# Improving Thing Segmentation for USV Panoptic Scene Understanding with Detector-Guided Class-Specific Refinement

Anonymous MaCVi submission

Paper ID 18

## Abstract

*Panoptic segmentation is well suited to unmanned surface vehicle (USV) scene understanding because safe navigation requires both dense parsing of background regions and explicit reasoning about obstacle instances. Maritime panoptic benchmarks remain difficult for a familiar reason: thing classes are often small, sparse, and long-tailed, while stuff classes dominate image area. We therefore build a hybrid system for the LaRS maritime panoptic benchmark that combines a refined MaskDINO panoptic ensemble with an RF-DETR-Seg Medium thing booster and class-specific GrabCut refinement during fusion, yet can still be developed on a single workstation GPU. Our development experiments point to thing segmentation, rather than stuff parsing, as the main source of remaining error. This leads us to focus on panoptic-branch refinement, detector-guided fusion, and targeted boundary refinement for row boats, paddle boards, buoys, swimmers, and selected rare classes. On the local development split used for method design, the system improves panoptic quality from 37.50 to 45.79 and raises thing panoptic quality from 17.47 to 28.49 while keeping stuff quality at 91.92. On the official hidden-test benchmark, the final anonymous submission achieves 42.6 PQ, 24.2 $PQ_{th}$, 91.8 $PQ_{st}$, 51.1 RQ, 71.0 SQ, and 54.4 F1. Overall, these results indicate that detector-guided, class-specific refinement can improve maritime panoptic segmentation without relying solely on larger end-to-end panoptic models.*

## 1. Introduction

Panoptic segmentation unifies semantic and instance segmentation by assigning every pixel a semantic label and, for thing regions, an instance identity [4]. For USV perception, this representation is especially attractive because navigation depends on both large background regions such as water and sky and small dynamic obstacles such as buoys, swimmers, row boats, and floating debris. The LaRS benchmark explicitly targets this regime and is, to the best of our knowledge, among the first large-scale benchmarks for maritime panoptic obstacle understanding [8].

Despite recent progress in transformer-based segmentation [3, 5], maritime panoptic segmentation remains challenging. In LaRS, the three stuff classes occupy most of the scene area and are comparatively easy to predict, whereas the eight thing classes are small, rare, and visually diverse. This asymmetry suggests that, under limited compute, simply scaling a unified panoptic model may not be the most effective strategy.

Our aim in this paper is narrower. Rather than proposing a new backbone or decoder, we focus on the hardest part of the task: improving thing segmentation under single-workstation-GPU constraints. The framework starts from a MaskDINO panoptic model, strengthens it through rare-thing rebalancing and thing-aware fine-tuning, and then ensembles two complementary checkpoints at inference time. In parallel, we train an RF-DETR-Seg Medium model on thing annotations only. Its masks are refined and inserted into the panoptic prediction only when confidence, area, and overlap constraints suggest that they help without disturbing already-correct stuff regions.

Our contributions are threefold. First, on our development split, we show that the main gap is between thing and stuff quality rather than in overall scene parsing. Second, we present a hybrid panoptic-plus-thing pipeline that pairs a stable panoptic base with a detector-style auxiliary branch and remains workable on a single workstation GPU. Third, we provide a component-wise analysis showing that global fusion alone does not deliver the full gain; the last improvements in thing quality come from class-specific refinement, while stuff performance stays nearly unchanged. Taken together, these results outline a simple accuracy-oriented design path for this compute regime.

## 2. Related Work

**Panoptic segmentation.** Panoptic segmentation was introduced by Kirillov et al. [4] and has since been addressed by both multi-branch and transformer-based approaches. Early

MaCVi #18

systems often relied on separate semantic and instance components plus heuristic merging. More recent unified models such as Mask2Former [3] and MaskDINO [5] substantially improve performance by directly predicting masks with transformer decoders.

