MINIMAX

One RL to See Them All
Visual Triple Unified Reinforcement Learning

Full author list in Contributions[1]

Reinforcement learning (RL) has significantly advanced the reasoning capabilities of vision-language models (VLMs). However, the use of RL beyond reasoning tasks remains largely unexplored, especially for perception-intensive tasks like object detection and grounding. We propose **V-Triune**, a **V**isual **Tri**ple **U**nified Rei**n**forcement L**e**arning system that enables VLMs to jointly learn visual reasoning and perception tasks within a single training pipeline. V-Triune comprises triple complementary components: *Sample-Level Data Formatting* (to unify diverse task inputs), *Verifier-Level Reward Computation* (to deliver custom rewards via specialized verifiers) , and *Source-Level Metric Monitoring* (to diagnose problems at the data-source level). We further introduce a novel Dynamic IoU reward, which provides adaptive, progressive, and definite feedback for perception tasks handled by V-Triune. Our approach is instantiated within off-the-shelf RL training framework using open-source 7B and 32B backbone models. The resulting model, dubbed **Orsta** (**O**ne **R**L to **S**ee **T**hem **A**ll), demonstrates consistent improvements across both reasoning and perception tasks. This broad capability is significantly shaped by its training on a diverse dataset, constructed around four representative visual reasoning tasks (Math, Puzzle, Chart, and Science) and four visual perception tasks (Grounding, Detection, Counting, and OCR). Subsequently, Orsta achieves substantial gains on MEGA-Bench Core, with improvements ranging from +2.1 to an impressive +14.1 across its various 7B and 32B model variants, with performance benefits extending to a wide range of downstream tasks. These results highlight the effectiveness and scalability of our unified RL approach for VLMs. The V-Triune system, along with the Orsta models, is publicly available at https://github.com/MiniMax-AI.
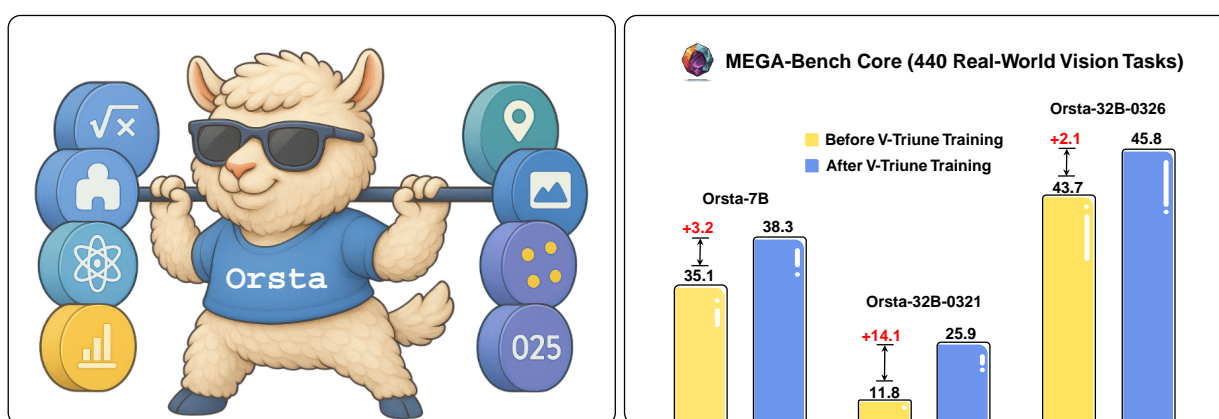

Figure 1 | **Performance of Orsta on MEGA-Bench Tasks**. V-Triune is evaluated across visual reasoning and visual perception tasks—Math, Science, Charting, Puzzle, Detection, Grounding, Counting, and OCR, demonstrating notable performance gains of Orsta over the backbone: +3.2%, +14.1%, and +2.1% in different model variants.

---

1. Introduction

The recent advancement of large language models (LLMs) (Guo et al., 2025; Jaech et al., 2024) and visual-language models (VLMs) (Google DeepMind, 2025; OpenAI, 2025) has seen a paradigm shift from pre-training scaling to test-time scaling. A key manifestation of this shift is the practice of extending context length for Chain-of-Thought (CoT) reasoning, which significantly enhances performance on complex tasks such as mathematics and coding (Guo et al., 2025; Xia et al., 2025).

While reinforcement learning (RL) has emerged as a promising method for post-training VLMs, current research remains limited (Li et al., 2025a; Liu et al., 2025d,e; Ma et al., 2025a; Shen et al., 2025; Tan et al., 2025; Wang et al., 2025b; Yang et al., 2025; Yu et al., 2025a). Most prior work has focused on narrow task domains—typically visual reasoning tasks like math QA and Science QA (Huang et al., 2025; Meng et al., 2025; Yang et al., 2025), where the RL setup closely mirrors RL training paradigms in LLMs. Moreover, existing works (Liu et al., 2025a,c) remain an open question whether RL can be effectively scaled to visual perception tasks such as object detection and grounding, which require distinct reward design and measures to ensure training stability.

We introduce V-Triune (**V**isual **Tri**ple **U**nified Rei**n**forcement **L**earning), the first unified RL system for post-training VLMs on both visual reasoning and perception tasks. V-Triune integrates three complementary components, each operating at a distinct level for this unification: *Sample-Level Data Formatting* (detailed in Sec. 3.1) handles diverse task and reward needs by allowing each sample to define its reward setup and chosen verifier. *Verifier-Level Reward Computation* (Sec. 3.2) offers key modularity and task-adaptability by assigning reward generation to specialized verifiers for specific task groups. Lastly, *Source-Level Metric Monitoring* (Sec. 3.3) provides essential tracking and diagnostics by logging metrics per data source, vital for spotting data issues and ensuring stable multi-task, multi-source learning. Beyond these core components, a key innovation in V-Triune is the Dynamic IoU reward (Sec. 3.4). This mechanism targets visual perception tasks like object detection and grounding, addressing issues with fixed IoU thresholds. By progressively adjusting the IoU reward threshold (from relaxed to stricter criteria), it ensures useful early learning signals, guides the model towards high-precision results, and ultimately enables stable, scalable training procedure.

Leveraging the V-Triune system, we develop the Orsta model series, featuring variants with sizes ranging from 7B to 32B, built upon the Qwen2.5-VL family of baselines. These models undergo joint optimization across a diverse set of tasks, spanning visual reasoning (mathematics, science, chart, puzzle) and visual perception (object detection, grounding, OCR, counting). On the comprehensive MEGA-Bench core benchmark (Chen et al., 2024), which covers over 400 real-world visual tasks, Orsta demonstrates substantial performance gains. These improvements range from +2.1% up to an impressive +14.1% across its various 7B and 32B model variants. These performance benefits extend to prominent downstream benchmarks (including MMMU, MathVista, COCO, and CountBench), validating V-Triune's effectiveness and scalability. Our core contributions are:

- We introduce V-Triune, the first unified, scalable, and extensible RL system designed for jointly training VLMs on both visual reasoning and perception tasks within a single paradigm.
- We propose the Dynamic IoU Reward, a novel, adaptive reward mechanism that significantly enhances stability and performance for visual perception tasks like detection and grounding.
- We establish and demonstrate a comprehensive training methodology, including key engineering optimizations, enabling effective and stable RL training across eight diverse VLM tasks spanning both reasoning and perception.
- We present Orsta, a family of high-performance models (7B-32B) trained with V-Triune, achieving substantial gains (up to +14.1%) on the MEGA-Bench Core and strong performance across various downstream benchmarks.

2. Preliminary

The advent of Deepseek-R1 (Guo et al., 2025) has established Reinforcement Learning as one of the dominant paradigms for post-training LLMs. Specifically, training through RL canlead to remarkably strong performance on reasoning tasks such as mathematics, code, and puzzles (OpenAI, 2025; Xia et al., 2025). RL encompasses two main key components: the algorithm and (verifiable) reward function.

2.1. RL Algorithm

We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), following standard practice, but introduce two key modifications based on recent work (Hu et al., 2025; Yu et al., 2025b). First, we remove the reference model and its associated KL loss. This avoids restricting exploration (Hu et al., 2025) and prevents unstable KL divergence estimates (Schulman, 2020), while also reducing GPU memory usage and speeding up training. Second, we apply the clip-high trick and use token-level loss, which help increase output entropy, encouraging better exploration and improving training stability. With these changes, the GRPO objective is updated as follows:

$$
\mathcal{J}(\theta) = \mathbb{E}_{q \sim \mathcal{P}(q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)}
$$
$$
\left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min\left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}}) \hat{A}_{i,t} \right) \right] \tag{1}
$$

where

$$
r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)}. \tag{2}
$$

In this equation, $\mathcal{P}(q)$ represents the distribution of queries. The behavior model $\pi_{\theta_{\text{old}}}$ samples a group of G responses $\{o_i\}_{i=1}^G$ for each query. Then, the advantage of the i-th response $\hat{A}_{i,t}$ is calculated from the normalized group-level rewards $\{R_i\}_{i=1}^G$, and then used to update the policy π_θ. The clipping range of the importance sampling ratio r is bounded by ϵ_{low} and ϵ_{high}.

