Appendix - Meta Challenge + Tech Reports

Generalist Meta Challenge

The Generalist Meta Challenge evaluates whether a method remains competitive across multiple maritime tasks rather than excelling on only one benchmark. Teams must participate in at least two active challenge tracks and submit a short report (maximum four double-column pages, excluding references) that justifies why the method should be considered generalist and lists all evaluated tracks.

Evaluation protocol and metrics. Let there be M challenge leaderboards, with leaderboard i containing N_i ranked entries. For each model j, the per-leaderboard rank is defined as

$$r_{ij} = \begin{cases} \text{observed rank of } j, & j \in \textit{leaderboard } i \\ N_i + 1, & j \notin \textit{leaderboard } i \end{cases} \tag{2}$$

Each rank is converted to a normalized score in $[0, 1]$ using

$$s_{ij} = \max\left(0, 1 - \frac{r_{ij} - 1}{N_i - 1}\right), \tag{3}$$

so rank 1 maps to 1, the last listed rank N_i maps to 0, and missing entries ($N_i + 1$) also map to 0. The final metric is the *Consistency Score*, computed as the arithmetic mean across leaderboards:

$$C_j = \frac{1}{M} \sum_{i=1}^{M} s_{ij}. \tag{4}$$

This protocol makes scores comparable across heterogeneous leaderboards and penalizes selective participation, because non-submitted tracks contribute zero to the final average.

Participation. As this is a newly introduced challenge in the current MaCVi edition, no teams submitted eligible entries for the Generalist Meta Challenge. Accordingly, no official rankings, consistency scores, or winning methods are reported for this track in this version of the workshop summary.

Technical Reports

A. Vision-to-Chart Data Association Challenge

A.1. (1) Skyline-Aware ROI-Calibrated Buoy Association

Wonwoo Jo, Hansol Kim, Hyewon Chun, Sangmun Lee, Jeeyeon Jeon
HD Korea Shipbuilding & Offshore Engineering Co., Ltd.

The winning submission uses a staged pipeline that explicitly injects geometric structure before final buoy–query assignment. The method first estimates the skyline to compensate vessel roll, then projects each chart query into a query-conditioned image ROI using learned distance- and bearing-dependent calibration curves. Buoy candidates are generated with RF-DETR 2XL from both the full image and ROI crops, matched to chart queries with Hungarian assignment, and filtered with a gradient-boosted calibrator trained on assignment features. According to the submitted report, this design improves the overall validation score from 0.317 for the organizer baseline to 0.885 on the validation and from X to 0.765 on the test split in the final packaged system, and achieved the top leaderboard performance on both splits.

A.2. (2) Learned World-to-Image Projection for Query-Conditioned DETR

Borja Carrillo-Perez
Arquimea Research Center

The second-place method extends the organizer's DETR-style baseline with a dedicated QueryMLP that predicts the buoy's waterline contact point directly in image coordinates from chart measurements and IMU orientation. The predicted pixel location is appended to the query embedding, reducing the geometric reasoning burden on the transformer decoder and providing a stronger spatial prior for association. The report emphasizes the use of normalized distance, inverse distance, bearing, pitch, roll, and heading as inputs to the projection MLP, followed by a standard DETR training pipeline with score-bias calibration at inference. This lightweight modification achieved an overall private-test score of 0.7386 and ranked second on the official leaderboard.

A.3. (3) Dynamic Chart-Derived Queries with DEIMv2

Yusi Cao, Jiahui Wang, Lingling Li, Xu Liu, Licheng Jiao

The third-place report builds on the DEIMv2 detection framework with a DINOv3 ViT-Tiny backbone and adapts it to the vision-to-chart setting through dynamic chart-derived queries. Each chart marker is encoded by a lightweight MLP

into a hidden embedding and processed jointly with multi-scale image features in a decoder that supports a variable number of valid queries via masking. The network predicts both a visibility confidence and normalized bounding-box coordinates for each query. The submitted model contains approximately 8.37M parameters, is trained for 120 epochs with AdamW and cosine decay, and uses horizontal flips together with query noise augmentation on distance and bearing. The reported validation performance was 0.409 precision, 0.385 recall, 0.397 F1, and 0.363 mIoU, corresponding to third place in the challenge ranking.

A.4. (4th) IMU-Conditioned Query DETR for Maritime Buoy Detection

Vinayak Nageli[1], *Arshad Jamal*[2], *Rama Krishna S. Gorthi*[1]
[1]*Indian Institute of Technology Tirupati,* [2]*Centre for Artificial Intelligence and Robotics (CAIR), DRDO*

The fourth-place submission replaces the standard learned object queries of the baseline with geometry-aware query embeddings derived from chart distance, chart bearing, and an encoded IMU state vector. The model uses a ResNet-50 backbone and a transformer detector whose query representation is produced by a small MLP operating on distance, sine/cosine bearing encoding, and IMU context. To improve localization of small buoys, the authors additionally introduce a weighted horizontal-center loss, motivated by the importance of accurate lateral placement in this task. Training is performed in two stages with AdamW and cosine annealing, first with BCE+L1+GIoU losses and then with an L1-focused continuation stage. The report highlights improved convergence from navigational priors. However, the submission reports a validation overall score of 0.2147, which could not be reproduced by the challenge organizers.