**Transformer detection and segmentation.** DETR reframed detection as direct set prediction [2], enabling architectures that avoid many hand-crafted post-processing steps. MaskDINO extends this line to universal segmentation [5]. RF-DETR further targets real-time transformer detection [6]. In our work, we use its released segmentation variant as a practical detector-style model for instance mask prediction.

**Maritime visual perception.** Maritime perception differs from common road-scene benchmarks because targets are often tiny, sparsely distributed, and heavily affected by reflections, clutter, and large uniform water regions. SeaDronesSee [7] and MODS [1] highlight the difficulty of small obstacle perception in marine environments. LaRS extends this line by providing a diverse maritime panoptic benchmark with both thing and stuff labels [8].

**Hybrid systems.** Our work is closest in spirit to practical hybrid pipelines that combine a strong dense predictor with a specialized auxiliary module. We do not claim architectural novelty. Instead, we study whether, in the maritime panoptic regime, a carefully constrained fusion of a panoptic model with a thing-specialized model can provide a practical operating point under single-workstation-GPU constraints.

# 3. Problem Setting

We study the LaRS maritime panoptic benchmark [8]. Each pixel must be assigned one of 11 classes: three stuff classes (static obstacles, water, sky) and eight thing classes (boat/ship, row boat, paddle board, buoy, swimmer, animal, float, other). The output is a panoptic map in which every pixel has a semantic class, and every thing region also has an instance identifier.

We report the standard panoptic quality metric PQ together with its decomposition into thing PQ ($PQ_{th}$) and stuff PQ ($PQ_{st}$) [4]. This decomposition is central to our analysis. High $PQ_{st}$ indicates reliable scene parsing of large background regions, whereas low $PQ_{th}$ reveals missed or poorly delineated obstacle instances. Our design objective is to raise $PQ_{th}$ while preserving the already-strong $PQ_{st}$ of the panoptic base model.

# 4. Method

## 4.1. Panoptic Base Model

Our base model is MaskDINO with a ResNet-50 backbone [5]. We choose it for its strong panoptic segmentation performance and its practicality on a single workstation GPU. We train the baseline on LaRS panoptic annotations using $896 \times 896$ large-scale-jitter crops. The baseline already provides reliable stuff parsing, but small obstacle instances are often missed or under-segmented.

## 4.2. Rare-Thing Rebalancing

The LaRS thing distribution is highly imbalanced. In the training split, boat/ship appears in 1491 images, whereas float appears in only 21. To reduce this skew, we create a rebalanced panoptic training stream by repeating images that contain rare thing categories. The repeat factors are category-dependent and largest for the rarest classes. This rebalancing increases how often the panoptic model sees small obstacle categories during fine-tuning.

## 4.3. Thing-Aware Crop Fine-Tuning

After rebalanced fine-tuning, we further refine the panoptic model with a thing-aware crop mapper. Instead of always sampling a random crop after resizing, the mapper selects a crop centered near a randomly chosen non-crowd thing instance with a fixed probability. The selected instance must satisfy minimum-size and maximum-area-ratio constraints so that crops preferentially expose small to medium obstacle regions rather than large easy objects. This change keeps the original training recipe largely intact while increasing the density of informative thing examples.

## 4.4. Inference-Time Panoptic Ensemble

Single-checkpoint inference still leaves many borderline thing errors. We therefore ensemble two complementary MaskDINO checkpoints. The primary checkpoint comes from the thing-aware fine-tuning stage, while the secondary checkpoint comes from the earlier rare-thing fine-tuning stage. For each image, we run both models at the selected test scale with horizontal-flip augmentation.

The ensemble merge is panoptic-aware. Stuff predictions are combined through pixelwise voting over stuff classes. Thing predictions are grouped by semantic class and mask IoU. Within each group, we accumulate support across model and augmentation variants and keep candidates that either appear in the base prediction or satisfy a minimum support threshold. The resulting merged panoptic map improves robustness without requiring retraining.

