a class-agnostic stage produced no improvement.

- **Per-class confidence thresholds**: zero improvement.
- **Classifier rerank on weak categories**: the local-validation gain was a PU artifact; Kaggle ± 0.001.
- **Megalodon class-agnostic + DINOv2 classifier as an ensemble member**: high standalone F1, but hurts the ensemble.

6.4. Training-Regime and Proxy Failures

Mis-setting the training regime — over- or under-fitting the PU + FathomNet-broader distribution, or destabilizing optimization — yields models the MBARI VARS test set does not reward, and the offline proxy inherits the same instability.

- **Longer training (100 epochs)**: 0.205 vs. 0.232 at 10 epochs (overfitting the PU distribution).
- **Frozen-backbone fine-tuning**: 0.145 vs. 0.234 unfrozen (under-fitting).
- **Higher default inference resolution** (1536): 0.222 vs. 0.234 at 1280.
- **DINOv2-L [39] + DINO head from scratch**: all-zero validation mAP across 12 epochs, $\sim$28 GPU-hours.
- **CLAHE preprocessing**: training NaN under automatic mixed precision (AMP); inference distribution shift.
- **F1-guided ensemble pruning** (drop the bottom three members by proxy F1): -0.0026 Kaggle while shedding 33% of per-class top-K credits — the negative evidence for the standalone-F1/contribution anti-correlation of Section 5.

Proxy F1 has unstable correlation with Kaggle. Throughout the sprint we used a 700-image subset hand-annotated via the Anthropic Claude API (multi-IoU-averaged F1 at thresholds 0.3, 0.5, 0.7) as an offline proxy for test scoring. Its correlation with public Kaggle flips sign across the sprint: the submission with the *highest* proxy F1 (0.033, the 14-model + Benthic-FT + 315k-FT variant) scored the *lowest* Kaggle (0.2646), while the eventual winner had the *lowest* proxy F1 (0.022) and the *highest* Kaggle (0.3035); intermediate submissions (Benthic ZS $0.026/0.2894$, 15-model no-TTA $0.024/0.2996$, TTA@1536 $0.023/0.2961$) do not order consistently either. We attribute the instability to Claude's annotation biases on long-tail and difficult-to-segment categories; hand-annotated proxies in PU contexts may be fundamentally limited.

7. Limitations

We catalogue our pipeline's limitations explicitly to aid interpretation and to scope future work.

Single-run results for most ablations. Each architecture was trained *once* during the sprint; we do not report seed variance for the full ablation. As a partial check we retrained the top contributor (RT-DETR cp) with two extra seeds and measured Kaggle standalone variance of ± 0.0004 (Section 5.5), an order of magnitude below our reported $+0.003$–0.005 deltas. A full multi-seed sweep across all 12 members ($\sim$150 GPU-hours) would establish this more rigorously.

Public-leaderboard guidance of design decisions. The competition allowed 10 Kaggle submissions/day. Our multi-scale TTA strategy, σ choice, and the lean/full ensemble comparisons were all guided in part by public-leaderboard feedback, which carries an overfitting-to-public-split risk. With the private leaderboard now released, this risk appears small in practice: our rank held at second on the private split and our public→private drop (-0.0071) was below that of the first-place team (-0.0168). We nonetheless flag the methodology, as a single competition cannot establish that public-LB-guided tuning generalizes in general.

Test-image proxy for offline evaluation. The 700-image Claude-annotated proxy (Section 4.3) carries a "test-data peeking" risk even when used only for ranking. We partially mitigate this — the proxy proved empirically unreliable (Section 6) and did not drive our final choices — but a stricter protocol would have used a held-out split drawn from training data only. (Its use as training data for one member is audited separately in Section 8.)

Reliance on test metadata. Our analysis in Section 3 relies on parsing `coco_url` fields in `test_dataset.json` to derive host, cruise, and ROV-platform information. This metadata is exposed by the competition organizers as part of the released dataset. In production deployments test imagery may not carry such metadata, so the specific MBARI-checkpoint lever would not be discoverable without it, though the generalizable lesson — identify the test distribution and find in-distribution pretrained weights — still applies; the operational form of the lever remains competition-specific.