B. Thermal Object Detection Challenge

B.1. ① Multi-Architecture Ensemble with Semi-Supervised Learning

Tze-Hsiang Tang
seantangth@gmail.com
Schneider Electric Taiwan Co., Ltd

This report describes a 6-model ensemble with multi-scale test-time augmentation (TTA), low-threshold pseudo-labeling, semi-supervised learning via MixPL [4], and CLAHE-based TTA diversity, fused via Weighted Boxes Fusion (WBF) [48].

Models. Five distinct architectures are combined for prediction diversity: Co-DINO [56] (Swin-L, O365→COCO pretrained, val AP ≈0.456), DDQDETR [55] (Swin-L, val 0.434), DINO [54] (Swin-L, val 0.428), RTMDet-l [37] (CSPNeXt, val 0.437), and RF-DETR [46] (DINOv2 ViT, val 0.442). A 6th model—Co-DINO trained with MixPL teacher-student EMA (val 0.472)—provides the semi-supervised component. All models except RF-DETR use MMDetection 3.3.0.

Training. Training proceeds in three phases on a single A100 40GB GPU. Phase 1 (∼10 h): five base models are trained on the 704 labeled images with AdamW (lr=10^{-4}, wd=5×10^{-4}), cosine schedule, EMA, and RandomResize (0.5–1.5×) + flip augmentation at resolutions 1333×1000 (DETR variants) and 960×720 (RTMDet). Phase 2 (∼5 h): pseudo labels are generated on the 381 test images from ensemble predictions at per-class thresholds (vessel ≥0.35, nav ≥0.25), yielding ∼794 pseudo annotations; all five models are fine-tuned on 1085 images (704 GT + 381 pseudo). Phase 3 (∼2 h): MixPL teacher-student EMA (momentum=0.0002) is applied to Co-DINO for 4000 iterations (batch size 2, gradient accumulation 4), producing the 6th model with val AP 0.472 (+0.016 over Phase 2).

Inference & Ensemble. Each model runs TTA at 4 scales × 2 orientations = 8 sources. For the two Co-DINO models, CLAHE preprocessing (clipLimit=3.0, tile=8×8) adds 16 extra sources. The resulting 64 sources (48 standard + 16 CLAHE) are fused via WBF (equal weights, iou_thr=0.74, skip=0.15). Full inference takes ∼2–3 min/image on an A100.

Domain observations. 63.4% of objects are $< 32 \times 32$ px (nav median 8.3×18.6 px). Higher inference resolution is the single largest lever. Architecture diversity (DETR + anchor-based + ViT) provides complementary predictions. CLAHE improves strong models but hurts weaker ones (-0.002 to -0.008 AP). Semi-supervised MixPL gave the largest late-stage gain (+0.0033 AP). Only the competition dataset was used; pretraining utilized COCO, Objects365 (Co-DINO), and ImageNet-22K (Swin-L).

B.2. ② Optimization-Diverse Multi-Scale Ensemble

Chun-Ming Tsai[1], *Jun-Wei Hsieh*[2], *Ming-Ching Chang*[3]
cmtsai@go.utaipei.edu.tw,
jwhsieh@nycu.edu.tw, mchang2@albany.edu
[1]*University of Taipei,* [2]*National Yang Ming Chiao Tung University,* [3]*University at Albany, SUNY*

This report presents an optimization-diverse multi-scale ensemble framework for thermal object detection. Multiple DEIMv2 detectors are trained with different random seeds and input resolutions, and their predictions are combined via WBF [48] to improve robustness and localization accuracy.

Architecture. All models are based on the DEIMv2 detection framework with a DINOv3-X transformer backbone, implemented in PyTorch. The final system ensembles 11 detectors operating at four resolutions (1024×1024, 1280×1280, 1440×1440, and 1600×1600). Predic-

tions from all models are merged using WBF with parameters IoU=0.72, skip_box_thr=0.004, and per-class top-k=250. Higher-resolution models are assigned slightly larger fusion weights.

Training. All models are initialized from the publicly available DEIMv2-X COCO pretrained checkpoint, which uses a DINOv3 backbone pretrained in a self-supervised manner on large-scale datasets (e.g., LVD). No additional external datasets beyond these publicly available pretrained weights are used; all fine-tuning is performed solely on the MaCVi 2026 thermal dataset. Training employs multi-scale augmentation, random horizontal flipping, geometric transformations, and photometric distortions. CLAHE-based contrast enhancement is applied only in the 1440-resolution configuration as a probabilistic augmentation ($p = 0.5$) during training; it is not used during inference.