## 4.5. Thing-Only RF-DETR Booster

To directly target the obstacle bottleneck, we train an RF-DETR-Seg Medium model only on the eight thing categories. This detector-style segmenter is optimized to predict localized object masks rather than to explain the entire scene. In practice, it is particularly useful for proposing small obstacle instances that the panoptic model misses, especially for row boats, paddle boards, buoys, swimmers, and partially occluded boats. We use the best EMA checkpoint from epoch 5 and run inference at resolution 720, which provided

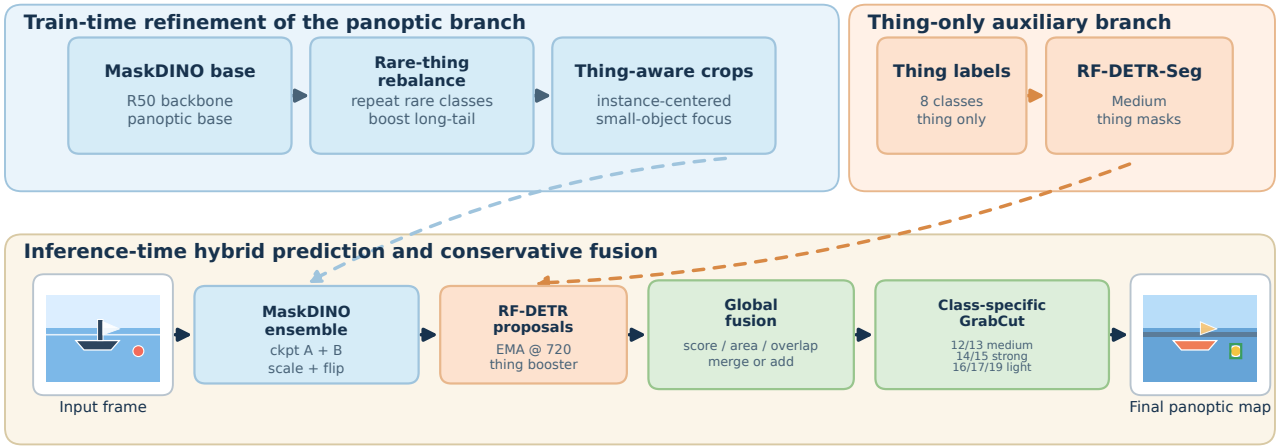


Figure 1. Overview of the proposed hybrid framework. Train-time refinement improves the MaskDINO panoptic branch through rare-thing rebalancing and thing-aware crop tuning, while a separate RF-DETR-Seg Medium model is trained only on thing classes. At inference, the refined panoptic checkpoints are ensembled, augmented with RF-DETR thing proposals, and then passed through conservative global fusion plus class-specific GrabCut refinement for the classes that benefit consistently on the development split.

a good trade-off between thing recall and memory usage in our development setting.

### 4.6. Constrained Fusion

The first inference-stage fusion step inserts RF-DETR masks into the panoptic ensemble output using conservative global rules. Let $m$ be an RF-DETR mask with confidence $s(m)$. We first discard low-confidence or tiny masks. If $m$ overlaps an existing same-class panoptic instance with sufficient IoU, we merge the two. Otherwise, we insert $m$ as a new instance only when its overlap with existing thing masks stays below a threshold. This rule is designed to add missed obstacles without degrading already-correct stuff and thing regions.

In the final configuration, global fusion uses a score threshold of 0.35, a minimum area of 40 pixels, a merge IoU threshold of 0.25, and a maximum overlap of 0.30 with existing panoptic instances. The key design principle is not aggressive replacement, but selective insertion of missing thing instances.

### 4.7. Class-Specific GrabCut Refinement

Global fusion already improves thing recall, but we found that the remaining errors are often boundary-quality failures concentrated in a few thin or low-area classes. We therefore refine detector masks with a lightweight GrabCut step before making the final merge decision. A global setting with margin 12, two iterations, and a minimum refined area of 48 pixels provides a reasonable default configuration.

