# FishDetector-R1: Unified MLLM-Based Framework with Reinforcement Fine-Tuning for Weakly Supervised Fish Detection, Segmentation, and Counting

Yi Liu Jingyu Song Vedanth Kallakuri
Katherine A. Skinner
University of Michigan, Ann Arbor, MI, USA
{yiliuhh, jingyuso, vkallaku, kskin}@umich.edu

## Abstract

*Analyzing underwater fish imagery is critical for ecological monitoring but remains difficult due to visual degradation and costly annotations. We introduce* ***FishDetector-R1****, a unified multimodal large language model (MLLM)-based framework for fish detection, segmentation, and counting under weak supervision. On the DeepFish dataset, our framework achieves substantial gains over baselines, improving AP by 20% and mIoU by 10%, while reducing MAE by 30% and GAME error by 35% on counting and localization tasks. These improvements stem from two key components: a novel* ***detect-to-count*** *prompt that encourages spatially consistent detections and counts, and Reinforcement Learning from Verifiable Reward (****RLVR****) with a complementary paradigm that leverages sparse point labels. Ablation studies further validate the effectiveness of our complementary reward design, which jointly optimizes detection and counting. Together, these findings demonstrate that our novel framework provides a reliable solution to enable multimodal large language models to achieve scalable, accurate marine visual understanding via weak supervision.*

## 1. Introduction

Understanding marine life through visual data is essential for ecological monitoring, fisheries management, and underwater exploration [28]. Effective analysis requires not only detecting fish but also performing instance-level segmentation and estimating their counts, which supports downstream tasks such as species identification, behavioral analysis, and habitat mapping. However, these tasks remain particularly challenging in underwater imagery, where low visibility, color distortion, and light scattering severely degrade the performance of conventional vision models.

Over the past decade, a range of learning-based approaches have been proposed to address these challenges. Fully supervised instance segmentation methods achieve strong performance but rely heavily on large-scale, densely annotated datasets [2, 5, 22, 34]. The high cost and labor intensity of generating pixel-wise annotations make such approaches difficult to scale in underwater environments [16, 17, 25]. As a more annotation-efficient alternative, point-level weak supervision offers significant advantages in terms of speed and scalability [4, 15]. However, existing weakly supervised methods based on point annotations often suffer from a substantial performance gap relative to fully supervised models because sparse points provide limited pixel-level guidance [14, 27]. This leaves a key question: how can we close the weak-to-dense performance gap in challenging underwater settings while still relying only on sparse, scalable point-level labels?

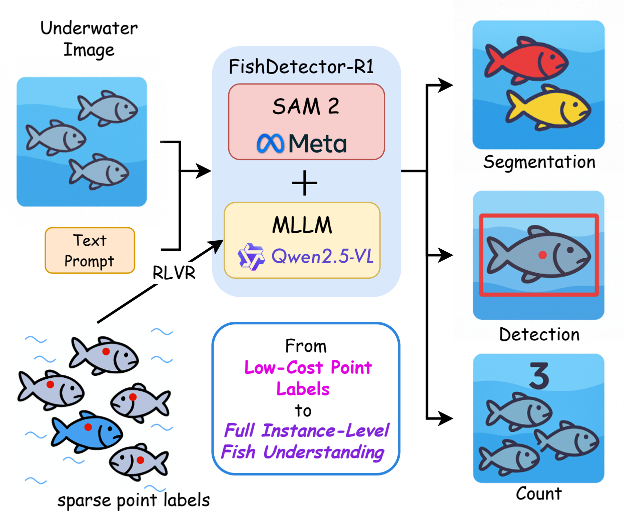


Figure 1. Our proposed FishDetector-R1 aims to achieve AI-enabled fish image analysis with the guidance of sparse point labels and text prompts.

We address this gap with two complementary ingredients. First, we find that foundation models are well positioned to fill this gap due to their transferable visual understanding capabilities from large-scale pretraining. Multimodal large language models (MLLMs) like GPT-4 series [1, 12, 21], Qwen2.5-VL series [3] and Gemini series [6, 29] combine rich semantic knowledge with strong reasoning capability, while segmentation foundation models such as SAM [13, 24] can complement them by providing robust semantic priors for accurate mask generation from sparse prompts, enabling effective deployment in visually challenging domains.

Second, we develop an effective framework that tailors the MLLM to the specific challenges of underwater fish visual analysis. We propose a novel joint detect–to-count task formulation that turns sparse point labels into consistent, verifiable reward signals to enforce spatial alignment between the predicted detection and counting number. Building on recent successes of Reinforcement Learning from Verifiable Rewards (RLVR) for adapting foundation models [18–20, 32, 33], we fine-tune the MLLM under this detect–to-count objective, yielding mutually reinforcing gains in detection and counting while supplying precise spatial priors that guide segmentation mask generation effectively. To the best of our knowledge, we are the first to integrate an MLLM with a segmentation foundation model to tackle scalable marine fish visual analysis—covering detection, instance segmentation, and counting—using only weak point-level supervision.

Together, these two ingredients constitute **FishDetector-R1** (Fig. 1), a unified framework for detection, segmentation, and counting from weak point-level supervision. FishDetector-R1 moves beyond prior approaches that treat these tasks in isolation and yields concurrent improvements across all three tasks. To summarize, our contributions are as follows:

1. We propose FishDetector-R1, the first unified framework to integrate an MLLM with a segmentation foundation model for comprehensive marine fish analysis (detection, segmentation, and counting) using only weak, point-level supervision.
2. We design a novel joint detect–to-count learning paradigm to adapt foundation models to the challenging underwater domain in a complementary manner. By formulating sparse point labels as verifiable rewards within an RLVR framework, our method enforces spatial and numerical consistency, enabling the generation of high-quality masks from minimal annotation.
3. We conduct extensive quantitative and qualitative experiments on the DeepFish dataset [22] to demonstrate the effectiveness of our complementary reward design, and for the first time, demonstrate performance competitive with and even exceeding fully supervised methods.