2.2. Reward function

As discussed in Sec. 1, visual tasks involved can be grouped into two categories: visual reasoning and visual perception. To mitigate the risk of reward hacking (Weng, 2024), rule-based reward functions are employed for each task. For visual reasoning tasks, a binary (0-1) reward function is employed based on accuracy, as defined by the following rule:

$$
R_{\text{acc}}(\hat{a}, a) = \begin{cases} 1, & \text{if } \texttt{verify}(\texttt{parse}(\hat{a}), \texttt{parse}(a)) \text{ is True} \\ 0, & \text{else} \end{cases} \tag{3}
$$

where $\hat{a}$ denotes the predicted answer and a represents the ground-truth (or golden) answer. $\hat{a}$ is parsed from model output, which is instructed to be enclosed within \boxed{}, and verified against the golden answer a using `math_verify` (Kydlíček, 2025).

For visual perception tasks, the format of the ground-truth varies considerably. The answers for counting and verifiable OCR tasks are typically numbers, words, or open-ended phrases, which can be conveniently enclosed in \boxed{}. Therefore, a rule-based accuracy reward, as defined in Eq. 3, is utilized for these tasks. In contrast, the answers for grounding and detection tasks are commonly structured in COCO-style JSON format, which includes both a bounding box and a label.

Initial experiments revealed that the model struggled to learn enclosing such JSON format within the \boxed{}, but readily learned to include it within <answer></answer> tags. Consequently, for these two tasks, a format reward is established by referencing (HuggingFace, 2025) setup:

$$R_{\text{format}}(o_q) = 0.25 \sum_{i=1}^{4} \mathbb{I}(\text{count}(o_q, s_i) = 1) \tag{4}$$

$$s_1 = \texttt{<think>}, s_2 = \texttt{</think>}, s_3 = \texttt{<answer>}, s_4 = \texttt{</answer>}$$

where o_q represents the model's response to question q, and s_i denotes a specific format tag. The indicator function $\mathbb{I}(\text{condition})$ evaluates to 1 if the condition is true, and 0 otherwise. For the accuracy reward, Intersection over Union (IoU) (Everingham et al., 2010) with a defined threshold and mean Average Precision (mAP) (Lin et al., 2014) are commonly adopted in prior works (Liu et al., 2025e; Shen et al., 2025; Yu et al., 2025a). The corresponding formulations for the IoU-based reward and mAP are provided in Eq. 5 and Eq. 7, respectively.

$$R_{\text{acc}}(\hat{a}, a) = \begin{cases} \text{IoU}(\hat{a}, a), & \text{if} \quad \text{IoU}(\hat{a}, a) \geq \epsilon \\ 0, & \text{else} \end{cases} \tag{5}$$

where

$$\text{IoU}(\hat{a}, a) = \frac{\text{Area}(\hat{a} \cap a)}{\text{Area}(\hat{a} \cup a)} \tag{6}$$

$\hat{a}$ represents the predicted bounding box, and a denotes the golden bounding box. The threshold ϵ controls the strictness of the reward function, with higher values enforcing tighter matches. We adopt a dynamic ϵ finally, as detailed in Sec. 3.4. The mAP is calculated using interpolated precision curve:

$$\text{AP}_c = \int_0^1 \underbrace{\left(\max_{\tilde{r} \geq r} P_c(\tilde{r}) \right)}_{\text{interpolated precision } P_c^{\text{interp}}(r)} \mathrm{d}r, \qquad \text{mAP} = \frac{1}{C} \sum_{c=1}^{C} \text{AP}_c. \tag{7}$$

Finally, the overall reward combines accuracy and format alignment, expressed as:

$$\alpha_{\text{acc}} \cdot R_{\text{acc}} + \alpha_{\text{format}} \cdot R_{\text{format}} \tag{8}$$

where α_{acc} and α_{format} are the respective weighting coefficients for each reward component.

3. V-Triune: Visual Triple Unified Reinforcement Learning

This section describes V-Triune, our visual triple unified RL system. As shown in Fig. 2, V-Triune's main goal is to jointly train VLMs on both visual reasoning and perception tasks using a single, unified training pipeline. The system is built upon three core, interconnected parts designed to handle these diverse tasks together. The following subsections will explain these three core components in detail and introduce our novel Dynamic IoU reward mechanism.

3.1. Sample-Level Data Formatting

This section introduces how data is formatted to support unified training across perception and reasoning tasks. A main challenge is that different tasks may require different types of rewards, components, and weighting strategies. For example, tasks like math, puzzle, and OCR compute rewards based on the correctness of textual answers, while detection and grounding tasks rely
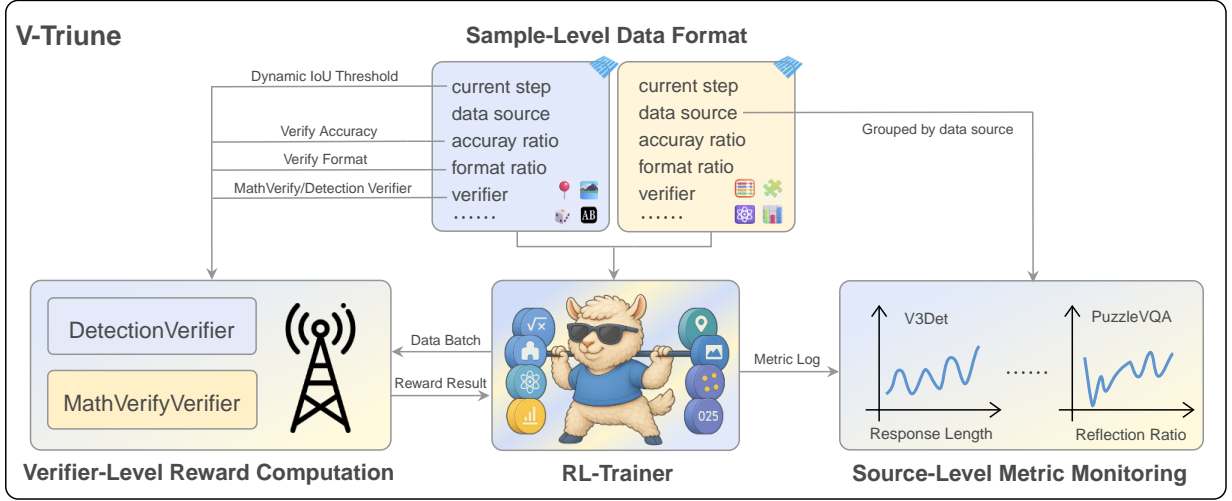
Figure 2 | **V-Triune System**. It integrates three complementary components: *Sample-Level Data Formatting* (to unify diverse task inputs), *Verifier-Level Reward Computation* (for custom rewards via specialized verifiers), and *Source-Level Metric Monitoring* (to diagnose data-source level problems). Additionally, a novel Dynamic IoU reward offers adaptive, progressive feedback for perception tasks.

on spatial metrics such as IoU and bounding box formatting. In conventional RL setups, reward computation is typically defined at the task level. While this allows modular reward functions to be implemented externally, it limits flexibility when fine-grained control is required. Many multimodal tasks may contain heterogeneous samples that demand different reward strategies. For instance, OCR data may include both plain-text lines and complex tables, each requiring different evaluation rules. Similarly, detection samples can differ significantly in terms of object count, annotation completeness, or visual difficulty, which suggests the need for sample-wise adjustment of reward behavior.

To support such flexibility, we define reward configurations directly at the sample level. Each sample specifies the types of rewards to compute, their relative weights, and the associated verifier to use. This allows dynamic reward routing and fine-grained weighting during training without modifying core training logic. It can also support curriculum learning or data ablation strategies simply by adjusting metadata, making the system more extensible and maintainable.

As shown in Fig. 3, we implement our data schema using the HuggingFace `datasets` (Lhoest et al., 2021), which serves as the unified interface for all sources described in Sec. 5.1.1. Besides common fields such as `images` and `prompt`, the format includes several task-agnostic keys to support reward control. The `reward_model` field encapsulates all reward-related metadata, including fields like `accuracy_ratio` and `format_ratio`, which define the weight of each reward type. By adjusting these values, specific rewards can be enabled, disabled, or reweighted per sample. The `verifier` field specifies which verifier should be used for evaluation and provides any required parameters. More details on the verifier system are discussed in Sec. 3.2. The `data_source` field indicates the origin of each sample and is used in source-specific evaluation in Sec. 3.3. In summary, the sample-level formatting design enables seamless integration of diverse datasets into a unified training pipeline, while allowing highly flexible and scalable reward control.

3.2. Verifier-Level Reward Computation

Unlike approaches using fixed reward functions, we implement a standalone, asynchronous reward server to generate RL signals. Decoupling reward computation from the main training loop offers key

Data Format

```
{
    "data_source": Value(dtype="string"),
    "images": Sequence(feature=Image(mode=None, decode=True)),
    "prompt": [
        {
            "content": Value(dtype="string"),
            "role": Value(dtype="string")
        }
    ],
    "ability": Value(dtype="string"),
    "reward_model": {
        "answer": Value(dtype="string"),
        "ground_truth": Value(dtype="string"),
        "accuracy_ratio": Value(dtype="float32"),
        "format_ratio": Value(dtype="float32"),
        "verifier": Value(dtype="string"),
        "verifier_parm": Value(dtype="dict")
    },
    "extra_info": {
        "id": Value(dtype="string"),
        "image_path": Value(dtype="string")
    }
}
```

Figure 3 | **Sample-level Data Scheme for Unified Training**. This format, implemented using HuggingFace datasets, allows fine-grained control over reward computation by defining `reward_model` (including reward types, weights like `accuracy/format_ratio`) and `verifier` specifications at the individual sample level. This enables flexible and scalable handling of diverse multimodal tasks.

advantages: it enhances modularity and extensibility for diverse tasks, supports flexible deployment and independent scaling (crucial for GPU-based evaluations), improves maintainability, and boosts throughput via distributed, asynchronous processing. The term *verifier-level* defines our reward computation granularity: instead of monolithic functions, rewards are delegated to specialized verifiers, each handling specific task groups.