Inference. Experiments are conducted on a workstation with an NVIDIA RTX 3090. Single-model inference takes approximately 0.2–0.3 s per image; the 11-model ensemble adds additional overhead due to multi-model inference and fusion.

B.3. ③ AGAF (Agreement-Gated Auxiliary Fusion)

Wonwoo Jo, Hyewon Chun, Sangmun Lee
{wonwoojo, hyewonchun, sangmunlee}@hd.com
HD Korea Shipbuilding & Offshore Engineering Co., Ltd.

This report presents AGAF, a pipeline combining RF-DETR [46] 2XLarge detectors with annotation refinement, skyline filtering, and agreement-gated auxiliary fusion via class-aware WBF [48].

Architecture. The system is built on RF-DETR 2XLarge with a DINOv2-Base [41] backbone and windowed attention. Multiple checkpoints trained at 960 px and 1280 px resolutions are combined into a 9-source ensemble via class-aware WBF (IoU = 0.76) with per-class weights for vessels and navigational objects, reflecting the different transfer characteristics of each category.

Domain-specific adaptations. Four key adaptations target the thermal maritime domain: (1) *Annotation refinement*: the provided labels contain inconsistent wind-turbine annotations; correcting these and producing *turbine-added* (primary) and *turbine-removed* (auxiliary) annotation variants yields the single largest gain (+0.027 AP). (2) *Skyline filtering*: a YOLO-based [16] horizon detector, trained on manually annotated skyline boxes from the challenge images, suppresses navigational-object false positives above the estimated horizon (+0.012 AP). (3) *Class-aware ensemble*: per-class WBF source weights account for category-dependent detection difficulty. (4) *Agreement-gated fusion*: an auxiliary model trained on the turbine-removed annotations confirms class-1 (vessel) hypotheses asymmetrically—it can only confirm, not create, detections, avoiding label-policy mismatch between annotation variants.

Training. All models use AdamW with base LR 10^{-4} and encoder LR 1.5×10^{-4}, step LR decay at epoch 40, ViT layer decay 0.8, EMA (decay 0.993), and multi-scale training with expanded scales. Training runs range from 10 to 50 epochs depending on the variant. Only the MaCVi 2026 dataset is used (667 train / 162 val images); no external maritime data or annotations are employed. The DINOv2 backbone uses standard ImageNet-pretrained initialization.

Inference. Experiments are conducted on a single NVIDIA RTX 5070 (12 GB). Single-model inference runs at ∼15 FPS at 960 px; the full 9-source ensemble pipeline operates at ∼1.5 FPS per image.

C. LaRS Panoptic Segmentation Challenge

C.1. ① M2F-DINOv3

Ivan Martinović
ivan.martinovic@fer.hr
Faculty of Electrical Engineering and Computing, University of Zagreb, Croatia

Our method is based on Mask2Former [5] with a pretrained DINOv3 [47] vision transformer backbone. The Mask2Former segmentation head expects a multi-scale feature pyramid, whereas the DINOv3 backbone produces features at a native stride of 16, denoted by $\mathbf{F}_{16}$. To bridge this mismatch, we construct a feature pyramid following the design used in ViTDet [30] and EoMT [18]. More specifically, we build feature maps at strides 4, 8, 16, and 32, corresponding to $\mathbf{F}_4$, $\mathbf{F}_8$, $\mathbf{F}_{16}$, and $\mathbf{F}_{32}$. Starting from $\mathbf{F}_{16}$, we obtain higher-resolution features using repeated upsampling blocks, and lower-resolution features using downsampling blocks. Each block consists of a 2×2 convolution (transposed for upsampling, standard for downsampling) with stride 2, followed by a GELU activation, a depthwise 3×3 convolution, and a normalization layer. In our implementation, the same stride-16 DINOv3 feature tensor is reused to generate all pyramid levels. The pixel decoder and transformer decoder follow the original Mask2Former design. We evaluate two backbone variants, DINOv3-L and DINOv3-H+. The models are trained on four NVIDIA RTX A6000 Ada GPUs (48 GB VRAM per GPU) with a total batch size of 16. Training is performed for 90,000 iterations using a crop size of 512×1024. Unless otherwise stated, the remaining optimization and decoder hyperparameters follow the standard Mask2Former configuration for Cityscapes. To improve performance on underrepresented categories, we apply rare-class oversampling. Following the repeat-factor sampling strategy introduced for LVIS [13], we first compute the image-level frequency $f(c)$ of each class c, i.e., the fraction of training images in which class c appears. We then define

Table 6. Panoptic segmentation results on the LARS validation set.