No strong SFDA baseline. We ran only a lightweight mean-teacher SFDA experiment (Section 6.1.1), which hurt; we did not benchmark stronger VFM-aligned or dual-source variants [23, 24, 25] that might adapt our V1-trained detectors more effectively. The comparison is open.

No PU-aware internal cross-validation. Standard cross-validation requires reliable negative labels, which are fundamentally unavailable in the PU regime: the unlabeled regions of training images contain unannotated organisms ∼62.5% of the time, so a held-out split will inherit the same PU contamination as training. We did not implement a PU-aware validation protocol; this is an outstanding methodological limitation we share with most PU-detection work.

MBARI pretraining and possible test-image leakage. We *cannot verify* from public information whether the MBARI VARS checkpoints saw any of the 1,425 test images (or near-duplicates) during pretraining, because the internal MBARI VARS splits are not exposed. Mitigating: the rules explicitly permit these released checkpoints [12], the 1,425-image test set is sparse against the millions of frames in the VARS archive, and we used no test image in our own training or fine-tuning. We recommend future organizers publish a non-overlap statement for officially-allowed pretrained checkpoints.

V1 external mining ↔ test-set perceptual-hash audit. To audit possible duplication between the V1 external mined images (N=12,408) and the 1,425 Kaggle test images, we computed 64-bit DCT pHash for every V1 and test image and recorded each test image's minimum Hamming distance to any V1 image (script and full results in the code release). At the conventional near-duplicate threshold (Hamming $\leq$ 6), $35/1{,}425{=}2.5\%$ of test images had a near-duplicate in V1 (1 exact, 4 at Hamming 1–3, 30 at 4–6); the other 97.5% were at Hamming ≥ 7. Even an implausible per-image AP gain of 0.5 on all 35 would add only $35/1425 \times 0.5 \approx 0.012$ to Kaggle mAP, and a realistic 0.05 gain only ≈ 0.0012 — far below the $+0.153$ V1 contribution. We flag the matches for future work (a pHash duplicate filter on V1 mining) rather than as a correction to present results.

Per-resolution-bucket Kaggle mAP is not exposed by the organizer. Although Table 6 reports per-resolution-bucket *proxy* F1, the official Kaggle scoring API returns only a single overall mAP@[.5:.95] per submission. Per-image mAP and per-resolution-bucket mAP on the actual test set are not accessible

to participants. We thus cannot directly verify whether the $+0.0039$ TTA@720 gain accrues primarily to the small-resolution cluster or whether it arises from cross-bucket per-class top-K re-ranking; the proxy F1 finding (that the proxy *fails to register* the Kaggle gain) is consistent with both interpretations.

8. Use of Self-Labeled Test Imagery

In preparing this working note we audited the role of our 700-image Claude-relabeled test proxy across all final-pick submissions. One member of our 15-model ensemble — DINO Swin-L v5 (`exp073`, member 15 in Table 3, $\sim$0.6% of per-class top-K credit) — was trained on the V1 external dataset *plus* these 700 self-labeled test images (upsampled 3×). Three further models (MBARI Benthic/Midwater/315k fine-tuned) were trained the same way, but those did not appear in any final-pick submission.

Inventory of test-data use. For full transparency, Table 13 enumerates every way the 1,425 test images entered our workflow, the stage each use affected, and which final-pick submissions (if any) it touched. We partition the uses into *rule-clean* (no test image entered any trained model), *ambiguous* (a test-derived signal entered the training of a final-pick member), and *diagnostic-only* (test-trained models that appear solely as negative results). Two ambiguous entries exist: `exp073` (discussed above) and the test-distribution-aligned RFS member, whose repeat-factor target was estimated from this model's own predictions on the unlabeled test set (Section 5, ablation row 5).