We use an asynchronous client-server architecture (Fig. 4) using FastAPI (Ramírez, 2021). The client collects batches of samples from the dataloader and processes them in thread level parallel. Each sample is transformed into a structured payload containing task-specific key-value pairs. The client consists of multiple proxy workers that asynchronously send requests to the server with a dynamic balance strategy. The proxy server routes each request to the appropriate verifier based on the "`verifier`" key in data format.

Each verifier is user-defined, containing custom functions to compute task-specific rewards based on model outputs and ground truth. In our implementation, we primarily use two types: *MathVerifyVerifier*: Handles reasoning, OCR, and counting tasks by evaluating answer correctness. *DetectionVerifier*: Manages detection and grounding tasks, notably employing our novel Dynamic IoU reward (Sec. 3.4) alongside format-based rewards. This verifier-level approach grants significant flexibility and modularity. It greatly simplifies adding new tasks or updating reward logic independently, without altering the core training pipeline.
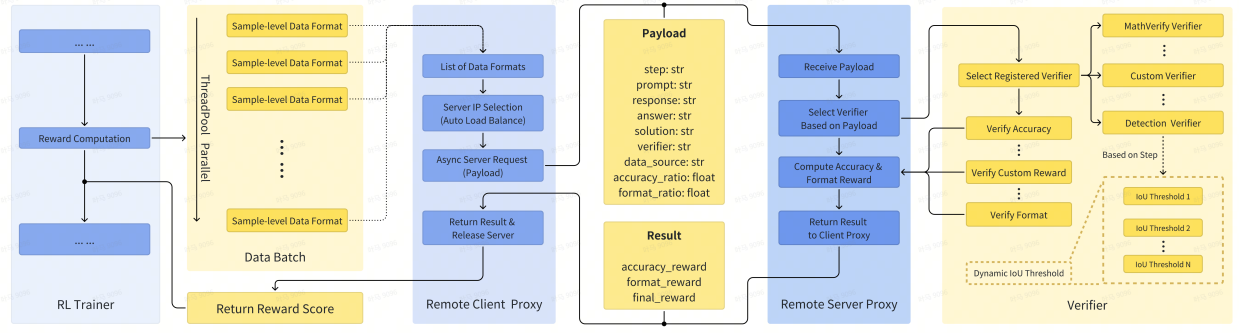
Figure 4 | **Architecture of the Asynchronous Reward Server.** The RL trainer interacts with a remote server via client-server proxies, where specialized verifiers (e.g., MathVerify, Detection) compute rewards using task-specific logic and dynamic thresholds (e.g., dynamic IoU threshold).

3.3. Source-Level Metric Monitoring

Monitoring training metrics is essential for understanding model dynamics and real-time issue diagnosis. However, for multi-task, multi-source training, aggregated or single-task metrics are often insufficient due to lack of traceability and per-source data variations. Therefore, we adopt source-level metric monitoring. This involves logging metrics per `data_source` for each batch. This approach helps identify faulty data sources, enables targeted debugging, and reveals cross-source learning interactions. Such granular monitoring is particularly vital in RL, where training can be unstable, to verify stability and model behavior. We note that while many RL infrastructures support metric logging (Hu et al., 2024; Sheng et al., 2024), our source-level detail provides deeper insights. Empirical signs of training stability are further discussed in Sec. 4. Specifically, we log per-source reward values to trace dataset-specific stability. For perception tasks like grounding and detection, we also log detailed per-source IoU values (at thresholds 50, 75, 95, 99) and mAP scores for fine-grained convergence insights.

Inspired by Ma et al. (2025b), we also track model behavior by monitoring response length and reflection ratio per source. For response length, we log various metrics including overall average, lengths of correct/incorrect responses, and truncation rates. Truncation rates (outputs hitting maximum length) can indicate overly verbose or collapsed generation. For reflection ratio, we track 15 predefined reflection-related words (e.g., "*re-check*", "*re-think*", "*verify*"). We then compute `reflection ratio`: the proportion of responses containing these reflection words and `correct ratio in reflection responses`: the accuracy of responses that include these words. This helps diagnose model tendencies like overthinking versus superficial responses by linking reflection to correctness. All mentioned metrics are consistently logged per data source.

3.4. Dynamic IoU Reward

In object detection and visual grounding, Intersection over Union (IoU) and mean Average Precision (mAP) are standard metrics for evaluating the overlap between predicted and ground-truth boxes. While mAP summarizes performance across multiple IoU thresholds, our preliminary detection experiments show that using a thresholded IoU reward achieves comparable overall performance while providing more interpretable and controllable feedback (Fig. 5a). Therefore, we adopt an IoU-based reward strategy for its robustness and stronger alignment with localization accuracy (Eq. 5).

However, relaxed thresholds like $\epsilon = 0.5$ (common option (Liu et al., 2025e; Yu et al., 2025a)) can be too lenient for RL in VLMs, especially when compared to the strict exact-match rewards used
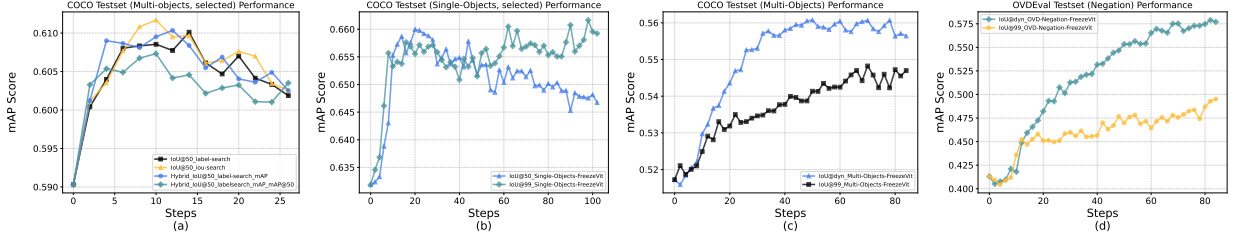
Figure 5 | **COCO Test Set Performance with Various Reward Designs.** (a) Comparison between IoU-based and mAP-based rewards on a selected COCO multi-object subset; (b) Comparison between vanilla IoU reward and rule-based IoU reward on a selected COCO single-object subset; (c, d) Comparison between rule-based IoU reward and dynamic IoU reward on the COCO multi-object subset and the OVDEval negation subset.

in reasoning tasks (Eq. 3). Moreover, these loose thresholds result in reward ambiguity: even if predicted bounding boxes merely shift or are oriented differently around the ground-truth, they can still receive the same high reward. Such ambiguity can ultimately lead to a decline in performance during later training stages for detection task (Fig. 5b).

To address these issues, we initially use a rule-based IoU reward scheme that enforces a stringent threshold of $\epsilon = 0.99$, requiring near-exact spatial alignment between predicted and ground-truth boxes (Fig. 5b). This strict criterion enhances consistency between perception and reasoning signals and, by providing an unambiguous target, aims to improve training stability. However, its very stringency introduces a cold-start problem in early rollouts, where most predictions receive 0 reward.

To mitigate this cold-start problem, we adopt a dynamic IoU reward strategy, inspired by curriculum learning (Bengio et al., 2009). The IoU threshold ϵ is adjusted in stages based on training progress: starting at 0.85 for the initial 10% of training steps, increasing to 0.95 for the subsequent 15% of steps (i.e., from 10% to 25% of total steps), and finally settling at 0.99 for the remainder of training (as illustrated in Fig. 6).

Figure 6 | **Training accuracy rewards under Dynamic IoU versus a fixed Rule-based IoU (IoU@99).**

4. Training Recipe

V-Triune enables a scalable system of data, tasks, verifiers, and metrics as the foundation of training. However, early experiments revealed that joint training can lead to instability: (1) degraded evaluation performance, (2) sudden spikes in gradient norms, (3) large entropy fluctuations, and (4) abrupt increases in response length, particularly in incorrect outputs. We address training instability and scalability through targeted adjustments, including freezing ViT to prevent gradient explosion, filtering spurious image tokens, randomizing CoT prompts, and decoupling evaluation to manage memory during large-scale training.