Method	PQ	SQ	RQ
Mask2Former + DINOv3-L	54.8	75.8	63.5
Mask2Former + DINOv3-H+	55.8	75.5	65.4

the class repeat factor as

$$r(c) = \max\left(1, \sqrt{\frac{t}{f(c)}}\right),$$

where t is the repeat threshold. The repeat factor assigned to an image is the maximum repeat factor among all classes present in that image. In our experiments, we use $t = 0.25$. In addition, we use a rare-class-aware cropping strategy. For images containing rare classes, we first sample a target segment with probability proportional to its class repeat factor. We then draw up to 10 random crops and accept the first one that contains at least 10% of the target segment area. If none of the sampled crops satisfies this condition, we keep the last sampled crop. This procedure makes rare classes more likely to appear in the final training crop. Table 6 reports validation-set results for the two backbone variants. For the final submission, we retrained the Mask2Former + DINOv3-H+ model on the union of the training and validation splits. The resulting model achieved 53.5 PQ on the LARS test set.

C.2. ② MaskDINOv3

Jannik Sheikh[1], Andreas Michel[1], Wolfgang Gross[1], Martin Weinmann[2]
firstname.lastname@iosb.fraunhofer.de
martin.weinmann@kit.edu
[1]*Fraunhofer IOSB, Germany*
[2]*Karlsruhe Institute of Technology, Germany*

This technical report describes our solution to the 4th Workshop on Maritime Computer Vision (MaCVi) USV-based Panoptic Segmentation challenge, building upon our previous top-3 submission (PQ 53.9) using MaskDINO [29].

This work evaluates DINOv3 [47], a vision foundation model for dense features, as a replacement for the ImageNet-22K [45] pre-trained Swin-L [32] backbone. Since MaskDINO requires multi-scale features but DINOv3 produces single-scale outputs, we employ DEIMv2 [15], which uses a Spatial Tuning Adapter (STA) to convert DINOv3 outputs into multi-scale features.

Experimental Settings. Experiments were conducted on four NVIDIA H100 GPUs using a distilled DINOv3 ViT-H+/16 model (840M parameters). Features were extracted from ViT blocks 14, 22, and 31 and passed to the STA module, which used a base channel dimension of 32 for its convolutions. Training with AdamW [35] followed two stages: (1) frozen backbone, training STA and MaskDINO head for 25k iterations with base LR 2.2×10^{-5}; (2) full fine-tuning in two runs with progressively reduced LR (2.2×10^{-5} then 5×10^{-6}) and backbone multipliers (0.1, 0.05) for 15k and 10k iterations respectively, along with per-class weighting to improve PQ^{Th}. Both stages used augmentations including 1024×1024 crops, horizontal flips, and multi-scale resizing.

Observations and Remarks. On LaRS [60] test, our model achieves PQ 48.3 (F1 69.5). Compared to MaskDINO with Swin-L, PQ^{St} improved notably (92.3 → 94.9), indicating DINOv3's suitability for stuff segmentation. However, PQ^{Th} declined (39.4 → 30.9) despite per-class weighting. We hypothesize this stems from pre-training differences: Swin-L benefits from COCO panoptic pre-training providing instance-level priors, while DINOv3's self-supervised training may lack instance-discriminative representations. The PQ^{St} gain highlights DINOv3's potential, warranting further research on thing-class adaptations.

C.3. ③ ThingSeg-DGCR

Hyewon Chun, Wonwoo Jo, Sangmun Lee
{hyewonchun, wonwoojo, sangmunlee}@hd.com
HD Korea Shipbuilding & Offshore Engineering Co., Ltd.

Algorithm outline. We use two MaskDINO-R50 checkpoints as the panoptic backbone and an RF-DETR-Seg Medium branch for thing recovery, building on the official MaskDINO [29] and RF-DETR [46] codebases. One MaskDINO checkpoint is trained with rare-class rebalancing and the other with thing-aware crop fine-tuning; both are evaluated at scale 896 with horizontal flips. The RF-DETR branch runs only on the eight thing classes at resolution 720. We merge detector masks into the panoptic output only when conservative overlap rules are satisfied, then apply class-specific GrabCut refinement.

Training. The MaskDINO models start from Detectron2 ImageNet-pretrained ResNet-50 weights and use batch size 1 with large-scale jitter at 896. We then run 16k iterations with rare-class repeat sampling and 8k more with thingaware crops. RF-DETR-Seg Medium uses EMA, gradient checkpointing, batch size 2 with gradient accumulation 8, and offline copy-paste of rare thing instances onto water regions, initialized from an earlier LaRS RF-DETR model.

Datasets. We used only official LaRS data and derivatives: the panoptic train/validation/test splits, a thing-only COCO export derived from LaRS, and synthetic copy-paste images built from LaRS training images. We did not use external maritime datasets, private annotations, or external imagery. The ResNet-50 backbone uses standard ImageNet pretraining.

Hardware and inference speed. All training and inference were run on a single NVIDIA RTX A4000 GPU. A single MaskDINO checkpoint runs at about 3.5–4.5 FPS; the full pipeline runs at roughly 0.5–0.7 FPS.