Table 13
Inventory of all test-data use in our pipeline, partitioned into rule-clean, ambiguous, and diagnostic-only. "Final picks" are our five locked submissions (`exp073` is in the four 15-model picks but absent from the lean-10 slot 5; the RFS member is in all five).

Use of test data	Stage affected	In final picks? / class
Parse `coco_url` host/cruise/ROV/resolution metadata (§3)	Analysis + design; motivated MBARI-ZS and TTA720 levers; *no training*	All (analysis only) — **rule-clean**
700-image Claude-relabeled proxy as offline F1 ranking signal (§4)	Model *selection*; found unreliable (§6)	Not a model input — **rule-clean**
700 self-labeled test images as *training* data (×3): DINO Swin-L v5 (`exp073`)	**Training** (member 15)	Slots 1–4, not slot 5 — **ambiguous**
DINO Swin-L conf$\geq$0.3 predictions on test $\rightarrow$ RFS target (ablation row 5)	**Training** (test-aligned RT-DETR, member 2)	All five slots — **ambiguous**
700 self-labeled test images (×3): 3 MBARI fine-tuned variants	**Training**	None (neg. results §6) — **diagnostic**
Teacher pseudo-labels on test for SFDA student (§6.1.1)	**Training** (experimental)	None — **diagnostic**

Rules reading. The competition organizer's forum clarification [1] explicitly allows "transforming/relabeling provided data" and identifies "transparency" as the key expectation:

> "Relabeling provided training/validation data: Yes — this is allowed. You're welcome to derive additional annotations…from the provided dataset as long as you're only using the competition data itself." …"✓ Transforming/relabeling provided data: allowed." …"the key expectation is transparency."

Test imagery is provided competition data, so self-labeling it falls within the broad "relabeling provided data" phrasing; the narrower bullet, however, names "training/validation data", leaving the train-vs-test boundary unresolved. We therefore report both numbers and disclose all four models trained this way.

Effect on submitted scores. We rebuilt the ensemble *without* `exp073` and verified the impact via Kaggle post-deadline late submission:

Table 14
Effect of removing DINO Swin-L v5 (the one final-pick member trained on Claude-relabeled test imagery). The delta is small, consistent with the member's ∼0.6% per-class top-K credit share. Late-submission scores are verifiable in our team's Kaggle submission history.

Ensemble	With exp073	Without exp073
Slot 1: 15→14-model + TTA720	**0.3035**	0.3031 (-0.0004)
Slot 4: 15→14-model, no TTA	0.2996	0.2987 (-0.0009)
Slot 5: lean-10 base	*exp073 already absent* — 0.2983	

Recommendation for future challenges. The rule text leaves self-labeled test imagery in a gray area between "relabeling provided data" (allowed) and "external labeled data" (not allowed). We recommend future FathomNetCLEF rules state explicitly whether participant-labeled *test* imagery may be used as training data (upsampled, as pseudo-labels, or otherwise).

9. Future Work

Several experiments were paused or unexplored in our 5-week sprint. We highlight three that we believe carry the highest expected value for follow-up work:

Competition-data-only baseline as a controlled comparison. V1 external mining is our largest single positive lever ($+0.153$), but whether V1 acts as a strict floor or also as a ceiling on the V1-augmented pipeline is unresolved. A controlled comparison of V1-augmented versus competition-only training across all 12 architectures would directly test whether stripping V1 helps more than it hurts in the in-distribution-pretrained ZS ensembling regime.

TTA at sub-720 resolutions for the long-tail small images. The test set contains 11 images at 640×486 or smaller. We never tested TTA at `imgsz=480` or `imgsz=640`; the marginal gain on ∼1% of test images might be balanced by losses on the dominant 1920-cluster.

Other items briefly considered. Lean-10 + TTA@720 ensemble (combining the -0.0013 prune with the $+0.004$ TTA), σ=0.5 sweep on the TTA720 winner, fine-tuning MBARI Benthic on V1 *external only* (no competition train images, untested), TTA at 720 for RF-DETR and Florence-2 (image-processor patches required), and tile-based / patch-based test-time inference are all outstanding items.