4.1. Disable ViT Training

In initial experiments, we performed full-parameter training by jointly optimizing the ViT and LLM. However, detection performance consistently collapsed after several dozen steps, regardless of
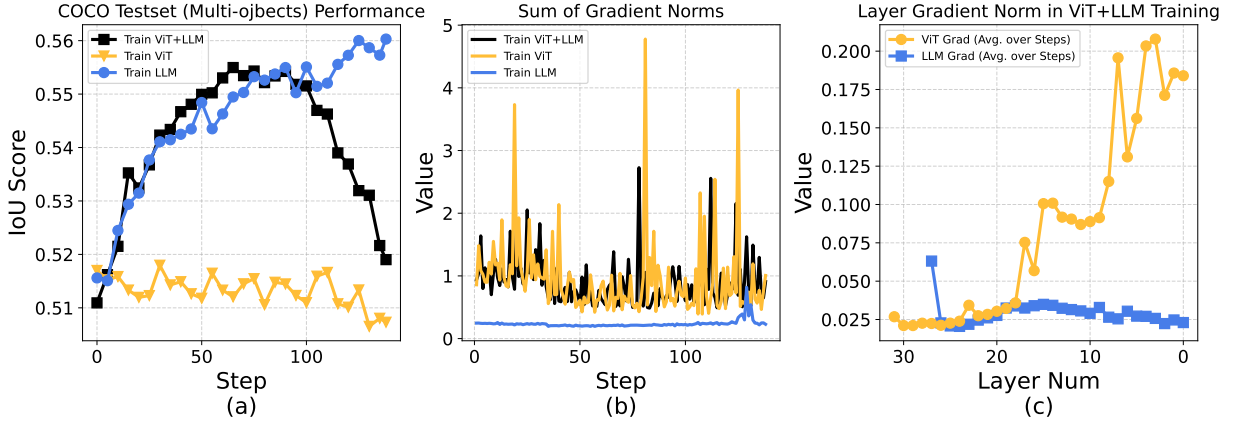
Figure 7 | **Analysis of ViT Training Instability.** (a) COCO testset (OOD) performance comparison. (b) Sum of gradient norms under different training schemes. (c) Layer-wise gradient norms of ViT and LLM during full parameter training. Notably, incorporating ViT training leads to a performance decline and highly unstable gradients; remarkably, ViT's gradients amplify during back-propagation, contrasting with the stable layer-wise gradients of the LLM.

hyperparameter settings. Log analysis revealed unusually large and spiking gradient norms (often >1), suggesting instability originating from the ViT. To verify this, we conducted three training configurations: (1) LLM-only, (2) ViT-only, and (3) full-parameter training, all using identical RL settings on Orsta-7B with mixed task data. We monitored: (a) COCO test set performance, (b) total gradient norm, and (c) layer-wise gradient trends during full-parameter training.

As shown in Fig. 7a, joint training leads to a performance drop, whereas LLM-only training maintains stable gains. ViT-only training yields minimal improvement, indicating that RL benefits primarily stem from updating the LLM. Figure Fig. 7b shows that ViT training produces significantly higher gradient norms—over 10× larger than LLM-only training.

Layer-wise analysis (Fig. 7c) confirms this: LLM gradients remain stable across layers, while ViT gradients amplify during backpropagation, with the first layer exhibiting 5–10× larger norms than the last. This gradient explosion destabilizes training and undermines visual performance. Consequently, we freeze ViT parameters in subsequent experiments.

The root cause of this instability remains an open research question, but we offer two key insights. First, RL not only activates VLM capabilities but also enforces modality alignment by grounding responses in visual content. When ViT and LLM are trained jointly, the visual representation—i.e., the alignment target—shifts constantly, leading to instability analogous to the concept drift problem in machine learning (Gama et al., 2014). This dynamic target undermines stable optimization and may cause model collapse. Alternating training, similar to GANs (Goodfellow et al., 2020), where one component is frozen while the other is updated, could offer a solution. Second, ViT's contrastive pretraining may limit its suitability for RL, as it encourages static, instance-level features rather than the dynamic and causal representations needed for RL tasks. To mitigate this mismatch, auxiliary self-supervised objectives could be introduced during RL to help ViT adapt to the evolving task demands.

4.2. Mitigating Spurious Image Special Tokens

To enable accurate advantage estimation, logits for both the query and the generated response are recomputed, as those returned by the inference engine may be imprecise. During the forward pass, image placeholders (highlighted in the red box in Fig. 8, appearing before the "vision_end" token) are replaced with visual features extracted by the ViT and adapter modules. However, the model may mistakenly generate special tokens (highlighted in the blue box in Figure 8), such as image or video

Figure 8 | **An Example of Spurious Image Tokens.**

placeholders, that lack corresponding features—particularly under RL-zero settings. To ensure input–feature alignment and maintain training stability, a filtering step is applied to remove all such special tokens from the rollout sequence prior to recomputation.

4.3. CoT Prompt Pool

In the early stages of training for visual mathematics tasks, variations in CoT prompts, despite conveying identical meanings, which can influence model performance, affecting metrics such as accuracy and response length. To reduce this variability, we construct a CoT prompt pool comprising 10 alternatives for "Let's think step by step" and 10 for "Place the answer in \boxed{}." During training, one sentence from each group is randomly selected and appended to the instruction. This strategy mitigates prompt-induced variance and is applied specifically to samples verified with `MathVerifyVerifier`.

4.4. System Memory Management

V-Trinue is implemented atop Verl, a single-controller training framework that can approach system memory limits on the master node, particularly with large-scale vision datasets. To enable effective OOD performance monitoring, we introduce online test-set benchmarking at regular intervals. To mitigate the resulting system overhead, we decouple the testing phase from the main training loop and batch-process benchmarks, bypassing the default vLLM data handling.

5. Experiment

5.1. Implementation Details

Model: We adopt Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-32B-Instruct (0321 and 0326 versions, denoting their release dates) as our base models, which are trained on 4 trillion tokens during the multimodal pre-training stage, followed by supervised fine-tuning (SFT) on 2 million instances. They are chosen for their strong performance in vision-language reasoning and perception tasks.

Framework and Hardware: V-Triune is implemented on verl (Sheng et al., 2024). We enable native FSDP for training and use vLLM for generation. All experiments are conducted on 64 NVIDIA H20 GPUs to meet the memory-intensive requirements of RL training.
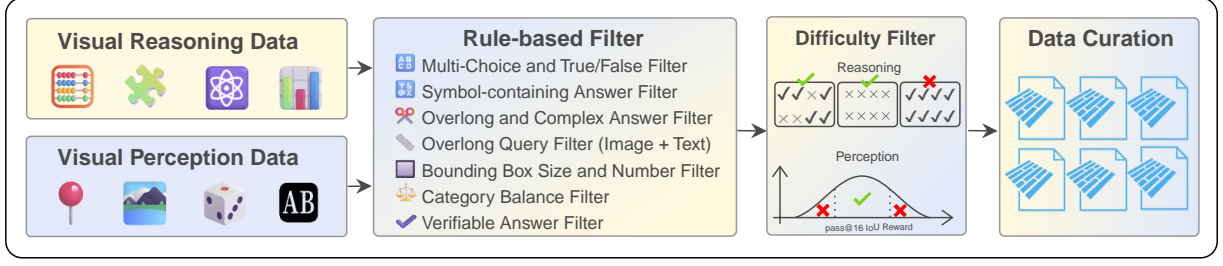
Figure 9 | **Data Curation Process**. First, visual reasoning and visual perception data pass through a rule-based filter, which removes samples that do not meet preset criteria. Subsequently, the data enters a difficulty filter, which removes samples that are too easy or too hard based on model performance, ultimately producing the Curated Dataset.

5.1.1. Data Curation

We select four reasoning tasks—Math, Puzzle, Science, and Chart—for their varied reasoning demands, and four perception tasks—Detection, Grounding, Counting, and OCR—for their broad coverage of visual understanding. Data sources for each task are listed below:

- For the Math task, `mm_math` (Sun et al., 2024), `geometry3k` (Lu et al., 2021), and `mmk12` (Meng et al., 2025) are chosen.
- For the Puzzle task, `PuzzleVQA`(Chia et al., 2024) and `AlgoPuzzleVQA`(Ghosal et al., 2024) are merged due to their shared origin, and `VisualPuzzles` (Song et al., 2025) is additionally included.
- For the Science task, `ScienceQA` (Lu et al., 2022), `SciVQA` (Borisova and Rehm, 2025), and the "Broader STEM Topics" and "(GradeSchool) Science" categories from `ViRL39K` (Wang et al., 2025b) are used.
- For the Chart task, `ChartQAPro` (Masry et al., 2025), `ChartX` (Xia et al., 2024), `Table-VQA` (Kim et al., 2024), and the Tables/Diagrams/Charts categories from `ViRL39K` (Wang et al., 2025b) are used.
- For the Detection task, `V3Det` (Xie et al., 2023) and `Object365` (Shao et al., 2019) are chosen.
- For the Grounding task, D^3 (Xie et al., 2023) is used.
- For the Counting task, `CLEVR` (Johnson et al., 2017; Tan et al., 2025) is used.
- For the OCR task, English OCR questions are extracted from `LLaVA-OV Data` (Li et al., 2024) and `EST-VQA` (Wang et al., 2020).

To reduce noise, we apply a two-stage data filtering process (Figure Fig. 9): (1) rule-based filtering and (2) difficulty-based filtering. This yields 47.7K high-quality samples across 18 datasets and 8 tasks. To mitigate dataset bias, puzzle data is duplicated to ensure sufficient coverage. The final corpus includes approximately **20.6K perception** and **27.1K reasoning** samples, primarily consisting of single-image, single-turn conversations.

5.1.1.1 First Stage: Rule-based Filter

For the four visual reasoning tasks, the following filters are applied:

- Multiple-choice and true/false questions that are prone to hacking are discarded. (Team et al., 2025b)

- Answers containing symbols such as "=", "[", "]", "(", ")", and ";" are removed, as the absence of these symbols may cause answer mismatches even if the numeric values are correct.
- Answers longer than 20 characters are discarded to avoid overly complex answers.

The filtering process for visual perception tasks involves additional complexity:

- **Detection:** Following `Qwen2.5-VL` (Bai et al., 2025), data is converted to relative coordinates. Single-box samples contain one box per category, while multi-box samples retain original annotations. Samples with over 10 boxes per category or boxes exceeding 50% of the image are removed. A 1:2 single-to-multi-box ratio is enforced, and category-level long tails are avoided.
- **Grounding:** Data is processed into relative coordinates, and data with a box size greater than 50% of the image is discarded. Complex phrase labels are filtered out.
- **Counting:** Data is balanced per category and only English data is retained.
- **OCR:** Only English OCR data is retained, and final labels must be verifiable by `math_verify` (Kydlíček, 2025). Since no verifiable reward model (RM) is designed, the OCR task data must pass this validation.