Domain-specific adaptations. In LaRS, stuff is relatively easy; the main challenge is recovering small maritime thing instances. Buoys, swimmers, paddle boards, row boats, and rare obstacles are sparse, small, and visually ambiguous, so we treated the task primarily as a thing-recovery problem. That led us to rare-class rebalancing, thing-aware crops, a conservative detector branch, and stronger GrabCut settings for buoys and swimmers.

D. LaRS Embedded Segmentation Challenge

D.1. ① DSOS-Net: DINOv3-based Water Surface Obstacle Segmentation Network

Yuan Feng
fengyuan9822@outlook.com,
931772830@qq.com
Independent Researcher

Method. Inspired by the encoder-decoder architecture, we propose DSOS-Net for real-time surface obstacle detection. The network architecture primarily consists of an encoder section and a decoder section. In the encoder, we employ ConvNeXt [33] as the backbone network, which has been self-supervised pre-trained on the LVD-1689M dataset using the DINOv3 method [47], enabling effective extraction of robust multi-scale features. The decoder adopts the RSOS-Net [50] architecture, which comprises a lightweight feature pyramid network, a fast pyramid pooling module, and an attention-based feature fusion module. Specifically, the fast pyramid pooling module effectively expands the receptive field by combining global average pooling and cascaded average pooling operations, enabling the capture of both global and local contextual information, which is crucial for distinguishing water surface disturbances from real obstacles in maritime environments. The attention-based feature fusion module, employing a channel-spatial attention mechanism, enables DSOS-Net to focus more on real obstacles, reducing false positives and missed detections. The combination of the robust feature representation from ConvNeXt and the efficient segmentation head from RSOS-Net achieves a balance between accuracy and inference speed.

Training. We implemented DSOS-Net using PyTorch with an image input size of 768×384 and a batch size of 6. The training was conducted on an NVIDIA RTX 2080 Ti GPU with 12 GB memory. We adopted a two-stage training strategy. In the first stage, the model was trained for 100 epochs using the AdamW optimizer with an initial learning rate of 1×10^{-4} and a cosine learning rate scheduler. The loss function combines cross-entropy loss and Dice loss with equal weights. To address the severe class imbalance problem, we set the class weights for obstacle, water, and sky classes as 3.0, 1.0, and 1.0, respectively. In the second stage, we continued training for an additional 20 epochs, resulting in a total of 120 epochs, with equal class weights of 1.0 for all classes to achieve better precision-recall balance. For data augmentation, techniques including random horizontal flip with probability 0.5, random vertical flip with probability 0.3, random rotation within $\pm 15°$, color jitter with brightness, contrast, and saturation adjustments within ± 0.3 and hue within ± 0.15, and Gaussian blur were applied. Regarding datasets, only the LaRS dataset [60] was utilized for training and validation. The backbone network was initialized with ConvNeXt weights pretrained by the DINOv3 method.

Observations

- **Two-stage Training Effect:** In our submissions, the version with ID 16965 from the first stage achieved a Q-score of 61.5, F1-score of 65.7, and mIoU of 93.5, with precision of 56.7% and recall of 78.2% at 55.9 FPS. The high class weight for the obstacle class resulted in high recall but relatively low precision. The version with ID 17010 from the second stage achieved the best Q-score of 61.9, F1-score of 66.0, and mIoU of 93.8, with precision of 62.6% and recall of 69.7% at 66.5 FPS, ranking 1st on the leaderboard. The inference speed was evaluated by submitting ONNX models to the official evaluation server.
- **Backbone Efficiency:** With a smaller ConvNeXt variant, DSOS-Net can achieve up to 98 FPS while maintaining competitive segmentation accuracy.
- **Feature Representation:** The self-supervised pre-training of DINOv3 on large-scale datasets provides robust feature representations that transfer well to maritime obstacle segmentation, effectively reducing the domain gap between pre-training and downstream tasks.
- **Evaluation Strictness:** The evaluation of the USV-based Embedded Obstacle Segmentation challenge appears to be stricter compared to last year.
- **Code Availability:** The code for DSOS-Net will be publicly accessible soon at https://github.com/Yuan-Feng1998.

D.2. ② PIDNet-S with Copy-Paste Obstacle Augmentation

Jose Mateus Raitz Persch[1], *Rahul Harsha Cheppally*[2]
jmraitzp@protonmail.com, r4hul@ksu.edu
[1]*Independent Researcher,* [2]*Kansas State University*

Architecture. We use PIDNet-S [51] (github.com/XuJiacong/PIDNet), a lightweight three-branch (Proportional–Integral–Derivative) network with 32 base channels, PPM pooling (96 channels), and a PIDHead decoder (128 channels, 3 classes). The Derivative branch explicitly models boundary detail, which is particularly relevant for maritime segmentation where water–obstacle boundaries are often ambiguous. The backbone is initialized from ImageNet-1K pretrained weights. We made no architectural modifications—PIDNet-S was selected specifically

because its BatchNorm + Conv + ReLU composition quantizes cleanly under the server's INT8 quantization pipeline on the Luxonis RVC4. During our evaluation phase, we found that architectures relying on attention mechanisms (SegFormer) or large-kernel depthwise convolutions (SegNeXt) suffered catastrophic INT8 degradation (-67% to -82% mIoU), making simple convolution-based designs the only viable choice.