## 2. Related Work

### 2.1. Fish Detection in Underwater Scenes

Fully supervised segmentation methods [7, 16, 34] achieve high accuracy in underwater scenes by training on dense pixel-wise annotations. However, such labels are time-consuming and expensive to obtain, especially in underwater imagery where object boundaries are often ambiguous [16, 25]. To reduce annotation cost, weakly supervised approaches [14, 15, 27] use point-level labels, which are significantly faster to collect [4], but typically yield lower segmentation performance due to the lack of dense spatial supervision. This results in a clear gap in mask quality between fully and weakly supervised models. While prior methods make progress in annotation efficiency, none have successfully closed this performance gap on challenging underwater segmentation tasks. In contrast, **FishDetector-R1** is the first framework that effectively leverages only point-level supervision to achieve high-quality instance segmentation, matching and even surpassing fully supervised baselines on the DeepFish benchmark.

### 2.2. Visual Foundation Models

Visual foundation models such as SAM [13] and SAM 2 [24] provide flexible segmentation from simple prompts like points or bounding boxes and demonstrate strong generalization across diverse visual domains. Their ability to operate in a zero-shot setting has made them attractive for domains with limited labels. Recent adaptations to underwater imagery, such as AquaSAM [31] and WaterSAM [11], attempt to specialize these models by either freezing encoders or introducing lightweight adapters to improve segmentation under challenging visual conditions like turbidity and color distortion. While effective, their reliance on dense supervision limits their scalability and practicality in annotation-scarce scenarios. In contrast, our method leverages the reasoning capability of MLLMs together with reinforcement fine-tuning, enabling joint detection, segmentation, and counting with only sparse point-level labels.

### 2.3. Multimodal Large Language Models

MLLMs such as GPT-4.1 [21], Llama [30], and Qwen2.5-VL [3] combine visual perception with natural language reasoning, enabling capabilities such as object grounding, spatial reasoning, and prompt-based visual interaction. These models have shown promise in general domains for tasks like zero-shot grounding and segmentation, by aligning semantic priors from natural language instructions with visual content. However, their application to underwater imagery remains largely unexplored, despite the fact that underwater monitoring often requires high-level reasoning to distinguish between subtle visual cues. In this work, we adapt an open-source MLLM using weak point-level supervision, allowing it to generate reliable bounding boxes and keypoints under noisy underwater conditions. These semantic priors are then used to guide SAM 2 for instance-level segmentation, bridging the gap between high-level reasoning and fine-grained perception in an annotation-efficient manner.

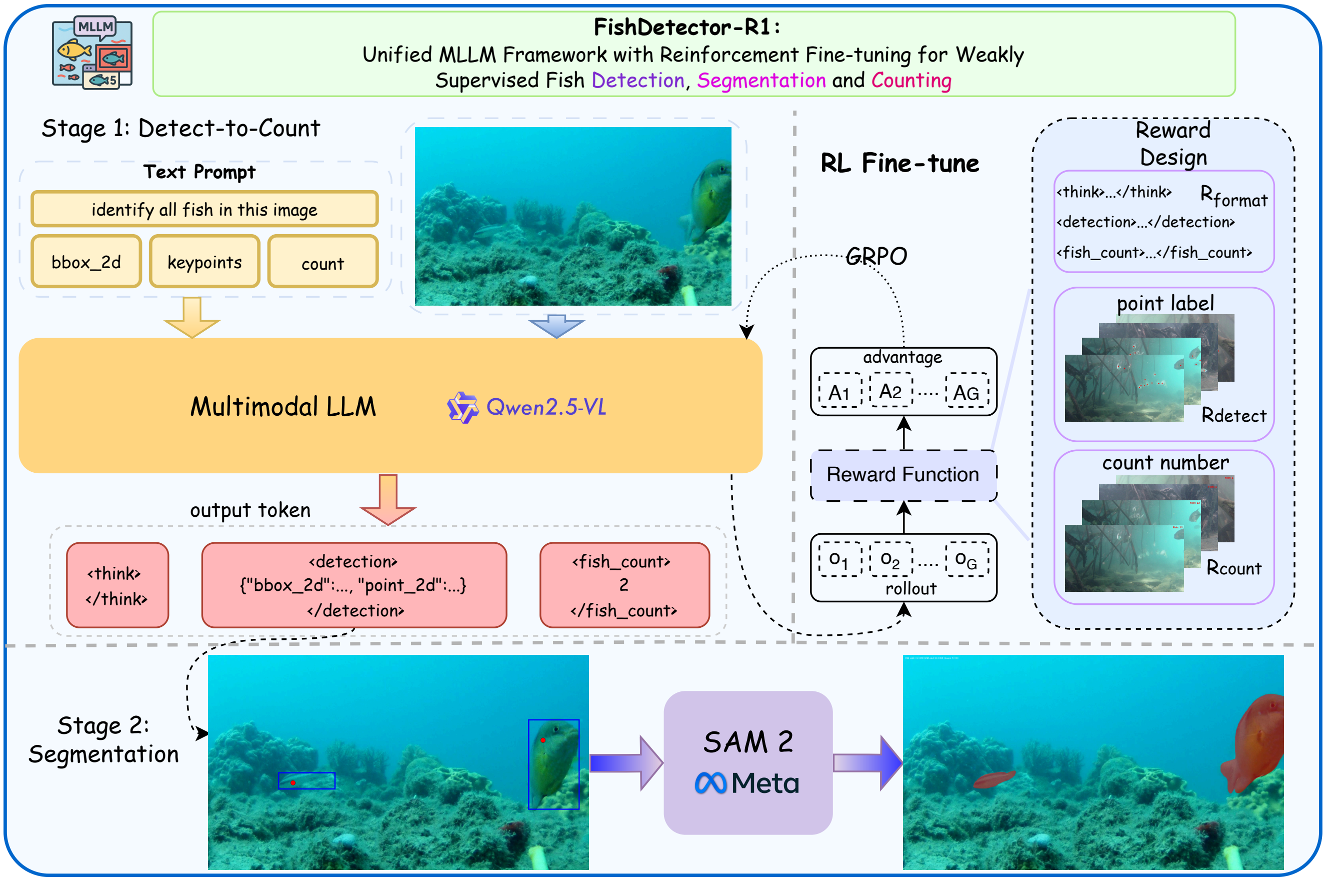


Figure 2. **Overview of the proposed FishDetector-R1 framework**. A two-stage detect-to-count pipeline integrates an MLLM with SAM 2 to jointly perform detection, segmentation, and counting. Reinforcement fine-tuning with GRPO and weak point-level supervision adapts the MLLM, ensuring consistency between detection and counting while enabling pixel-wise segmentation with only sparse labels.