5.1.1.2 Second Stage: Difficulty-based Filter

To remove low-value samples, easy questions already solvable by the base model are filtered out.

For reasoning tasks, we use `Qwen2.5-VL-32B-0321` to compute pass@8, retaining only samples with $0 \leq$ pass@8 $< 100\%$. For perception tasks, specifically detection and grounding, pass@16 is computed using `Qwen2.5-VL-7B` with a 0.5 IoU threshold, and samples with cumulative IoU rewards between 2 and 10 are selected.

All curated data is stored in Parquet format (Apache Software Foundation, 2025) and uniformly mixed for training without online filtering or curriculum scheduling.

5.1.2. Training Details

Two distinct RL training paradigms are explored: on-policy and off-policy. For both settings, the rollout batch size is fixed at 1024 across all experiments. The backward batch size is set to 1024 for on-policy and 128 for off-policy. All experiments use the GRPO algorithm, generating 8 candidate sequences per prompt.

To promote exploration while maintaining stability, we apply the clip-high strategy from DAPO (Yu et al., 2025b) with clipping thresholds $\varepsilon_{\text{high}} = 0.28$ and $\varepsilon_{\text{low}} = 0.2$. This encourages low-probability token sampling without destabilizing training. No reference model is used; the final loss is computed solely via token-level mean PPO-clip loss.

To further stabilize learning, the ViT and connector modules remain frozen throughout. Learning rates are set to 1×10^{-6} (on-policy) and 5×10^{-7} (off-policy), with a 5% warmup before being held constant. During rollout, we use the vLLM engine with a temperature of 1.0, top-p of 1.0, and a maximum sequence length of 2048. Greedy decoding is applied during testing.

5.1.3. Evaluation Benchmarks

To comprehensively assess the model's capabilities, we evaluate performance across three domains: real-world tasks, visual reasoning, and visual perception.

Query Example of Detection and Grounding

```
Please detect all instances of the following category within the image:
{LABEL}.

Let's think step by step and output the final answer in <answer> and </
answer> tags.
For example:
Your detailed reasoning process here.
<answer>
[{'bbox_2d': [x1,y1,x2,y2],'label': label_name}]
</answer>
```
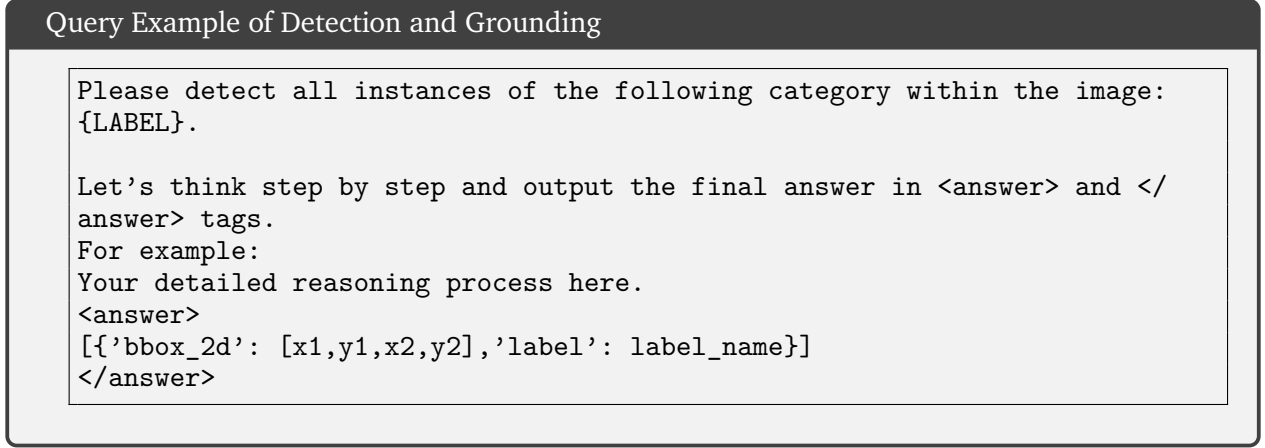

Figure 10 | **Example query format for detection and grounding tasks.** The query instructs VLMs to identify instances of a given object and format the output in a specific reasoning-answer format.

For real-world task evaluation, we employ the core subset of **MEGA-Bench** (Chen et al., 2024), which consists of 440 diverse, long-tailed tasks encompassing over 6,000 expert-curated samples. All results are reported using the official MEGA-Bench evaluation implementation to ensure consistency.

To evaluate reasoning and knowledge capabilities, we adopt **MMMU**(Yue et al., 2024) and **MathVista**(Lu et al., 2023). Both benchmarks are assessed using VLMEvalKit with default prompts (i.e., without Chain-of-Thought) and inference settings. Since GPT-4o is used for both answer extraction and scoring—which may introduce variability—we report both rule-based and GPT-4o-based scores for MMMU.

For visual perception evaluation, we include **COCO** (val-2017)(Lin et al., 2014), **OVDEval**(Yao et al., 2023), **CountBench**(Paiss et al., 2023), **OCRBench** (v2)(Fu et al., 2024), and **ScreenSpot-Pro** (Li et al., 2025b). We report both IoU and mAP for COCO and OVDEval. Notably, mAP and NMS-mAP are computed as the mean of sample-level metrics, rather than over the entire dataset, due to the absence of confidence scores in model outputs. These benchmarks are evaluated using simple task-specific queries, with examples provided below. OCRBench is evaluated using LMMs-Eval(Zhang et al., 2024) on the entire dataset, encompassing both English and Chinese samples. ScreenSpot-Pro is assessed with its official evaluation code, using instruction-style English prompts, positive ground truth, and all task types. All bounding boxes and keypoints are represented using coordinate values relative to the original input image dimensions.

5.2. Performance

5.2.1. MEGA-Bench

A comprehensive comparison of Orsta against its backbone and leading general-purpose/reasoning-enhanced VLMs is available in Tab. 1. Orsta shows consistent gains at both 7B and 32B scales: Orsta-7B achieves 38.31 (+3.2) on MEGA-Bench Core, and Orsta-32B reaches 45.78 (+2.1). Our method V-Triune notably boosts performance in domains with enriched training data—mathematics (+3.8 at 7B, +5.4 at 32B), perception, planning, and science—indicating strong generalization in both reasoning and perception tasks. In contrast, coding and metric-related tasks show limited improvement due to sparse supervision, underscoring the targeted scalability of our unified RL training approach.

Table 1 | **Performance of Orsta on MEGA-Bench core**. Comparison of general-purpose and reasoning-enhanced VLMs (7B+ & 32B+). Models with improved reasoning are marked by 💡; final scores are reported as weighted averages. QwenVL-2.5-32B-0321 has known issues, which are resolved in the 0326 version. All results are obtained using the official MEGA-Bench evaluation code, except for Gemma3-27B (†).

Model	Knowledge	Mathematics	Perception	Coding	Info. Ex.	Planning	Science	Metrics	MEGA-Bench Core
7B+ Model									
QwenVL-2-7B	39.96	25.95	39.99	31.49	40.29	16.64	28.59	43.61	34.47
QwenVL-2.5-7B	38.84	27.67	41.24	28.93	50.23	16.32	36.75	41.64	35.07
InternVL-3-8B	36.64	**32.75**	42.17	**35.11**	48.92	14.35	36.51	**53.94**	36.48
Gemma3-12B	41.11	29.10	37.38	30.27	46.56	16.10	36.83	50.40	35.04
Kimi-VL-A3B	37.63	27.07	39.50	22.30	40.99	**22.17**	33.94	46.65	34.40
MM-Eureka-7B 💡	40.12	31.59	39.71	28.75	49.32	16.64	37.25	46.39	35.96
VL-Rethinker-7B 💡	40.65	30.08	42.02	29.87	52.03	17.83	36.82	46.90	37.25
Kimi-VL-A3B-Thinking 💡	33.45	17.76	28.11	14.69	41.14	12.64	28.60	43.97	27.08
Orsta-7B (Ours) 💡	**41.65**	31.48	**43.84**	32.82	**54.07**	17.83	**36.91**	41.66	**38.31**
△ (Ours - Backbone)	+2.8	+3.8	+2.6	+3.9	+3.8	+1.5	+0.2	+0.0	+3.2
32B+ Model									
QwenVL-2.5-32B-0321	8.48	12.62	11.99	13.59	15.44	8.61	16.78	14.91	11.87
MM-Eureka-32B 💡	12.20	20.19	21.88	15.86	21.23	15.47	19.95	22.77	18.57
VL-Rethinker-32B 💡	12.16	28.09	22.99	11.89	21.50	15.09	28.10	15.73	19.41
Orsta-32B-0321 (Ours) 💡	**21.33**	**28.55**	**32.23**	**19.44**	**26.38**	**17.78**	**33.20**	**24.18**	**25.94**
△ (Ours - Backbone)	+12.9	+15.9	+20.2	+5.9	+10.9	+9.2	+16.4	+9.3	+14.1
Gemma3-27B †	49.43	42.20	45.46	40.18	49.30	24.96	47.08	58.99	41.82 †
QwenVL-2.5-32B-0326	46.09	32.04	47.55	38.36	61.65	28.43	37.55	50.38	43.67
InternVL-3-38B	46.32	**40.29**	**55.05**	**45.29**	56.63	22.88	**52.04**	**58.04**	**46.69**
Skywork-R1V-38B 💡	25.59	28.45	22.95	19.88	19.53	9.74	22.64	37.55	21.54
Skywork-R1V2-38B 💡	17.08	12.38	15.65	7.14	9.90	17.60	14.29	0.0	15.39
Orsta-32B-0326 (Ours) 💡	**46.78**	37.43	50.86	38.92	**63.14**	28.05	42.68	53.01	45.78
△ (Ours - Backbone)	+0.7	+5.4	+3.3	+0.6	+1.5	-0.4	+5.1	+2.6	+2.1

Fig. 11 shows the MEGA-Bench performance trajectories of three Orsta variants (7B, 32B-0321, 32B-0326) under on-policy and off-policy RL. All variants exhibit stable improvement, with on-policy training generally outperforming off-policy. The 7B model shows smoother and more pronounced gains, while the 32B models experience slower or more variable progress, indicating increased optimization challenges at larger scales.