Training. We train at the target resolution of 768×384 using SGD (lr = 0.133, momentum = 0.9, weight decay = 5×10^{-4}) for 15,000 iterations with polynomial LR decay (power = 0.9) and a 2,500-iteration linear warmup. Batch size is 320 (160 per GPU). Losses: CrossEntropy ($w = 0.4$), two OHEM losses ($w = 1.0$, threshold = 0.9), and BoundaryLoss ($w = 20$). Augmentations include Large Scale Jittering (ratio $\in [0.5, 2.0]$), random crop, horizontal flip, enhanced photometric distortion (brightness, contrast, saturation, hue, gamma jitter), and reflect padding matching the server preprocessing. Framework: mmsegmentation [6], PyTorch 2.1.2, CUDA 12.1.

Copy-Paste augmentation. The key component of our method is a Copy-Paste obstacle augmentation [11] using crops extracted from three external maritime datasets: **WaterScenes** [53], **ROSEBUD** [28], and the night subset of **MULTIAQUA** [40]. All three were remapped to the LaRS 3-class schema and deduplicated with perceptual hashing. We mined ∼30,000 obstacle crops (water-adjacent, area 50–100k px at target resolution) and stored them in a shared database. During training, with probability 0.7, 1–3 crops are sampled with size-based tier weights (oversampling small obstacles), brightness-matched to the local water region, and pasted before photometric distortion. This approach was motivated by a failure analysis of our Phase 2 baseline server submission and a spatial out-of-distribution study on the LaRS test set, both of which showed that the majority of false negatives involved small or rare obstacle types under unusual lighting—gaps that external maritime datasets could address. An initial run with a shorter schedule (8k iterations) failed due to "augmentation shock," which we resolved by extending to 15k iterations with longer warmup.

Datasets. LaRS [60] (train: 2,605 images) for training; ImageNet-1K for backbone pretraining; WaterScenes, ROSEBUD, and MULTIAQUA_night as external augmentation sources (crops only, no joint training). No custom-annotated data was used.

Hardware & inference speed. Training was performed on 2× NVIDIA RTX A6000 (48 GB) with DDP. On the challenge server (Luxonis RVC4, INT8 quantization), our model achieves 67.7 FPS.

D.3. ③ RSOS-Net R50 PyTorch + YOLOX-HR

Justin Davis, Mehmet E. Belviranli
{jcdavis, belviranli}@mines.edu

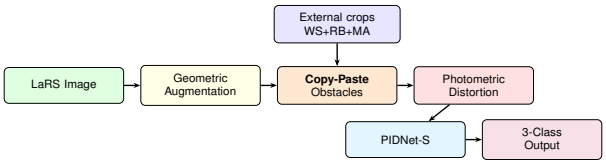

Figure 12. Training pipeline. WS = WaterScenes, RB = ROSEBUD, MA = MULTIAQUA_night.

Colorado School of Mines, Colorado, United States

Method. Our submission is a PyTorch reimplementation of RSOS-Net [50], the 1st-place solution from the MaCVi 2025 Embedded Obstacle Segmentation Challenge [24]. RRSOS-Net is a lightweight encoder-decoder segmentation network designed for water-surface obstacle detection, using multi-scale context pooling and channel-spatial attention to suppress false positives from reflections, sun glitter, and wakes. In the MaCVi 2025 challenge proceedings, the authors submitted both a ResNet-50 variant (Q = 63.6, 102.8 FPS) and a ResNet-101 variant (Q = 64.2, 85.1 FPS), with the R101 achieving the top accuracy score. The subsequent journal paper [50] provides a comprehensive description of the architecture using a ResNet-18 backbone. Our reimplementation is based on this full architectural specification, adapting the design to a ResNet-50 backbone with a 4-scale feature pyramid (layers 1–4) and applying a revised training pipeline. The ResNet-50 backbone is initialized with ImageNet-1K pretrained weights and uses output stride 16 via dilated convolutions in stage 4.

Training. All training used only the LaRS [60] dataset (SGD, 16-bit mixed precision, batch size 8, 200 epochs). The loss combines cross-entropy, dice loss (weight 2.0), and focal loss (weight 1.0) on the main head, with standard cross-entropy on two auxiliary FCN heads (weight 0.4). Augmentations include random scale/crop/flip, color jitter, brightness–contrast and HSV shifts, Gaussian and motion blur, noise, random shadows, and coarse dropout. Training was performed on a single NVIDIA RTX 5080 GPU. The original authors trained for 160,000 steps at batch size 16 (∼1,000 epochs), roughly 5× our training volume, which likely accounts for some of the accuracy gap between our R50 (Q = 46.4) and theirs (Q = 63.6).