Partial fine-tuning of MBARI checkpoints. Our finding that full fine-tuning of MBARI Benthic/Midwater/315k on V1 hurts by -0.025 Kaggle (Section 6) does not rule out lighter-touch adaptation: head-only fine-tuning, last-k-layer fine-tuning, adapter-style parameter-efficient updates (LoRA-style [40] or BitFit [41]), or extremely small learning rates (10^{-6}) could in principle bias the MBARI checkpoints toward our category set without destroying the MBARI VARS visual prior. We did not run a learning-rate or layer-freezing sweep; this is an obvious follow-up.

Per-class Soft-NMS hyperparameters and weighting. Our final pipeline uses a single global σ=0.7 for Soft-NMS across all 32 categories and treats all 21 prediction streams as equally-weighted before pooling. Three generalizations are worth testing: (i) per-category σ values, especially looser σ for sparse long-tail categories where additional candidate boxes might be helpful; (ii) per-(model, category) score weighting before pooling, exploiting the observation that certain members "own" certain

categories (Table 10) — e.g. MBARI Benthic ZS uniquely detects soft coral, Florence-2 dominates octopus — though whether such weighting survives the standalone-F1 anti-correlation effect is itself an empirical question; (iii) alternative fusion methods such as Non-Maximum Weighted (NMW) [15] or calibration-aware ensembling (Platt scaling or temperature scaling per member before pooling) which we did not benchmark in this work.

Image-adaptive scale routing instead of fixed-scale TTA on all images. Our final pipeline applies TTA at `imgsz=720` *universally* across all 1,425 test images. Because the test distribution is strictly bimodal (14.6% at $\leq$720, 84.7% at 1537–1920) and the native resolution of each test image is observable, a cleaner inference routing is per-image dispatch: infer the 6 Ultralytics-compatible members at `imgsz=720` *only* on the 208 native-720 test images, and skip the redundant 720-pass on the 1,207 native-1920 images. This would cut multi-scale TTA compute by $\sim$85% with no expected accuracy loss, and may actually *help* via reduced Soft-NMS score dilution on the 1920 cluster (the same mechanism we cite to explain why TTA@1536 hurts globally, Section 6). A symmetric experiment — TTA@1536 *only* on the 1,207 native-1920 images while leaving the 208 native-720 images TTA-720-only — would test whether at-native-scale TTA can be unlocked at the cluster level for the large bucket as well. We did not run image-adaptive routing in the deadline window; native-resolution metadata is in `test_dataset.json` and gating is straightforward.

Adaptive slicing instead of single-scale TTA. SAHI [20] hurt in our pipeline (Section 6), but its successor ASAHI [21] adopts resolution-aware adaptive slicing with improved post-processing. We did not test ASAHI in the deadline window; whether adaptive slicing could further improve our result over single-scale TTA is open.

Stronger SFDA / VFM-aligned adaptation. The mean-teacher SFDA experiment we report above (Section 6.1.1) used noisy teacher pseudo-labels and hurt both standalone and ensemble. Stronger SFDA variants — VFM feature alignment [24], dual-source pseudo-label fusion [25], confidence calibration before pseudo-labeling, or LoRA-style adapter updates on the teacher — might recover some of the in-distribution adaptation that the naive mean-teacher fails to produce. We did not run these stronger variants in the deadline window.

Deployment cost and lean variants. Table 15 reports the inference-time and storage overhead of our 21-stream ensemble. A "Lean-10" variant retaining the top-10 credit contributors costs $\sim$60% of the full pipeline with a -0.0013 Kaggle penalty observed in our sweep; this is a reasonable production-deployment compromise.