Qwen2.5-VL-0321, a publicly released checkpoint, exhibits known issues in perception and output formatting but performs reliably in reasoning tasks, as confirmed by our evaluations and VL-Rethinker (Wang et al., 2025a). These issues are addressed in the subsequent 0326 release. We regard the 0321 version as a clean baseline with core knowledge capabilities. As shown in Fig. 12, Orsta-32B-0321 demonstrates that reinforcement learning primarily enhances existing model strengths, serving
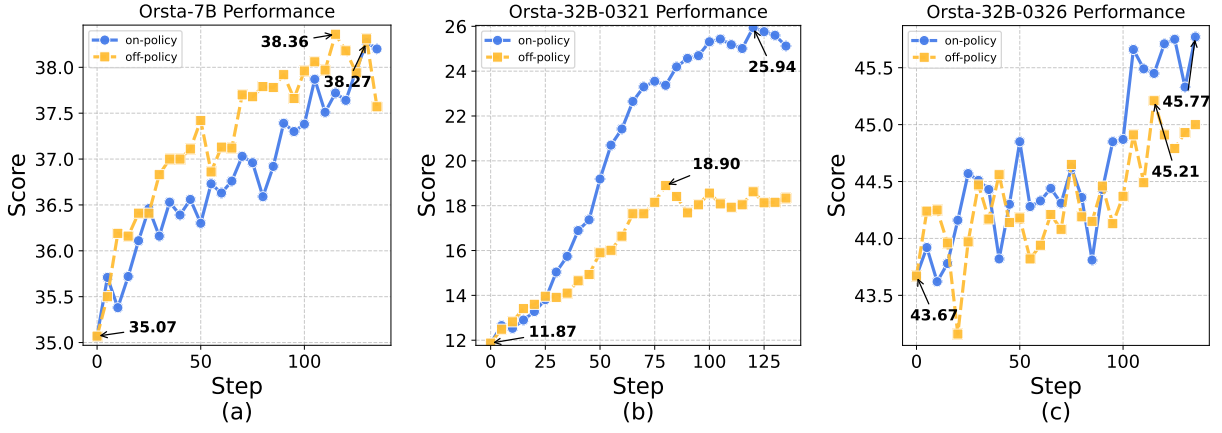
Figure 11 | **Training Trends of On-Policy vs Off-Policy Across Three Model Variants on MEGA-Bench core (7B, 32B-0321, 32B-0326).** Models are evaluated every 5 steps from step 0 to 135. Starting points and peak performances are annotated on the curves.

as an alignment mechanism rather than introducing new capabilities. Performance gains are most notable in in-domain tasks—mathematics, perception, science, and planning, while out-of-domain tasks like coding see limited improvement, underscoring RL's alignment-focused impact.

In summary, our results demonstrate that reinforcement learning effectively enhances both visual reasoning and perception within a unified framework. RL consistently improves performance across MEGA-Bench core's 440 diverse tasks, supporting its role as a general-purpose alignment strategy that unlocks latent capabilities in pretrained vision-language models.

5.2.2. Common Downstream Tasks

As shown in Tab. 2, on the general knowledge benchmark MMMU, Orsta outperforms its backbone by 4% at 7B and 1% at 32B-0326. On the math-focused MathVista benchmark, it achieves over 5% gains across all model sizes. These results align with improvements seen on math tasks in MEGA-Bench, reinforcing Orsta's strength in enhancing reasoning capabilities.

Orsta consistently improves visual perception across benchmarks. On COCO detection, Orsta-7B achieves notable gains (+7.81 mAP and +12.17 mAP@50 for single-object; +3.77 mAP and +5.48 mAP@50 for multi-object), with stronger gains in simpler scenarios. Orsta-32B-0321 shows marked improvements, addressing prior perception issues, while Orsta-32B-0326 yields 3% mAP gains across both subsets. On OVDEval, Orsta-7B and 32B improve by +5.3 and +3.5 mAP, validating the dynamic IoU reward. GUI and OCR tasks (ScreenSpot-Pro, OCRBench) show consistent 1–2% gains. CountBench sees the most significant boost, with Orsta-7B outperforming 32B SFT models and Orsta-32B setting a new state of the art. Overall, V-Triune delivers greater perception improvements for less-aligned base models (0321) than for already-instructed ones (0326).
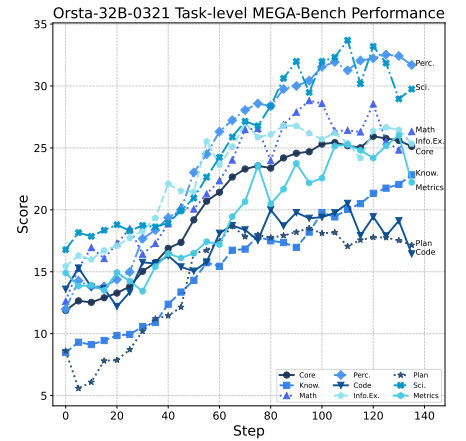

Figure 12 | **Training Trends of Orsta-32B-0321 on MEGA-Bench core.** The dark line denotes the overall MEGA-Bench Core, linking to the performance shown in Fig. 11.

Table 2 | **Benchmarking Orsta Against Baselines on Common Visual Reasoning and Perception Tasks**. For COCO, we report both mAP and mAP@50 as evaluation metrics (mAP|mAP@50), while OVDEval is evaluated using NMS-AP (task-wide mAP). Accuracy serves as the evaluation metric for all remaining tasks.

Tasks	QwenVL-2.5-7B	**Orsta-7B**	QwenVL-2.5-32B 0321	**Orsta-32B 0321**	QwenVL-2.5-32B 0326	**Orsta-32B 0326**
			Visual Reasoning			
$MMMU_{val,rule}$	45.56	49.70	37.11	34.67	39.22	38.00
$MMMU_{val,model}$	54.40	57.10	60.80	64.11	64.20	64.78
$MathVista_{testmini}$	67.50	72.50	70.80	76.30	73.40	76.40
			Visual Perception			
$COCO_{single}$	35.02 | 62.26	42.83 | 74.43	12.19 | 21.43	31.64 | 53.44	40.26 | 68.07	42.12 | 71.55
$COCO_{multiple}$	59.59 | 74.74	63.36 | 80.22	1.61 | 2.00	50.32 | 63.33	62.80 | 78.75	64.28 | 81.19
OVDEval	52.23	57.52	3.84	35.65	56.80	60.36
ScreenSpot-Pro	22.71	23.91	49.46	50.98	51.23	52.69
CountBench	71.69	82.28	83.71	88.59	83.91	88.19
OCRBenchv2	55.11	56.05	44.85	43.78	58.14	59.09

5.3. Training Metric Analysis

V-Triune's source-level metric monitoring enables detailed analysis of cognitive behaviors learned during training, such as reflection patterns. To explore whether perception tasks can benefit from test-time scaling via extended CoT, we analyze Orsta-32B-0321 off-policy logs, which show a marked increase in response length. The analysis focuses on four tasks—Math (MMK12), Puzzle (PuzzleVQA), Detection (V3Det), and OCR (estvqa)—each represented by a single dataset. We report three key metrics: response length, reflection ratio, and accuracy of reflection responses, as defined in Sec. 3.3.

Fig. 13 shows the evolution of key metrics across training steps for the selected tasks. Reasoning tasks (Math and Puzzle) consistently exhibit higher response lengths and reflection ratios than perception tasks (Detection and OCR). For reasoning tasks, response length generally increases over time, with Puzzle showing a steady rise and Math displaying more fluctuation. Among perception tasks, OCR also shows a clear upward trend in response length, while Detection initially decreases before stabilizing around a mean of 325 tokens, without a clear upward trajectory.

The middle row of Fig. 13 shows a clear increase in reflection ratios for Math and Puzzle, indicating a growing use of reflective steps during training. OCR also exhibits an upward trend, though with greater variability. In contrast, Detection maintains a consistently low reflection ratio, near zero, suggesting little engagement in reflective reasoning. In the bottom row, the correctness of reflection responses improves steadily for Puzzle and gradually for OCR, rising from 0.4 to 0.7–0.8. Math shows fluctuating correctness between 0.3 and 0.7, without a clear trend but indicating partial effectiveness. Detection, however, remains at zero throughout, reflecting the absence or failure of reflective responses.