### 2.4. Reinforcement Learning for Multimodal Large Language Models

Reinforcement Learning from Verifiable Reward (RLVR) has emerged as an effective strategy to improve the reasoning, alignment, and perception capabilities of both language models and multimodal models [35].While traditional methods like PPO [26] and DPO [23] are widely adopted, more recent approaches such as GRPO [9] offer improved stability and efficiency via group preference optimization. Building on this, frameworks like Perception-R1 [33], Seg-R1 [32], and VisionReasoner [19] extend RL fine-tuning to multimodal perception, showing strong results in detection, segmentation, and counting. However, these works typically treat each task in isolation with separate reward functions, and focus on general-domain benchmarks, leaving domain-specific settings like underwater imagery underexplored. Our work addresses this gap by applying GRPO with point-level supervision and a unified reward design that jointly couples detection and counting. This enables all three tasks—detection, segmentation, and counting—to reinforce one another, improving adaptation under weak supervision.

## 3. Methodology

### 3.1. Overview

We propose a two-stage framework, **FishDetector-R1**, that integrates an MLLM (Qwen2.5-VL [3]) with a segmentation foundation model (SAM 2-Large [24]) for underwater fish detection, segmentation, and counting as illustrated in Fig. 2. In the first stage, guided by a detect-to-count prompt, Qwen2.5-VL takes an input image, localizes each fish with a bounding box and keypoint, and then derives the total count from its detections, promoting consistency between localization and counting. In the second stage, these spatial priors are passed to SAM 2 to generate high-resolution pixel-wise instance masks. To further adapt the framework to underwater imagery, we apply RL fine-tuning to Qwen2.5-VL with weak point labels. This training step precedes SAM 2, ensuring that the MLLM learns to generate spatially consistent detections and counts, which then serve as strong priors for segmentation. Unlike conventional pipelines that combine supervised fine-tuning (SFT) with reinforcement learning, we directly adopt RL fine-tuning. This choice is motivated by findings in recent work such as Perception-R1 [33], which shows that RL with task-aligned rewards can be more effective than SFT in perception tasks, while also avoiding additional annotation costs. As a result, our framework delivers improved detection precision and counting reliability, while also producing refined pixel-wise masks from only sparse point annotations, offering both accuracy and annotation efficiency.

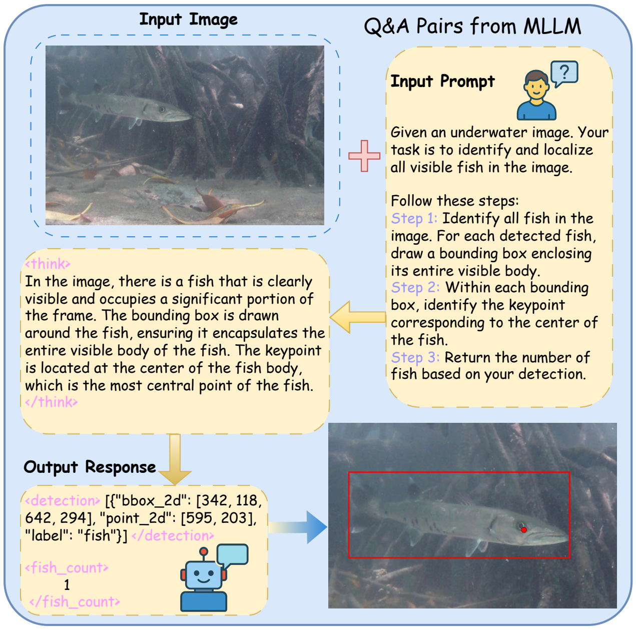
## 3.2. Prompt Design

To support joint detection, segmentation, and counting, we design a structured prompt tailored for **Qwen2.5-VL**, an MLLM with strong grounding and reasoning capabilities. As shown in Fig. 3, given an underwater RGB image, the model is prompted to first localize each fish instance with the total count directly derived from detections, following a detect-to-count strategy. This formulation encourages the model to understand that reliable counting depends on accurate localization, i.e., it must "know where the fish are" before reporting how many fish there are (example in Fig. 4). The resulting detection outputs – bounding boxes and keypoints – are also passed as spatial priors to **SAM 2**, enabling high-quality instance segmentation. In this way, the prompt design unifies all three tasks within a single pipeline.

We adopt a structured output format composed of three distinct components: `<think>`, `<detection>`, and `<fish_count>`.

- The `<think>` field records the model's internal reasoning and visual understanding process.
- The `<detection>` field contains structured outputs for each fish instance, including a bounding box and a central keypoint, which both support counting and serve as effective prompts for SAM 2.
- The `<fish_count>` field provides the total number of fish, derived from the detections to ensure consistency between localization and counting.

This design enforces a detect-to-count reasoning process, provides explicit spatial cues to guide segmentation, and ensures response completeness. Furthermore, during RL fine-tuning, the predicted count is compared against weak point-level annotations to construct reward signals, aligning detection and counting objectives without requiring dense labels.



Given an underwater image. Your task is to identify and localize all visible fish in the image.

Follow these steps:
Step 1: Identify all fish in the image. For each detected fish, draw a bounding box enclosing its entire visible body.
Step 2: Within each bounding box, identify the keypoint corresponding to the center of the fish.
Step 3: Return the number of fish based on your detection.