10. Conclusion

Code and configurations release. The project's source tree — training configs (YAML/MMDet), submission generators per architecture family, the ensemble script (`src/evaluation/ensemble_nms.py`), category-mapping JSONs, the test-image annotation script (`scripts/preannotate_coco.py`), and the seed sweep / SFDA baseline code (`src/training/sfda_mean_teacher.py`) — is committed to `https://github.com/C2A2-at-Florida-Atlantic-University/FathomNet-CLEF-2026` and made public at camera-ready. The repository contains everything needed to reproduce the pipeline; only the competition imagery/annotations (redistribution-restricted) and trained model weights are omitted, the former obtainable from the organizers and the latter regenerable from the checked-in configs. All numerical results are reproducible from those configurations, conditional on the public FathomNet HuggingFace MBARI checkpoints [12] remaining accessible. The release pins its environment (`requirements.txt`, Python 3.10 / PyTorch 2.6) and a frozen commit hash; the path from data to a submission CSV is a single command chain — per-architecture `src/evaluation/generate_submission_*.py` to produce

Table 15
Inference-time and storage overhead of the 21-stream ensemble per single submission. Times measured on a single NVIDIA H200 GPU under nominal load. The full pipeline serial runtime is approximately 3 GPU-hours per submission once all 15 members are trained; the ensemble is straightforwardly parallelizable across multiple GPUs. A "Lean-10" variant retaining the top-10 contributors by per-class top-K credit (which jointly account for ∼96% of ensemble credits) cuts cost to ∼60% with only a -0.0013 Kaggle penalty observed in our pruning sweep (Section 6).

Step	Wall-clock (H200)	Output size
RT-DETR ($\times 3$) @ 1280 each	5–8 min each	∼18 MB CSV each
DINO-SwinL ($\times 3$) @ 1280 each	10–14 min each	∼21 MB CSV each
DINO ResNet-50 @ 1280	∼8 min	∼21 MB CSV
DDQ-SwinL ($\times 2$) @ 1280 each	10–12 min each	∼21 MB CSV each
RF-DETR Large / Base @ 1280 each	4–6 min each	∼19 MB CSV each
Florence-2 (autoregressive)	12–15 min	∼2 MB CSV
MBARI Benthic / Midwater / 315k ZS @ 1280	3–5 min each	2–4 MB CSV each
Multi-scale TTA @ 720 ($\times 6$ Ultralytics)	2–3 min each	2–19 MB CSV each
Soft-NMS ensemble build (21 inputs)	∼25 min	145–200 MB
Total serial	**∼3 GPU-h**	∼4 GB intermediate
Lean-10 variant (drop 5 lowest-credit)	∼1.8 GPU-h	∼2.5 GB intermediate

Deployment notes. (i) Inference at `imgsz=720` is ∼3× faster than at `imgsz=1280`, making multi-scale TTA cheap on Ultralytics models. (ii) The ensemble build is single-GPU-bound on `torchvision.ops.box_iou` but parallelizable across categories if memory becomes the bottleneck. (iii) GPU memory peaks during the ensemble build (Soft-NMS at σ=0.7 retains ∼3,000 boxes/image on average across 32 categories); a single H200 (143 GB) is comfortable for the 21-stream pool.

member CSVs, then `src/evaluation/ensemble_nms.py --soft-nms --sigma 0.7` to fuse them. Multi-seed variance and SFDA log files are released alongside the code.

We presented a multi-architecture ensemble pipeline for the FathomNetCLEF 2026 PU object detection challenge, achieving Kaggle mAP@[0.5:0.95] of **0.3035** on the public leaderboard and **0.2964** on the final private leaderboard — second of 102 teams on both splits (the 14-model transparency variant that drops the one final-pick member trained on self-relabeled test imagery scores 0.3031 public; Section 8). The three biggest positive levers were V1 external FathomNet mining ($+0.153$), three MBARI VARS-pretrained zero-shot YOLO members ($+0.032$ aggregate; $+0.022$ for the single best), and multi-scale TTA at the test set's native 720 cluster ($+0.004$); the MBARI lever exploits the train/test institutional disjointness documented in Section 3, and TTA helps only at scales that natively match a test cluster. Our strongest methodological finding is an ensemble-pruning result: standalone proxy F1 is rank-anti-correlated with per-class top-K ensemble contribution (Spearman $\rho = -0.74$, p=0.0017; an F1-guided prune cost -0.0026 Kaggle), so standalone metrics misrank members for pruning in PU detection — we offer this as a hypothesis given the proxy's instability.