Overall, reasoning tasks and OCR demonstrate increasing reflection usage and, to varying degrees, improved reflection quality. Detection diverges from this pattern, showing minimal reflective behavior and no apparent benefit from longer, reasoning-style responses.
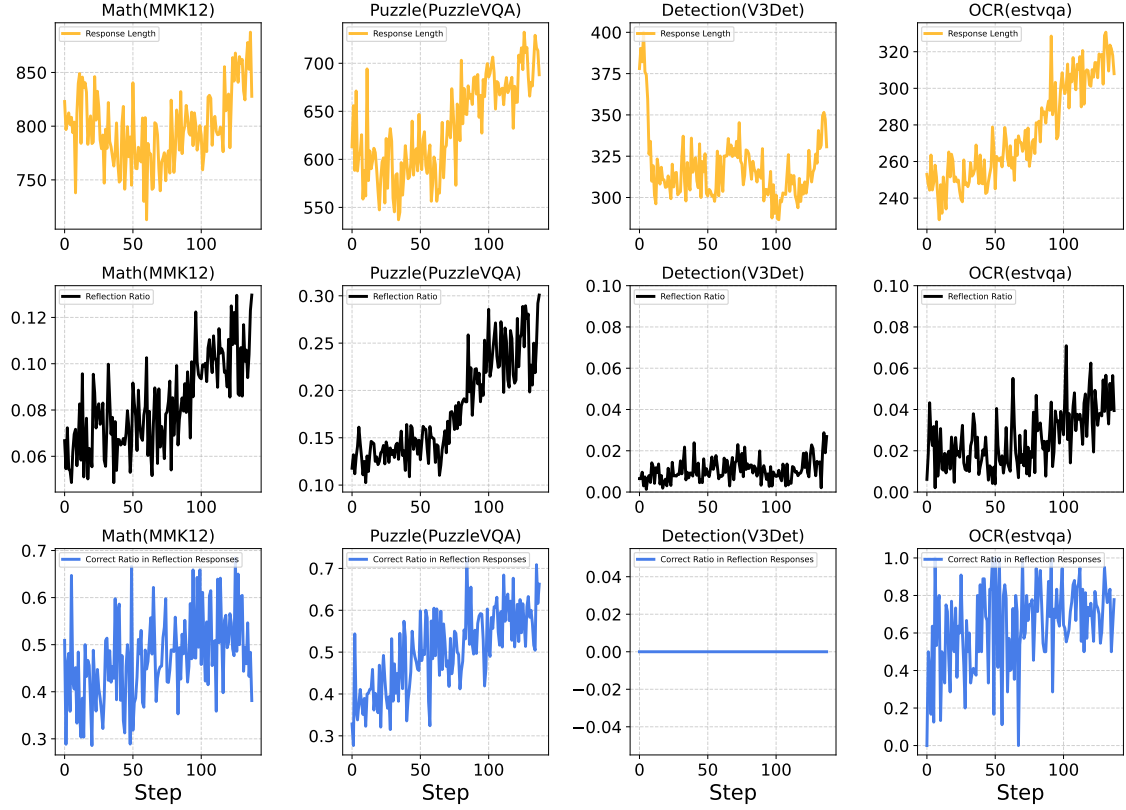
Figure 13 | Training dynamics of response length (top row), reflection ratio (middle row), and correct ratio in reflection responses (bottom row) during training steps for Math (MMK12), Puzzle (PuzzleVQA), Detection (V3Det), and OCR (estvqa) tasks using the Orsta-32B-0321 off-policy setting. Each column corresponds to a different task, and each row represents a distinct metric.

5.4. Ablation Study

Training-strategy ablation As shown in Fig. 14a, updating only the LLM stack raises the score from approximately 35 to 38 by step 120, indicating that reasoning weights are the primary driver of performance improvement. In contrast, tuning only the vision backbone results in negligible gains, with scores plateauing around 35.5. When both branches are optimized jointly, performance initially mirrors the LLM-only trajectory but slightly surpasses it in later stages (peaking at approximately 38.5).

Task-composition ablation Training on both reasoning and perception data yields the strongest performance, reaching approximately 37.5. A reasoning-only curriculum closely trails (within 0.3 points) and briefly surpasses it mid-training, suggesting that the benchmark prioritizes logical competence over pure perception. Perception-only training consistently lags by 0.7–1.0 points but still shows steady improvement, indicating that visual supervision provides transferable alignment signals. The consistent performance hierarchy (Reasoning+Perception > Reasoning > Perception) underscores the value of mixed-task corpora: combining complementary signals leads to additive gains rather than diluted optimization.

Learning-rate ablation For the Orsta-32B model, conservative step sizes are essential. A learning rate of 1e-6 yields the highest and most stable plateau (45.5), while 1.5e-6 performs similarly until a mild degradation after 80 steps. Increasing the rate to 2e-6 causes a late-stage collapse to 38, and 3e-6 diverges catastrophically after 50 steps, dropping below 36. This pattern suggests that larger
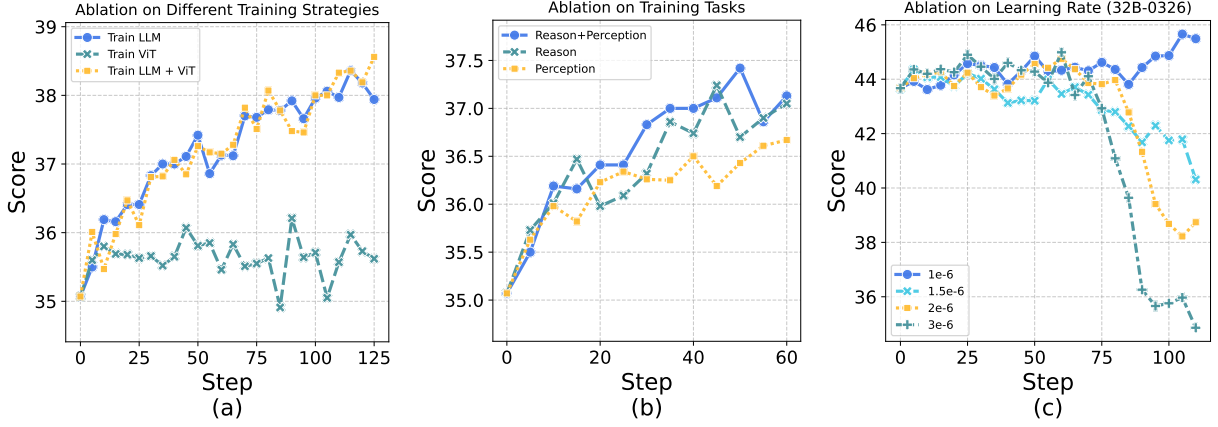
Figure 14 | **Ablation Study on Training Strategies (a), Task Composition (b), and Learning Rates (c).** (a) Various training strategies evaluated on the 7B model (LLM-only, ViT-only, joint training); (b) Different task compositions evaluated on the 7B model (reasoning, perception, or both); (c) Effects of learning rates (1e-6, 1.5e-6, 2e-6, 3e-6) on the performance of the 32B model. Each subfigure shows the score progression over training steps.

Model	Size	Backbone	Math	Science	Chart	Puzzle	OCR	DET	GND	CNT	Others
Visual-RFT (Liu et al., 2025e)	2B	Qwen2-VL	-	-	-	-	-	✓	✓	-	CLS
DeepPerception (Ma et al., 2025a)	2B	Qwen2-VL Base*	-	-	-	-	-	-	-	-	S.QA
Vision-R1 (Huang et al., 2025)	7B,72B	Qwen2.5-VL	✓	✓	-	-	-	-	-	-	-
R1-Onevision (Yang et al., 2025)	7B	Qwen2.5-VL	✓	✓	✓	-	-	-	-	-	V.QA
Reason-RFT (Tan et al., 2025)	2B,7B	Qwen2-VL	✓	-	-	-	-	-	-	✓	S.QA
OThink-MR1 (Liu et al., 2025d)	2B,7B	Qwen2.5-VL	✓	-	-	-	-	-	-	✓	-
Perception-R1 (Yu et al., 2025a)	2B,3B	Qwen2 & 2.5-VL	-	-	-	-	✓	✓	✓	-	-
VLM-R1 (Shen et al., 2025)	3B	Qwen2.5-VL	-	-	-	-	-	✓	✓	-	-
MM-EUREKA (Meng et al., 2025)	7B,32B	Qwen2.5-VL	✓	-	-	-	-	-	-	-	S.QA
VL-Rethinker (Wang et al., 2025a)	7B,32B,72B	Qwen2.5-VL	✓	✓	✓	-	-	-	-	-	S.QA
Orsta (ours)	7B,32B	Qwen2.5-VL	✓	✓	✓	✓	✓	✓	✓	✓	-

Table 3 | **Task-wise Comparison of Public VLM-RL Models with Varying Model Configurations.** The evaluated tasks include Math, Science, Chart, Puzzle, OCR, Detection, Grounding, Counting, Classification, Spatial question answering, and Visual question answering. Backbones marked with a star* indicate models trained from pre-trained backbone (i.e. zero-RL setting), while others are trained from instructed models.

models sit closer to the edge of the loss landscape, benefiting from small, stable updates, with 1e-6 offering the best trade-off between convergence speed and final performance on MEGABench.