<think>
In the image, there is a fish that is clearly visible and occupies a significant portion of the frame. The bounding box is drawn around the fish, ensuring it encapsulates the entire visible body of the fish. The keypoint is located at the center of the fish body, which is the most central point of the fish.
</think>

<detection> [{"bbox_2d": [342, 118, 642, 294], "point_2d": [595, 203], "label": "fish"}] </detection>

<fish_count>
1
</fish_count>

Figure 3. Example Q&A pairs from Qwen2.5-VL using our designed prompt.

## 3.3. Group Relative Policy Optimization

Following recent RL fine-tuning work on MLLMs [18, 19, 32], we adopt Group Relative Policy Optimization (GRPO) [9] as our post-training strategy. GRPO is an efficient reinforcement learning framework that removes the need for a separate critic by directly comparing the relative quality of responses within a group. Given a task input $t$, the current policy $\pi_{\theta_{\text{old}}}$ generates a set of $G$ candidate responses $\{o_1, o_2, \dots, o_G\}$ with corresponding rewards $\{r_1, r_2, \dots, r_G\}$. These rewards are normalized within the group to compute relative advantages, which are then used to update the policy. This group-wise formulation provides more stable optimization while reducing training costs compared to traditional actor–critic methods.

The GRPO objective function is defined as:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{o_i \sim \pi_{\theta_{\text{old}}}} \Bigg[ \frac{1}{G} \sum_{i=1}^{G} \min \Bigg( \frac{\pi_\theta(o_i|t)}{\pi_{\theta_{\text{old}}}(o_i|t)} \hat{A}_i, \\ \text{clip} \left( \frac{\pi_\theta(o_i|t)}{\pi_{\theta_{\text{old}}}(o_i|t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_i \Bigg) \Bigg] \\ - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \end{aligned} \quad (1)$$

where $\epsilon$ is the clipping threshold, $\beta$ is the coefficient of the KL penalty, and $\hat{A}_i$ denotes the normalized advantage for response $o_i$, computed as:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \dots, r_G\})}{\text{std}(\{r_1, \dots, r_G\})} \quad (2)$$

By leveraging group-wise comparisons and reward normalization, GRPO enables stable and sample-efficient policy optimization purely based on relative preferences.

## 3.4. Reward Design

The overall reward function used for RL fine-tuning consists of four components: (1) a format reward, (2) an accuracy reward, (3) a count reward, and (4) a non-repetition reward. Each component is designed to encourage the model to produce syntactically valid, semantically accurate, and non-redundant outputs that align with weak point-level supervision.

**(1) Format Reward.**
The format reward $R_{\text{format}}$ has two sub-parts: response structure formatting and detection content formatting.

- The response is required to contain three structured tags: `<think>`, `<detection>`, and `<fish_count>`. A correct structural response yields 1.0 reward.
- The content within the `<detection>` tag must follow the format: {`"bbox_2d": [x1, y1, x2, y2], "point_2d": [x, y], "label": "fish"`}. If all predicted instances match this structure, the model receives up to 3.0 additional reward points.

The total format reward is therefore bounded by $R_{\text{format}} \leq 4.0$.

**(2) Detection Accuracy Reward.**
To encourage correct detection and precise localization, we design an accuracy reward $R_{\text{detect}}$. Predicted keypoints are matched to ground-truth points using the Hungarian algorithm within a Euclidean distance threshold. A prediction is considered valid if its distance to a ground-truth point is within a predefined threshold. The reward is defined as:

$$\text{accuracy\_reward} = \lambda_{\text{detect}} \cdot \left( \frac{N_{\text{valid}}}{N_{\text{gt}}} \right), \tag{3}$$

where $N_{\text{valid}}$ denotes the number of matched predictions and $N_{\text{gt}}$ the total number of ground-truth fish. $\lambda_{\text{detect}}$ specifies the maximum reward assigned to the accuracy performance, which is set to 4.0 empirically. The accuracy reward scales proportionally with the fraction of correctly matched instances.

In addition, we enforce consistency between the number of detected instances $N_{\text{pred}}$ and the reported count $N_{\text{count}}$ in the `<fish_count>` tag by introducing a match reward:

$$\text{match\_reward} = \begin{cases} 0, & \text{if } N_{\text{pred}} = N_{\text{count}}, \\ -1, & \text{otherwise.} \end{cases} \tag{4}$$

The overall detection-related reward is then computed as:

$$R_{\text{detect}} = \text{accuracy\_reward} + \text{match\_reward}. \tag{5}$$

This formulation jointly optimizes detection and counting: accurate localization improves counting reliability, while consistent counting further encourages complete detection.

**(3) Count Reward.**
To further enforce correct enumeration, a count reward $R_{\text{count}}$ is assigned based on whether the predicted count matches the number of ground-truth instances:

$$R_{\text{count}} = \begin{cases} 1, & \text{if } N_{\text{count}} = N_{\text{gt}} \\ -1, & \text{otherwise} \end{cases} \tag{6}$$

where $N_{\text{count}}$ is the number of fish reported by the model and $N_{\text{gt}}$ the total number of ground-truth fish.

**(4) Non-Repetition Reward.**
To mitigate repetitive responses and promote output diversity, we adopt a non-repetition reward $R_{\text{non-repeat}}$ inspired by Seg-Zero [18].

**(5) Total Reward.**
The total reward used for RL optimization is defined as:

$$R_{\text{total}} = R_{\text{format}} + \alpha \cdot R_{\text{detect}} + \beta \cdot R_{\text{count}} + R_{\text{non-repeat}} \tag{7}$$

where $\alpha$ and $\beta$ control the relative weight of detection and counting rewards. This formulation jointly accounts for syntactic correctness, localization accuracy, count fidelity, and output diversity, thereby providing rich supervision signals at minimal annotation cost.

In practice, the absolute scale of rewards has little effect, while the relative balance between components is key to final performance—consistent with GRPO's use of groupwise relative advantages over absolute magnitudes.