Additional Submission—YOLOX-HR. We also submitted YOLOX-HR [10], a custom high-resolution variant of YOLOX using the medium CSPDarknet53 backbone (YXHR_medium), motivated by the observation that small dynamic obstacle detection was critical for F1/Q metrics. YOLOX-HR employs a stacked dual Path Aggregation FPN (PAFPN) that fuses features from stride 2 through 32, yielding predictions at stride 2 to preserve spatial detail for small objects. Its purely convolutional architecture (Conv-BN-SiLU) typically has highly optimized kernels on embedded

platforms such as the RVC4.

Observations.

- RSOS-R50 achieves Q = 46.4 at 96.3 FPS, while YXHR_medium achieves Q = 45.3 at 64.9 FPS. Despite YOLOX-HR having more parameters and higher mIoU, the overall Q is lower at a significant FPS cost, showing that RSOS-Net's lightweight decoder modules scale more efficiently at nearly $1.5\times$ the inference speed.
- YOLOX-HR showed better Q scores than our RSOS-Net variant during training but overfit to the train/val split, most likely due to the low amount of training data relative to the parameter count.
- The accuracy gap between our R50 and the original authors' R50 is likely attributable to: training duration ($5\times$ fewer samples), adaptation of the published R18 architecture to an R50 backbone, and training pipeline differences.

E. Multimodal Semantic Segmentation Challenge

E.1. ① GatedMemorySAM

Jemo Maeng, Sangmin Park, Seongju Lee, Kyoobin Lee
{maengjemo, leowiu24, lsj2121}@gm.gist.ac.kr, kyoobinlee@gist.ac.kr
GIST AI LAB, Gwangju Institute of Science and Technology, South Korea

Method. We introduce **GatedMemorySAM**, which builds upon MemorySAM [31] that repurposes SAM2's [44] temporal memory attention for cross-modal fusion by treating each input modality (RGB, LiDAR, Thermal) as a separate "frame" in SAM2's video pipeline. We extend this framework with two key modifications: (1) Soft MoE LoRA adaptation, and (2) quality-aware modality scoring.

For backbone adaptation, we replace standard LoRA [14] with Soft MoE routing [43] over multiple LoRA experts, inserted into every attention block (Q and V projections, 48 layers total) with rank $= 4$ and 3 experts. Each spatial token is softly routed to all experts via a learned gating network, enabling per-token specialization across modalities.

For modality weighting, we introduce a `CrossModalFusionHead` that compares all modality features via global average pooling and a shared comparison MLP to produce per-modality softmax weights. These weights drive two score-based fusion mechanisms: (i) *memory modulation*, which max-normalizes the scores and scales each modality's backbone features before they enter SAM2's memory bank, so that the best modality retains full strength while weaker ones are suppressed; and (ii) *weighted mask fusion*, which averages the per-modality decoder outputs using the same softmax weights. The overall pipeline is illustrated in Figure 13.

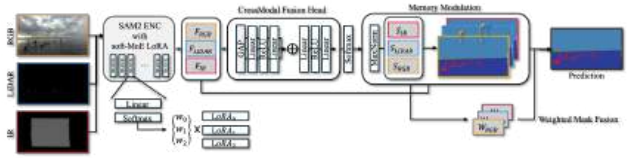

Figure 13. Overview of GatedMemorySAM. Each modality is encoded by SAM2 with Soft-MoE LoRA, scored by the CrossModalFusionHead, and fused via memory modulation and weighted mask fusion.

Training Details. We train with the AdamW optimizer (lr $= 6 \times 10^{-4}$, weight decay $= 0.01$) using a warmup polynomial LR schedule (10-epoch warmup, power $= 0.9$) for 200 epochs. The loss function is Online Hard Example Mining Cross-Entropy (OHEM-CE). Input images are resized to 1024×1024. We use DDP training with an effective batch size of 16 across 8 GPUs with gradient accumulation.

Night Augmentation. Since the test set consists entirely of nighttime images while training data is daytime, we apply an extensive nighttime simulation pipeline: brightness darkening (range $[0.03, 0.45]$ with 60% dark-biased sampling), contrast reduction, gamma correction ($[0.4, 0.8]$), Gaussian and Poisson shot noise, cross-modal replacement (CRM, $p = 0.20$), and PhysAug [52]-inspired spatial perturbations (random convolution filters and planar Fourier wave patterns, $p = 0.40$).

Datasets and Pretrained Weights. We train exclusively on the MULTIAQUA dataset provided by the challenge organizers. The SAM2 Hiera-B+ backbone is initialized from publicly available pretrained weights. No additional external datasets or annotations are used.