We then add class-specific overrides only where the development split shows consistent gains. Buoy and swimmer masks (classes 14 and 15) benefit from the strongest refinement, with a wider context margin and more GrabCut iterations. Row boat and paddle board masks (classes 12 and 13) benefit from a more moderate setting. Animal, float, and other (classes 16, 17, and 19) require a more conservative policy with slightly higher score thresholds and smaller valid areas. This targeted policy yields the final improvement from 45.58 PQ to 45.79 PQ on the development split while keeping $PQ_{st}$ effectively unchanged.

## 5. Experimental Setup

### 5.1. Dataset and Evaluation

The local LaRS split used in our experiments contains 2605 training images, 198 development images, and 1203 hidden-test images for official benchmark evaluation. We use the development split for method iteration, ablation, and fusion-threshold selection. Final submission performance is reported with the official hidden-test metrics: PQ, $PQ_{th}$, $PQ_{st}$, RQ, SQ, and F1.

### 5.2. Implementation Details

All experiments were developed under constrained hardware. The MaskDINO runs use a single NVIDIA RTX A4000 GPU with batch size 1 and image size 896. The rare-thing fine-tuning stage runs for 16k iterations, and the thing-aware stage runs for 8k additional iterations starting from the best rare-thing checkpoint. The RF-DETR-Seg Medium model is trained at resolution 432 with batch size 1 and gradient accumulation of 8, then evaluated with the EMA checkpoint at inference resolution 720. The final fusion stack uses the global thresholds described above, followed by class-specific GrabCut overrides for classes 12–17 and 19. Here, *resource-efficient* refers to development feasibility on a single work-

| Thing class | Train imgs | Train inst. |
|---|---|---|
| Boat/ship | 1491 | 5921 |
| Row boat | 232 | 444 |
| Paddle board | 110 | 154 |
| Buoy | 734 | 1568 |
| Swimmer | 122 | 349 |
| Animal | 72 | 361 |
| Float | 21 | 23 |
| Other | 281 | 501 |

Table 1. Long-tail frequency of LaRS thing classes in the training split. The rarest classes have very limited supervision, motivating our rebalancing and thing-focused design choices.

station GPU rather than to a deployment-optimized real-time system. We therefore do not report end-to-end FPS for the final pipeline, because the submitted method combines a two-checkpoint panoptic ensemble, horizontal-flip augmentation, an auxiliary detector branch, and GrabCut-based post-processing, all of which were tuned for benchmark accuracy rather than throughput.

### 5.3. Class Difficulty Analysis

Table 1 summarizes the long-tail nature of thing categories. Classes such as animal and float are extremely rare in the training set, which supports our claim that the main difficulty lies less in general scene parsing than in long-tail thing segmentation.

## 6. Results

### 6.1. Main Results

Table 2 reports both development-set and official benchmark results. The development rows show how the submitted system was assembled, while the official row gives the final blind evaluation. Relative to the base MaskDINO model, the final system gains 8.29 PQ points on the development split, including an 11.02-point increase in $PQ_{th}$. By contrast, $PQ_{st}$ changes by less than one point across the full pipeline. This pattern again points to thing segmentation, rather than stuff parsing, as the main source of remaining error on the development data.