## 4. Experiments

### 4.1. Evaluation Metrics

We evaluate our framework on two main capabilities: **(1) grounding**, which includes detection and segmentation, and **(2) counting**, which includes count accuracy and localization.

**(a) Grounding (detection, segmentation).** We evaluate grounding performance on the test split of the DeepFish segmentation subset. For segmentation, we measure mean Intersection-over-Union (mIoU) between predicted masks and ground-truth annotations. For detection, we follow the COCO evaluation protocol [17] and report Average Precision (AP) and Average Recall (AR) across multiple IoU thresholds. Here we report the value of $\text{AP}_{0.5:0.95}$ and $\text{AR}_{0.5:0.95}$, representing the mean AP and AR computed at IoU thresholds from 0.5 to 0.95. Together, these metrics provide a comprehensive assessment of grounding ability, capturing both mask quality and instance-level accuracy.

**(b) Counting (count accuracy, localization).** We utilize the test split of the DeepFish localization subset to evaluate counting performance. To measure counting accuracy, we employ several complementary metrics. Mean Absolute Error (MAE) quantifies overall number prediction accuracy while the Match Rate evaluates the consistency between the predicted and ground-truth counts. Given predicted counts $\hat{y}_i$ and ground-truth counts $y_i$ for $N$ images, the MAE and Match Rate are calculated as:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|, \tag{8}$$

| Model | Detection | | Segmentation | | | Counting | | |
|---|---|---|---|---|---|---|---|---|
| | **AP$_{0.5:0.95}$ ↑** | **AR$_{0.5:0.95}$ ↑** | **Foreground ↑** | **Background ↑** | **mIoU ↑** | **MAE ↓** | **Match Rate ↑** | **GAME ↓** |
| | | | | **Baseline** | | | | |
| GPT-4.1 | 5.47 | 17.43 | 43.77 | 98.69 | 71.23 | 0.387 | **0.796** | 1.394 |
| Gemini-2.0-flash | 54.60 | 61.24 | 84.01 | 99.64 | 91.83 | 1.879 | 0.582 | 3.434 |
| Gemini-2.5-flash | 24.10 | 46.46 | 57.14 | 98.85 | 78.00 | 2.228 | 0.568 | 3.081 |
| Qwen2.5-VL 3B | 44.63 | 62.12 | 45.10 | 97.64 | 71.67 | 0.604 | 0.616 | 1.136 |
| Qwen2.5-VL 7B | 48.05 | 55.84 | 81.25 | 99.58 | 90.42 | 0.579 | 0.706 | 0.915 |
| | | | | **Ours** | | | | |
| FishDetector-Base 3B | 53.58 | 62.21 | 67.88 | 99.10 | 83.49 | 0.647 | 0.691 | 0.901 |
| FishDetector-R1 3B | **61.71** | **66.64** | 86.47 | 99.69 | 93.08 | **0.386** | 0.760 | 0.613 |
| FishDetector-Base 7B | 47.52 | 58.67 | 80.86 | 99.56 | 90.21 | 0.497 | 0.715 | 0.924 |
| FishDetector-R1 7B | 60.71 | 63.63 | **87.90** | **99.78** | **93.84** | 0.398 | 0.765 | **0.587** |

Table 1. Unified comparison of detection (AP$_{0.5:0.95}$, AR$_{0.5:0.95}$), segmentation (Foreground, Background, mIoU), and counting (MAE, Match Rate, GAME) performance across baseline and proposed MLLM variants. The **best** result in each column is shown in bold, and the second best is underlined.

$$\text{Match Rate} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i)\,, \tag{9}$$

where $\mathbb{1}(\cdot)$ is the indicator function that equals 1 if the predicted count exactly matches the ground truth and 0 otherwise.

In addition, we report the Grid Average Mean Absolute Error (GAME) [8] to compute counting errors at different spatial scales, where each image is divided into $4^L$ non-overlapping grids at level $L$. The error is then computed over all sub-regions:

$$\text{GAME}(L) = \frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{4^L} \left|\hat{y}_i^r - y_i^r\right|\,, \tag{10}$$

$$\text{GAME} = \frac{1}{4}\sum_{L=1}^{4} \text{GAME}(L), \tag{11}$$

where $\hat{y}_i^r$ and $y_i^r$ denote the predicted and ground-truth counts in the $r$-th region of the $i$-th image. Lower values of GAME indicate better spatial consistency in counting predictions, reducing cases where totals are correct but fish are mislocalized.

## 4.2. Model and Implementation

We build our framework based on the 3B and 7B variants of Qwen2.5-VL. For **FishDetector-Base**, we use the frozen pretrained Qwen2.5-VL with our detect-to-count prompt design. For **FishDetector-R1**, we apply GRPO fine-tuning on the training split of the DeepFish localization subset [22], which contains 1,600 images with point-level fish annotations. Fine-tuning is performed on 4 × NVIDIA A100 GPUs, with a batch size of 16 per device and 8 rollouts per input. Training is conducted for 4 epochs (about 400 optimization steps in total) with a learning rate of $1 \times 10^{-6}$.

For baselines, we adopt both open-source and closed-source MLLMs, including GPT-4.1, Gemini-2.0/2.5, and Qwen2.5-VL, with their publicly available prompting strategies. We provide detailed prompt implementation for Qwen2.5-VL and our method in the supplementary material. All methods, including baselines and ours, are evaluated under the same resolution setting by rescaling predictions back to the original image size (1920 × 1080) to allow direct comparison with ground truth.