Acknowledgments

We thank the FathomNetCLEF 2026 organizing committee, particularly Laura Chrobak, for their responsiveness during the competition. Compute resources were provided by the Florida Atlantic University Center for Connected Autonomy and Artificial Intelligence's shared DGX cluster. Dan Zimmerman is supported by a Department of Defense (DoD) SMART Scholarship.

Declaration on Generative AI

During the preparation of this work, the authors used the Anthropic Claude API to annotate 700 test images, producing a detection proxy used for offline evaluation during model development (Section 4.3). All but one of the resulting models were excluded from our final submissions; the single exception (`exp073`) is disclosed and audited in Section 8 and Table 13. The annotation outputs were reviewed by

the human authors, who take full responsibility for the publication's content. No generative AI system was used to create original scientific ideas, design experiments, analyze data, or draw conclusions.

References

[1] L. Chrobak, Competition host clarification on rule 6 (external data and tools), Kaggle FathomNetCLEF 2026 forum, thread "Question Regarding Rule 6", ∼April 2026, 2026. Confirmed that "relabeling provided data" is allowed and "transparency" is the key expectation; left the boundary between provided training/validation and provided test imagery for self-labeling not explicitly addressed.

[2] FathomNetCLEF Organizing Committee, FathomNetCLEF 2026: Object detection in deep-sea marine imagery, Kaggle competition, 2026. URL: https://www.kaggle.com/competitions/fathomnet-2026, accessed 2026-05-08.

[3] 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.

[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] 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: European Conference on Computer Vision (ECCV), 2014, pp. 740–755.

[6] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision (ECCV), 2020, pp. 213–229.

[7] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, J. Chen, DETRs beat YOLOs on real-time object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.

[8] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H.-Y. Shum, DINO: DETR with improved denoising anchor boxes for end-to-end object detection, in: International Conference on Learning Representations (ICLR), 2023.

[9] S. Zhang, X. Wang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo, K. Chen, Dense distinct query for end-to-end object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023.

[10] I. Robinson, P. Robicheaux, M. Popov, D. Ramanan, N. Peri, RF-DETR: Neural architecture search for real-time detection transformers, arXiv:2511.09554; software: https://github.com/roboflow/rf-detr, 2025. arXiv:2511.09554.

[11] B. Xiao, H. Wu, W. Xu, X. Dai, H. Hu, Y. Lu, M. Zeng, C. Liu, L. Yuan, Florence-2: Advancing a unified representation for a variety of vision tasks, 2024.

[12] FathomNet, FathomNet huggingface model releases (MBARI vars pretrained detectors), https://huggingface.co/FathomNet, 2025.

[13] N. Bodla, B. Singh, R. Chellappa, L. S. Davis, Soft-NMS – improving object detection with one line of code, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2017.

[14] R. Solovyev, W. Wang, T. Gabruseva, Weighted boxes fusion: Ensembling boxes from different object detection models, Image and Vision Computing 107 (2021).

[15] H. Zhou, Z. Li, C. Ning, J. Tang, CAD: Scale invariant framework for real-time object detection, in: Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), 2017. Introduces Non-Maximum Weighted (NMW) as part of the multi-resolution detection framework.