The numbers also clarify where the gains come from. The panoptic model already handles the globally dominant stuff regions and provides a reasonable obstacle layout, so the later stages can concentrate on underrepresented thing instances. The biggest jumps appear in the detector-guided stages: global RF-DETR fusion lifts PQ from 41.14 to 43.68, and global GrabCut refinement raises it again to 45.58. Class-specific policies contribute smaller but repeatable gains, with the clearest increments coming from buoy and swimmer refinement.

| Split | Method | PQ | $PQ_{th}$ | $PQ_{st}$ | Notes |
|---|---|---|---|---|---|
| Dev | Base MaskDINO | 37.50 | 17.47 | 90.93 | single checkpoint |
| Dev | + panoptic branch refinement | 39.99 | 20.88 | 90.93 | rebalance + crop tuning |
| Dev | + panoptic ensemble | 41.14 | 22.15 | 91.78 | v6 + v31, scale 896 + flip |
| Dev | + global RF-DETR fusion | 43.68 | 25.60 | 91.89 | Medium, res 720 |
| Dev | + global GrabCut refinement | 45.58 | 28.20 | 91.92 | margin 12, iters 2 |
| Dev | + class-specific refinement | **45.79** | **28.49** | **91.92** | final dev system |
| Official | Final submission | 42.6 | 24.2 | 91.8 | RQ=51.1, SQ=71.0, F1=54.4 |

Table 2. Main results. The development split is used for method analysis, while the official row reports the final anonymous challenge submission. The final system combines a panoptic ensemble, RF-DETR-guided fusion, and class-specific GrabCut refinement.

| Variant | PQ | $PQ_{th}$ | ΔPQ |
|---|---|---|---|
| Panoptic ensemble only | 41.14 | 22.15 | – |
| + global fusion only | 43.68 | 25.60 | +2.54 |
| + global GrabCut refinement | 45.58 | 28.20 | +1.90 |
| + class-specific on 14,15 | 45.71 | 28.38 | +0.13 |
| + extend to 12,13 | 45.77 | 28.46 | +0.06 |
| + conservative 16,17,19 | 45.79 | 28.49 | +0.02 |

Table 3. Ablation of the final refinement stack on the development split. Gains are measured relative to the preceding row. The main improvements come from global fusion and global GrabCut refinement, while class-specific policies provide smaller but consistent final gains.

### 6.2. Ablation of the Refinement Policy

Table 3 shows that global fusion alone is not enough. The largest increment after detector insertion comes from refining detector boundaries before merge decisions, indicating that recall and mask quality must be improved together. The final class-specific steps are much smaller in magnitude, but they remain useful because the best operating point differs for thin classes such as row boats and paddle boards and for buoy- and swimmer-like instances.

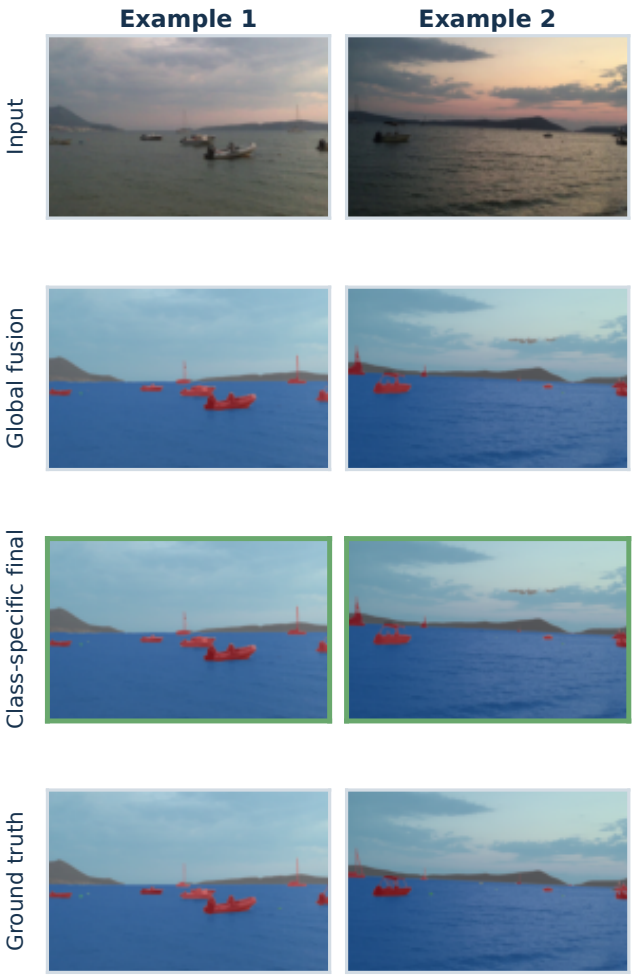


Figure 2. Qualitative comparison on validation images. The two columns show selected validation examples, while the rows show the input crop, the global-fusion baseline, the class-specific final prediction, and the ground-truth overlay. The class-specific final row is outlined in green.