## 4.3. Experimental Results

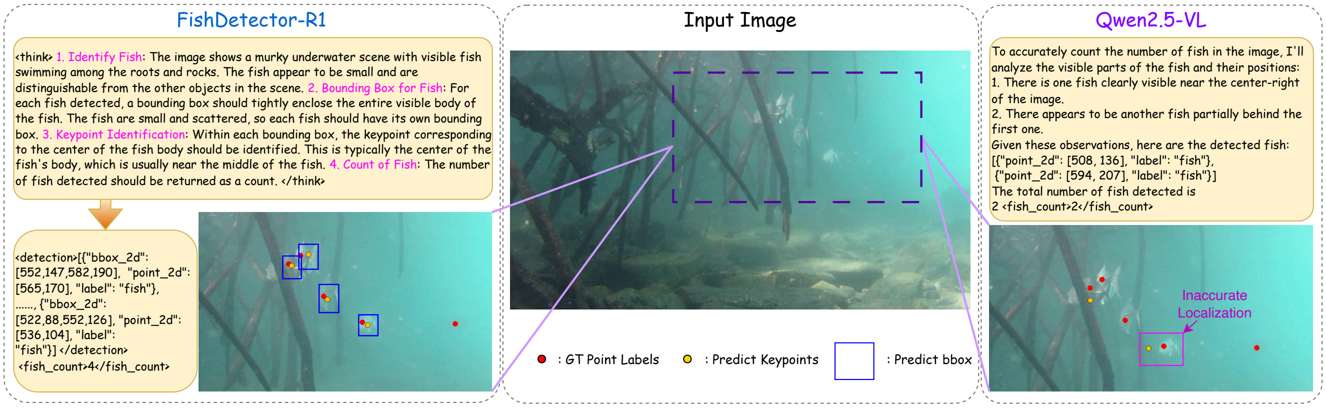
### (a) Baseline Comparison

We benchmark our framework against strong foundation models, including GPT-4.1, Gemini-2.0/2.5, and Qwen2.5-VL, across detection, segmentation, and counting. As summarized in Tab. 1, Qwen2.5-VL consistently achieves the strongest zero-shot grounding and counting performance among open-source and closed-source MLLMs, underscoring its suitability as a foundation model for real-world visual applications, motivating its selection as the backbone of our framework.

On top of Qwen2.5-VL, we further investigate two variants of our framework: **FishDetector-Base** and **FishDetector-R1**. This design enables a controlled evaluation of the contributions from prompt design and reinforcement learning, thereby clarifying how each component advances detection, segmentation, and counting under weak supervision. Detailed ablation studies are provided in Sec. 4.4, and additional qualitative analyses are included in the supplementary material.

### (b) FishDetector-Base: Prompt-Only Adaptation

FishDetector-Base uses our detect-to-count prompt without gradient-based fine-tuning. This lightweight adaptation yields notable improvements, particularly for the 3B model. Compared to Qwen2.5-VL 3B, it boosts detection AP (44.6 → 53.6), segmentation mIoU (71.7 → 83.5), and counting accuracy (GAME: 1.14 → 0.90; Match Rate: 0.62 → 0.69). These results show that well-designed



<think> 1. Identify Fish: The image shows a murky underwater scene with visible fish swimming among the roots and rocks. The fish appear to be small and are distinguishable from the other objects in the scene. 2. Bounding Box for Fish: For each fish detected, a bounding box should tightly enclose the entire visible body of the fish. The fish are small and scattered, so each fish should have its own bounding box. 3. Keypoint Identification: Within each bounding box, the keypoint corresponding to the center of the fish body should be identified. This is typically the center of the fish's body, which is usually near the middle of the fish. 4. Count of Fish: The number of fish detected should be returned as a count. </think>

```
<detection>[{"bbox_2d":
[552,147,582,190],  "point_2d":
[565,170], "label": "fish"},
......, {"bbox_2d":
[522,88,552,126], "point_2d":
[536,104], "label":
"fish"}] </detection>
 <fish_count>4</fish_count>
```

To accurately count the number of fish in the image, I'll analyze the visible parts of the fish and their positions:
1. There is one fish clearly visible near the center-right of the image.
2. There appears to be another fish partially behind the first one.
Given these observations, here are the detected fish:

```
[{"point_2d": [508, 136], "label": "fish"},
 {"point_2d": [594, 207], "label": "fish"}]
```

The total number of fish detected is
2 <fish_count>2</fish_count>

Figure 4. **Qualitative Comparison between Qwen2.5-VL and FishDetector-R1.** On a challenging scene from DeepFish *FishLoc*, our detect-to-count strategy enables more accurate localization and structured outputs.

prompts can enhance smaller models by better aligning detection with counting and segmentation. In contrast, FishDetector-Base 7B sees a slight drop on segmentation quality (mIoU: 90.42 → 90.21), suggesting that larger models benefit less from prompt-only adaptation and may require additional alignment.

**(c) FishDetector-R1: Reinforcement Fine-Tuning**
Building on FishDetector-Base, **FishDetector-R1** applies reinforcement learning with a reward function that jointly optimizes detection and spatially grounded counting. This fine-tuning leads to consistent improvements across all tasks. For the 3B variant, R1 raises AP to 61.7 and mIoU to 93.1, both the highest among 3B models, while reducing MAE from 0.65 to 0.39 and GAME to 0.61. The 7B model also improves notably, achieving 60.7 AP, 93.8 mIoU, and a best-in-class 0.59 GAME, effectively recovering from the slight degradation seen in its Base counterpart. Figure 4 presents a comparison with the original Qwen2.5-VL in a challenging crowded scene, illustrating that FishDetector-R1 maintains stronger performance even under difficult conditions.

**(d) Comparison with Traditional Fully & Weakly Supervised Methods**
We further compare our FishDetector-R1 framework with traditional fully and weakly supervised methods. For the fully supervised setting, we report the results from the DeepFish benchmark [25], which utilizes pixel-wise dense annotations to train a segmentation model with a pretrained ResNet-50 [10] backbone. For the weakly supervised setting, we report the results of A-LCFCN [14], a point-supervised baseline trained on the same DeepFish *FishLoc* subset. As shown in Tab. 2, a substantial weak-to-dense performance gap exists (86.2 vs. 93.0 mIoU). Notably, our FishDetector-R1 framework closes this gap entirely. Despite relying solely on sparse point-level annotations, our 3B variant matches the fully supervised baseline with 93.1 mIoU, while the 7B variant surpasses it, achieving 93.8 mIoU. These results underscore the strength of our method in combining weak supervision with the spatial reasoning capabilities and semantic priors of foundation models. By leveraging multimodal prompting and reinforcement alignment, our approach outperforms traditional weakly supervised methods and rivals dense annotation-based models, offering a more scalable and annotation-efficient solution for underwater segmentation tasks.

| Method | Supervision Type | mIoU |
|---|---|---|
| DeepFish [25] | Dense Annotations | 93.0 |
| A-LCFCN [14] | Point Labels | 86.2 |
| FishDetector-R1 3B | Point Labels | <u>93.1</u> |
| FishDetector-R1 7B | Point Labels | **93.8** |

Table 2. Comparison of segmentation accuracy (mIoU) on DeepFish *FishSeg* dataset across different supervision methods. The **best** result in each column is shown in bold, and the <u>second best</u> is underlined.