[16] M. Xu, Y. Bai, B. Ghanem, Positive-unlabeled object detection with confidence-aware self-training, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.
[17] Y. Yang, K. J. Liang, L. Carin, Object detection as a positive-unlabeled problem, in: Proceedings of the British Machine Vision Conference (BMVC), 2020. ArXiv:2002.04672.
[18] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR), 2015. Multi-scale evaluation / test-time augmentation popularized in Section 3.2.
[19] A. Van Etten, You Only Look Twice: Rapid multi-scale object detection in satellite imagery, arXiv preprint arXiv:1805.09512, 2018.
[20] F. C. Akyon, S. O. Altinuc, A. Temizel, Slicing aided hyper inference and fine-tuning for small object detection, in: Proceedings of the IEEE International Conference on Image Processing (ICIP), 2022.
[21] H. Zhang, C. Hao, W. Song, B. Jiang, B. Li, Adaptive slicing-aided hyper inference for small object detection in high-resolution remote sensing images, Remote Sensing 15 (2023) 1249. doi:10.3390/rs15051249.
[22] A. Tarvainen, H. Valpola, Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results, in: Advances in Neural Information Processing Systems (NeurIPS), 2017.
[23] Y. Liu, et al., Source-free domain adaptation for object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.
[24] J. He, et al., VFM-guided source-free object detection with mean-teacher self-training, in: Proceedings of the European Conference on Computer Vision (ECCV), 2024.
[25] W. Huang, et al., Dual-source pseudo-label fusion for cross-domain object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.
[26] M. Wortsman, G. Ilharco, J. W. Kim, M. Li, S. Kornblith, R. Roelofs, R. G. Lopes, H. Hajishirzi, A. Farhadi, H. Namkoong, L. Schmidt, Robust fine-tuning of zero-shot models, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.
[27] H. Wei, et al., LoFT: Long-tail semi-supervised learning via frequency-aware pseudo-labeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024.
[28] B. Trabucco, K. Doherty, M. Gurinas, R. Salakhutdinov, Effective data augmentation with diffusion models, in: International Conference on Learning Representations (ICLR), 2024.
[29] L. Dunlap, A. Umino, H. Zhang, J. Yang, J. E. Gonzalez, T. Darrell, Diversify your vision datasets with automatic diffusion-based augmentation, arXiv preprint arXiv:2305.16289, 2023.
[30] Black Forest Labs, FLUX.2: Rectified flow transformer for production image pipelines, Released 2025-11-25. https://bfl.ai/models/flux-2, 2025.
[31] L. Zhang, A. Rao, M. Agrawala, Adding conditional control to text-to-image diffusion models, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2023.
[32] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, L. Zhang, Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection, in: European Conference on Computer Vision (ECCV), 2024.
[33] G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, B. Zoph, Simple copy-paste is a strong data augmentation method for instance segmentation, 2021.
[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin Transformer: Hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2021.
[35] J. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999.
[36] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, On calibration of modern neural networks, in: International Conference on Machine Learning (ICML), 2017.
[37] A. Gupta, P. Dollár, R. Girshick, LVIS: A dataset for large vocabulary instance segmentation, in:

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. Introduces repeat-factor sampling (RFS) for long-tail instance segmentation.

[38] D. Zimmerman, S. Shukla, G. Sklivanitis, FathomNetCLEF 2026 submission pipeline, Code and supplementary materials: https://github.com/C2A2-at-Florida-Atlantic-University/FathomNet-CLEF-2026, 2026.

[39] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al., DINOv2: Learning robust visual features without supervision, Transactions on Machine Learning Research (TMLR) (2024). ArXiv:2304.07193.

[40] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, LoRA: Low-rank adaptation of large language models, in: International Conference on Learning Representations (ICLR), 2022.

[41] E. Ben Zaken, Y. Goldberg, S. Ravfogel, BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL), 2022.

A. Full MBARI 315k Category Distribution

For full reproducibility, the per-(target-category) distribution of the 315k YOLOv8 model's mapped species classes is provided as `data/concept_to_category.json` in our public code repository. Of the model's 4,218 species classes, 3,212 (76.1%) map to a competition category and the remaining 1,006 (23.9%) are dropped at inference (non-organism concepts, out-of-scope taxa); head categories such as bony fish absorb hundreds of distinct species while long-tail categories such as chiton and pyrosome map from only a few.

Benthic / Midwater dictionaries. The 29-supercategory Benthic mapping and 22-supercategory Midwater mapping are inlined as Python dictionaries in the inference scripts `src/evaluation/generate_submission_mbari_pretrained.py` and `src/evaluation/generate_submission_mbari_old_midwater.py`. These dictionaries fit on a single screen each; we omit them here to save space but they are part of the released code archive.