### 6.3. Qualitative Analysis

Figure 2 shows two validation examples that reflect the same trend. In both cases, the globally fused baseline already captures the dominant water/sky layout, and class-specific refinement improves selected thing regions without changing the overall scene structure. This is in line with the quantitative results: most of the gain comes from targeted corrections on thing regions, not from large changes in stuff prediction.

The official benchmark result of 42.6 PQ and 24.2 $PQ_{th}$ is lower than the development score, possibly due in part to blind-test distribution shift, but the overall picture is unchanged: the final system is solid on stuff and remains limited mainly by rare thing segmentation.

## 7. Discussion

Our results suggest that USV panoptic segmentation is not uniformly hard across all parts of the label space. In LaRS, a strong panoptic model already handles stuff comparatively well. The harder cases are the long-tail thing categories. That is why a detector-style auxiliary model and targeted refinement help in ways that pure panoptic scaling does not fully recover. In that sense, a staged accuracy-first design remains a sensible option when memory and compute make end-to-end scaling expensive.

The study also exposes the limits of our current approach. First, the class-specific refinement policy is manually tuned on a small development split and may not transfer perfectly across domains. Second, the RF-DETR booster still struggles on the rarest categories and under severe appearance changes. Third, our system is image-based; restricted temporal experiments plateaued below the best class-specific refinement configuration and were therefore excluded from the final submission. These limitations point to clear next steps: learned refinement policies, stronger rare-class supervision, and more reliable temporal propagation.

## 8. Conclusion

We presented a hybrid framework for maritime panoptic segmentation that combines a MaskDINO panoptic base model, thing-aware fine-tuning, a panoptic ensemble, RF-DETR-guided fusion, and class-specific GrabCut refinement. The method is built around the part of LaRS that remains hardest: small and rare thing instances. On the development split, the largest gains come from detector-guided refinement, and the final anonymous submission reaches 42.6 PQ with 24.2 $PQ_{th}$ and 91.8 $PQ_{st}$ on the official hidden-test benchmark. Our main takeaway is straightforward: under single-workstation-GPU constraints, a hybrid pipeline can be a better use of limited compute than relying only on a larger unified panoptic model.

## References

[1] Borja Bovcon, Jon Muhovič, Dušan Vranac, Dean Mozetič, Janez Perš, and Matej Kristan. MODS: A USV-oriented object detection and obstacle segmentation benchmark. *arXiv preprint arXiv:2105.02359*, 2021. 2

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 2

[3] Bowen Cheng, Alexander Schwing, and Alexander Kirillov. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1290–1299, 2022. 1, 2

[4] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Pro-*

*ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9404–9413, 2019. 1, 2

[5] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M. Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3041–3050, 2023. 1, 2

[6] Isaac Robinson, Peter Robicheaux, Matvei Popov, Deva Ramanan, and Neehar Peri. RF-DETR: Neural architecture search for real-time detection transformers. *arXiv preprint arXiv:2511.09554*, 2025. 2

[7] Leon Amadeus Varga, Benjamin Kiefer, Martin Messmer, and Andreas Zell. Seadronessee: A maritime benchmark for detecting humans in open water. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2260–2270, 2022. 2

[8] Lojze Žust, Janez Perš, and Matej Kristan. Lars: A diverse panoptic maritime obstacle detection dataset and benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20304–20314, 2023. 1, 2