## 4.4. Ablation Study

After comparing our framework with external baselines and its own variants, we conduct ablation studies to examine the role of each reward component across different tasks shown in Tab. 3, focusing on four key questions:

**Q1.** What is the impact of applying only the count reward $R_{\text{count}}$, and does it harm grounding quality while improving the global numerical accuracy?

**Q2.** How does single detection reward affect model performance and does $R_{\text{detect}}$ maintain reliable count predictions while simultaneously enhancing grounding ability?

**Q3.** What are the benefits of combining both rewards? Does the joint design yield complementary gains across detection, segmentation, and counting while mitigating the limitations of using a single reward in isolation?

**Q4.** How effective is the detect-to-count prompt in enforcing internal reasoning alignment between localized de-

| Reward Setting | Detection | | Segmentation | | | Counting | | |
|---|---|---|---|---|---|---|---|---|
| | $\mathbf{AP}_{0.5:0.95}$ | $\mathbf{AR}_{0.5:0.95}$ | **Foreground** | **Background** | **mIoU** | **MAE** | **Match Rate** | **GAME** |
| Base | 47.52 | 58.67 | 80.86 | 99.56 | 90.21 | 0.497 | 0.715 | 0.924 |
| $+R_{\text{count}}$ | 26.46 | 39.91 | 51.71 | 98.59 | 87.82 | 0.414 | **0.770** | 1.346 |
| $+R_{\text{detect}}$ | 57.10 | 63.00 | 87.10 | 99.68 | 93.41 | 0.442 | 0.757 | 0.693 |
| $+R_{\text{count}}$+$R_{\text{detect}}$ | **60.71** | **63.63** | **87.94** | **99.78** | **93.84** | **0.398** | 0.765 | **0.587** |

Table 3. Ablation study of FishDetector-R1 (7B) with different reward configurations. Detection ($\text{AP}_{0.5:0.95}$, $\text{AR}_{0.5:0.95}$), segmentation (Foreground, Background, mIoU), and counting (MAE, Match Rate, GAME) metrics are reported. The **best** result in each column is shown in bold, and the second best is underlined.

tections and global count predictions, and can reinforcement fine-tuning with verified rewards further enhance this consistency?

Results are reported for the 7B model as an example, with additional ablations provided in the supplementary material.

**Count Reward Only.** Applying only $R_{\text{count}}$ improves global numerical accuracy, as reflected by reduced MAE (0.414 vs. 0.497) and higher Match Rate (0.770 vs. 0.715) in Tab. 3. However, grounding ability collapses: AP and AR drop by over 20 points, mIoU decreases by 2.4, and GAME worsens from 0.924 to 1.346. This indicates that while $R_{\text{count}}$ enforces numerical regularity, it fails to provide spatial guidance, leading to mislocalized predictions that undermine detection and segmentation quality.

**Detection Reward Only.** Using only $R_{\text{detect}}$ substantially improves grounding performance, with AP increasing from 47.5 to 57.1, AR from 58.7 to 63.0, and mIoU from 90.2 to 93.4 (Tab. 3). Better localization also translates to stronger counting consistency, as GAME improves from 0.924 to 0.693. However, global count accuracy remains limited: MAE (0.442) is slightly worse than the count-only setting, and Match Rate (0.757) is lower. These results show that $R_{\text{detect}}$ excels at spatial precision but still lacks global numerical control.

**Joint Reward.** Combining $R_{\text{count}}$ and $R_{\text{detect}}$ delivers the most balanced improvements across tasks. Detection achieves the highest AP (60.7) and AR (63.6), while segmentation quality remains strong (mIoU 93.8). On the counting side, MAE (0.398) is the lowest, and GAME (0.587) shows the best spatial distribution of counts. This configuration preserves the global accuracy gains of $R_{\text{count}}$ while retaining the spatial precision of $R_{\text{detect}}$, demonstrating their complementary nature in achieving robust detection, segmentation, and counting under weak supervision.

| Model | Alignment Rate (%)↑ | |
|---|---|---|
| | **3B** | **7B** |
| FishDetector-Base | 97.2 | 98.6 |
| FishDetector-R1 | **99.6** | **100** |

Table 4. Alignment rate between detected instances and predicted fish counts from model output response. Higher is better.

**Internal Detect-to-Count Consistency.** Tab. 4 reports the alignment between detected fish instances in the `<detection>` tag and the total count prediction in the `<fish_count>` tag. While FishDetector-Base exhibits minor inconsistencies, RL fine-tuning with our complementary reward design raises alignment to nearly perfect. This confirms that our *detect-to-count* design with reward-based training enforces coherent internal visual reasoning in the MLLM, yielding outputs that are not only more self-consistent but also more reliable for downstream ecological applications.

## 5. Limitations and Future Work

While FishDetector-R1 achieves notable gains, several limitations remain. First, the computational overhead of large MLLMs makes real-time deployment on resource-constrained underwater platforms challenging. Future work will explore quantization and edge-optimization to improve efficiency. Second, the framework can still hallucinate spurious detections or counts in cluttered scenes, highlighting the need for uncertainty modeling or verification mechanisms to enhance reliability. Finally, although FishDetector-R1 is class-agnostic by design, in this study we restrict supervision and evaluation to *fish-only* detection and segmentation on the DeepFish dataset, which covers a relatively narrow range of habitats and species. Extending to multi-class and species-level settings (including non-fish marine life and anthropogenic objects) is an important step toward broader ecological impact.