Hardware and FPS. Training was conducted on $8\times$ NVIDIA RTX 3090 GPUs (24GB each). Inference runs on a single NVIDIA Titan RTX GPU at approximately 2.1 FPS (1024×1024 input, 3 modalities processed sequentially).

Maritime Domain Adaptations. The primary challenge is the extreme day-night domain gap (daytime training vs. nighttime-only test). Our nighttime simulation augmentation was the single most impactful component, nearly doubling the test mIoU.

E.2. ② Adapted MFNet-SAM-LoRA

Yusi Cao, Jiahui Wang, Lingling Li, Xu Liu, LiCheng Jiao
cys734511@163.com, 18257864149@163.com, llli@xidian.edu.cn, xuliu361@163.com, lchjiao@mail.xidian.edu.cn
School of Artificial Intelligence, Xidian University, China

Method. We adapted the existing Multimodal Finetuning Network (MFNet) by leveraging the Segment Anything Model (SAM, ViT-Large) as our foundational backbone. To process three distinct modalities within MFNet's

dual-branch architecture, we feed the RGB image into the primary branch, while concatenating the Thermal image and projected LiDAR data (distance d and reflectivity r) into a 3-channel auxiliary tensor for the secondary branch. Both inputs are processed concurrently by the frozen SAM encoder, which is equipped with trainable Low-Rank Adaptation (LoRA) layers for parameter-efficient tuning. The extracted features are then fused across scales via Squeeze-and-Excitation (SE) blocks and decoded using a UNet-style head. Observing the drastic day-to-night domain shift in the maritime environment, this Thermal-LiDAR fusion ensures robustness against nighttime RGB degradation. Furthermore, to handle recording boat artifacts, we explicitly set `ignore_index=0` in our loss computation to prevent erroneous structure learning.

Training. Our training strictly utilized the daytime splits of the provided MULTIAQUA dataset, relying solely on SAM's native SA-1B pre-trained weights without any additional maritime datasets or custom pseudo-labels. We optimized the network using AdamW (initial LR=0.001, weight decay=1e-4) with a CosineAnnealingWarmRestarts scheduler over 80 epochs. Data augmentation was minimal, utilizing only 512×512 random cropping and horizontal flipping. To manage hardware constraints, Automatic Mixed Precision (AMP) and gradient accumulation were employed to achieve an effective batch size of 8. All procedures were executed on a single NVIDIA GeForce RTX 4090 GPU, achieving an efficient inference speed of approximately 12 FPS.

E.3. ③ Modified DustNet

Andreas Michel[1], Jannik Sheikh[1], Jannick Kuester[1], Bettina Felten[1], Wolfgang Gross[1], Martin Weinmann[2]
`firstname.lastname@iosb.fraunhofer.de`
`martin.weinmann@kit.edu`
[1]*Fraunhofer IOSB, Germany*
[2]*Karlsruhe Institute of Technology, Germany*

Methodology. Our methodology builds upon the DustNet architecture [38, 39], a deep neural network specifically designed for visual density estimation tasks. The DustNet-C variant leverages a dual-branch encoder architecture to extract temporal, global, and local feature representations.

We adapt this architecture for multi-modal sensor fusion by repurposing the dual input streams. Hereby, the primary branch processes RGB imagery, while the secondary branch receives a fused representation of LiDAR and thermal infrared data. For the latter, 2D projected LiDAR data is concatenated channel-wise with the thermal infrared image, preprocessed using Contrast Limited Adaptive Histogram Equalization (CLAHE) [57]. Each modality stream is processed through dedicated backbone networks for domain-specific feature extraction. The resulting feature maps are subsequently integrated via a cross-attention mechanism.

Experimental Settings. All experiments were conducted using two Nvidia A100 GPUs, each equipped with 80 GB of VRAM. The modified DustNet architecture was implemented with a Swin-L [32] backbone, initialized with weights pretrained on the ImageNet-1K dataset [7] to leverage learned visual representations. The model was trained on the maritime subset of the MULTIAQUA dataset for 36 epochs, employing cross-entropy as the loss function and AdamW [34] as the optimizer. Data augmentation during this phase was limited to large-scale jitter and stochastic horizontal flipping. Subsequently, fine-tuning was performed for 2 additional epochs, incorporating stochastic brightness attenuation on the RGB input channels to enhance robustness under variable illumination conditions. The trained model achieves an inference throughput of 7 fps on a single A100 GPU.

Observations And Remarks. Employing separate backbones for each input branch enables modality-specific feature learning. However, this approach does not inherently outperform a single-backbone architecture with concatenated inputs. Moreover, achieving optimal performance likely requires extensive hyperparameter tuning and a refined training strategy. The exclusive use of the MultiAqua dataset may further limit the model's ability to fully exploit the potential of multi-modal fusion, indicating that a larger and more diverse dataset may be required. Finally, the inherently low contrast of thermal imagery presents an additional challenge, suggesting that more sophisticated preprocessing techniques warrant further investigation.