## 6. Conclusion

We propose **FishDetector-R1**, a unified framework for underwater fish detection, segmentation, and counting that combines an MLLM with SAM 2. Through detect-to-count prompting and reinforcement fine-tuning with sparse point labels, our method achieves strong performance across tasks on the DeepFish dataset. Notably, FishDetector-R1 bridges the gap between traditional weak and fully supervised models, delivering high-quality pixel-wise segmentation with minimal annotation cost. This enables scalable, annotation-efficient fish analysis, supporting real-world applications in ecological monitoring and marine habitat assessment.

# References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Abdullah Al Muksit, Fakhrul Hasan, Md Fahad Hasan Bhuiyan Emon, Md Rakibul Haque, Arif Reza Anwary, and Swakkhar Shatabda. Yolo-fish: A robust fish detection model to detect fish in realistic underwater environment. *Ecological Informatics*, 72:101847, 2022. 1

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 3

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *European conference on computer vision*, pages 549–565. Springer, 2016. 1, 2

[5] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 1

[6] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025. 2

[7] Rafael Garcia, Ricard Prados, Josep Quintana, Alexander Tempelaar, Nuno Gracias, Shale Rosen, Håvard Vågstøl, and Kristoffer Løvall. Automatic segmentation of fish using deep learning with application to fish size measurement. *ICES Journal of Marine Science*, 77(4):1354–1366, 2020. 2

[8] Ricardo Guerrero-Gómez-Olmedo, Beatriz Torre-Jiménez, Roberto López-Sastre, Saturnino Maldonado-Bascón, and Daniel Onoro-Rubio. Extremely overlapping vehicle counting. In *Iberian conference on pattern recognition and image analysis*, pages 423–431. Springer, 2015. 6

[9] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3, 4

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[11] Yang Hong, Xiaowei Zhou, Ruzhuang Hua, Qingxuan Lv, and Junyu Dong. Watersam: Adapting sam for underwater object segmentation. *Journal of Marine Science and Engineering*, 12(9):1616, 2024. 2

[12] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, pages 4015–4026, 2023. 2

[14] Issam Laradji, Alzayat Saleh, Pau Rodriguez, Derek Nowrouzezahrai, Mostafa Rahimi Azghadi, and David Vazquez. Affinity lcfcn: Learning to segment fish with weak supervision. *arXiv preprint arXiv:2011.03149*, 2020. 1, 2, 7

[15] Issam H Laradji, Negar Rostamzadeh, Pedro O Pinheiro, David Vazquez, and Mark Schmidt. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the european conference on computer vision (ECCV)*, pages 547–562, 2018. 1, 2

[16] Shijie Lian, Hua Li, Runmin Cong, Suqi Li, Wei Zhang, and Sam Kwong. Watermask: Instance segmentation for underwater imagery. In *ICCV*, pages 1305–1315, 2023. 1, 2

[17] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 5

[18] Yuqi Liu, Bohao Peng, Zhisheng Zhong, Zihao Yue, Fanbin Lu, Bei Yu, and Jiaya Jia. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*, 2025. 2, 4, 5

[19] Yuqi Liu, Tianyuan Qu, Zhisheng Zhong, Bohao Peng, Shu Liu, Bei Yu, and Jiaya Jia. Visionreasoner: Unified visual perception and reasoning via reinforcement learning. *arXiv preprint arXiv:2505.12081*, 2025. 3, 4

[20] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025. 2

[21] OpenAI. Introducing GPT-4.1 in the api. `https://openai.com/index/gpt-4-1/`, 2025. Accessed: 2025-09-16. 1, 2

[22] Hongwei Qin, Xiu Li, Jian Liang, Yigang Peng, and Changshui Zhang. Deepfish: Accurate underwater live fish recognition with a deep architecture. *Neurocomputing*, 187:49–58, 2016. 1, 2, 6

[23] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023. 3

[24] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 3

[25] Alzayat Saleh, Issam H Laradji, Dmitry A Konovalov, Michael Bradley, David Vazquez, and Marcus Sheaves. A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. *Scientific reports*, 10(1):14671, 2020. 1, 2, 7

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[27] Feifei Shao, Long Chen, Jian Shao, Wei Ji, Shaoning Xiao, Lu Ye, Yueting Zhuang, and Jun Xiao. Deep learning for

weakly-supervised object detection and localization: A survey. *Neurocomputing*, 496:192–207, 2022. 1, 2

[28] Austin Stankus. State of world aquaculture 2020 and regional reviews: Fao webinar series. *FAO aquaculture newsletter*, (63):17–18, 2021. 1

[29] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 2

[30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[31] Muduo Xu, Jianhao Su, and Yutao Liu. Aquasam: Underwater image foreground segmentation. In *International Forum on Digital TV and Wireless Multimedia Communications*, pages 3–14, 2023. 2

[32] Zuyao You and Zuxuan Wu. Seg-r1: Segmentation can be surprisingly simple with reinforcement learning. *arXiv preprint arXiv:2506.22624*, 2025. 2, 3, 4

[33] En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun, Chunrui Han, Zheng Ge, et al. Perception-r1: Pioneering perception policy with reinforcement learning. *arXiv preprint arXiv:2504.07954*, 2025. 2, 3, 4

[34] Wenbo Zhang, Chaoyi Wu, and Zhenshan Bao. Dpanet: dual pooling-aggregated attention network for fish segmentation. *IET computer vision*, 16(1):67–82, 2022. 1, 2

[35] Zhihao Zhang, Qiaole Dong, Qi Zhang, Jun Zhao, Enyu Zhou, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Yanwei Fu, et al. Reinforcement fine-tuning enables mllms learning novel tasks stably. *arXiv preprint arXiv:2506.23508*, 2025. 3