6. Related Work

Adopted from DeepSeek R1 (Guo et al., 2025), recent advancements in visual reinforcement learning (summarized in Table 3) have explored diverse strategies to enhance multimodal reasoning and perception. Models such as Vision-R1 (Huang et al., 2025), LMM-R1 (Peng et al., 2025), R1-

OneVision (Yang et al., 2025), VisualThinker-R1-Zero (Zhou et al., 2025), and MM-Eureka (Meng et al., 2025) primarily focus on strengthening reasoning capabilities through instruction tuning or reinforcement learning, leveraging CoT datasets, image-to-text conversion, or rule-based signals to elicit problem-solving behavior. In parallel, models like Visual-RFT (Liu et al., 2025e), R1-V (Chen et al., 2025), Reason-RFT (Tan et al., 2025), and DeepPerception (Ma et al., 2025a) apply task-specific, verifiable reward signals (e.g., IoU, mAP, math_verify) to improve perception tasks such as detection, grounding, and counting. Seg-Zero (Liu et al., 2025b), Perception-R1 (Yu et al., 2025a), and VLM-R1 (Shen et al., 2025) further propose tailored reward functions for segmentation and OCR, though they largely remain within task-specific generalization boundaries. To address broader cross-task generalization, OThink-MR1 (Liu et al., 2025d) incorporates GRPO with Dynamic KL, while VL-Rethinker (Wang et al., 2025b) introduces selective sample replay and forced rethinking to enhance reasoning capabilities. Additionally, several general-purpose VLMs have demonstrated the promise of visual reinforcement learning (Team, 2025; Team et al., 2025a). In contrast to prior efforts that treat reasoning and perception in isolation, our proposed V-Triune unifies both domains under a single reinforcement learning system.

7. Discussion & Future Work

In this paper, we propose V-Triune, the first visual triple-unified reinforcement learning system for vision-language models (VLMs) that effectively scales across both reasoning and perception tasks. Unlike previous approaches that are often limited to single-domain or lightweight tasks, V-Triune achieves substantial performance improvements across a broad spectrum of real-world vision-language challenges through its three-tier component design and dynamic IoU-based reward.

Notably, V-Triune yields Orsta-3B and Orsta-32B, with improvements of 3.2% and 2.1% over strong backbones, respectively. Our experiments indicate that reinforcement learning in VLMs primarily serves as an alignment strategy, refining the model's decision-making and response behavior rather than enabling new knowledge acquisition. This supports the notion that RL fine-tuning enhances the utility and robustness of pre-trained VLMs without altering their foundational capabilities.

We also acknowledge several limitations that merit further investigation:

- **Limited performance scaling in perception tasks:** For visual reasoning tasks, we observe clear trends of increasing response length and reflection ratio as training progresses, akin to test-time scaling laws observed in LLMs. However, such trends are not evident in visual perception tasks. The underlying factors driving performance improvements in perception remain unclear. Exploring multi-step RL for perception tasks, similar to approaches in OpenAI's o3, may offer new insights and avenues for progress.
- **Underexplored potential of RL-zero in VLMs:** There is currently limited research on RL-zero for vision-language models. Nonetheless, we see early indications that RL-zero has the potential to surpass the current limitations imposed by supervised fine-tuning (SFT). Given that multi-modal alignment remains a fundamental challenge in VLMs, and that supervised fine-tuning (SFT) may be inherently limited in addressing this issue, we believe that RL-zero has the potential to redefine the optimization paradigm and unlock new capabilities for vision-language models.

The V-Triune system and Orsta models will be publicly available at https://github.com/MiniMax-AI. We hope this work inspires further research into reinforcement learning as a general-purpose training paradigm for vision-language understanding, and encourages exploration of richer reward schemes, advanced reasoning strategies, and task-specific adaptations.

Contributions

Core Contributions

Yan Ma*[1,4], Linge Du*[1,3], Xuyang Shen*†[1], ✉Junjie Yan[1]

Contributions

Shaoxiang Chen[1], Pengfei Li[1], Qibing Ren[1,2]

Advisor

Junjie Yan[1], Pengfei Liu[2,4], Yuchao Dai[3], Lizhuang Ma[2]

Affiliation

[1] MiniMax

[2] Shanghai Jiao Tong University

[3] Northwestern Polytechnical University

[4] Generative Artificial Intelligence Lab (GAIR)

* Equal Contribution; † Project Lead; ✉ Corresponding Author

References

Apache Software Foundation. Apache parquet documentation. https://parquet.apache.org/docs/, 2025. Accessed: 2025-05-20.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

Ekaterina Borisova and Georg Rehm. Scivqa: Scientific visual question answering. *SDProc 2025*, 2025. URL https://sdproc.org/2025/scivqa.html.

Jiacheng Chen, Tianhao Liang, Sherman Siu, Zhengqing Wang, Kai Wang, Yubo Wang, Yuansheng Ni, Wang Zhu, Ziyan Jiang, Bohan Lyu, et al. Mega-bench: Scaling multimodal evaluation to over 500 real-world tasks. *arXiv preprint arXiv:2410.10563*, 2024.

Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V, 2025. Accessed: 2025-02-02.

Yew Ken Chia, Vernon Toh Yan Han, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. Puzzlevqa: Diagnosing multimodal reasoning challenges of language models with abstract visual patterns. *arXiv preprint arXiv:2403.13315*, 2024.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88:303–338, 2010.

Ling Fu, Biao Yang, Zhebin Kuang, Jiajun Song, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Mingxin Huang, et al. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*, 2024.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Deepanway Ghosal, Vernon Toh Yan Han, Yew Ken Chia, and Soujanya Poria. Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning. *arXiv preprint arXiv:2403.03864*, 2024.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63 (11):139–144, 2020.

Google DeepMind. Gemini model thinking updates: March 2025, 2025. URL https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/. Accessed: May 11, 2025.

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.

Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, et al. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*, 2024.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.

HuggingFace. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720. URL https://doi.org/10.48550/arXiv.2412.16720.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*, 2024.

Hynek Kydlíček. Math-verify: A library for rule-based verification of mathematical answers, 2025. URL https://github.com/huggingface/Math-Verify.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic,

November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-demo.21.

Aonian Li, Bangwei Gong, Bo Yang, Boji Shan, Chang Liu, Cheng Zhu, Chunhao Zhang, Congchao Guo, Da Chen, Dong Li, et al. Minimax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025a.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. Screenspot-pro: Gui grounding for professional high-resolution computer use. *arXiv preprint arXiv:2504.07981*, 2025b.

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 *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.

Qianchu Liu, Sheng Zhang, Guanghui Qin, Timothy Ossowski, Yu Gu, Ying Jin, Sid Kiblawi, Sam Preston, Mu Wei, Paul Vozila, et al. X-reasoner: Towards generalizable reasoning across modalities and domains. *arXiv preprint arXiv:2505.03981*, 2025a.

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*, 2025b.

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*, 2025c.

Zhiyuan Liu, Yuting Zhang, Feng Liu, Changwang Zhang, Ying Sun, and Jun Wang. Othink-mr1: Stimulating multimodal generalized reasoning capabilities via dynamic reinforcement learning. *arXiv preprint arXiv:2503.16081*, 2025d.

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*, 2025e.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *CoRR*, 2023.

Xinyu Ma, Ziyang Ding, Zhicong Luo, Chi Chen, Zonghao Guo, Derek F Wong, Xiaoyi Feng, and Maosong Sun. Deepperception: Advancing r1-like cognitive visual perception in mllms for knowledge-intensive visual grounding. *arXiv preprint arXiv:2503.12797*, 2025a.

Yan Ma, Steffi Chern, Xuyang Shen, Yiran Zhong, and Pengfei Liu. Rethinking rl scaling for vision language models: A transparent, from-scratch framework and comprehensive evaluation scheme. *arXiv preprint arXiv:2504.02587*, 2025b.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, et al. Chartqapro: A more diverse and challenging benchmark for chart question answering. *arXiv preprint arXiv:2504.05506*, 2025.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, et al. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

OpenAI. Introducing openai o3 and o4-mini, April 2025. URL https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2025-04-18.

OpenAI. Thinking with images, 2025. URL https://openai.com/index/thinking-with-images/. Accessed: 2025-05-23.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3170–3180, 2023.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025.

Sebastián Ramírez. Fastapi. https://fastapi.tiangolo.com, 2021. URL https://github.com/fastapi/fastapi. FastAPI framework, high performance, easy to learn, fast to code, ready for production.

John Schulman. Approximating kl divergence, 2020. URL http://joschu.net/blog/kl-approx.html. Accessed: 2025-05-23.

Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8430–8439, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Yueqi Song, Tianyue Ou, Yibo Kong, Zecheng Li, Graham Neubig, and Xiang Yue. Visualpuzzles: Decoupling multimodal reasoning evaluation from domain knowledge. *arXiv preprint arXiv:2504.10342*, 2025.

Kai Sun, Yushi Bai, Ji Qi, Lei Hou, and Juanzi Li. Mm-math: Advancing multimodal math evaluation with process evaluation and fine-grained classification. *arXiv preprint arXiv:2404.05091*, 2024.

Huajie Tan, Yuheng Ji, Xiaoshuai Hao, Minglan Lin, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Reason-rft: Reinforcement fine-tuning for visual reasoning. *arXiv preprint arXiv:2503.20752*, 2025.

ByteDance Seed Team. Seed1.5-vl technical report. *arXiv preprint arXiv:2505.07062*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025a.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, Guokun Lai, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haoning Wu, Haotian Yao, Haoyu Lu, Heng Wang, Hongcheng Gao, Huabin Zheng, Jiaming Li, Jianlin Su, Jianzhou Wang, Jiaqi Deng, Jiezhong Qiu, Jin Xie, Jinhong Wang, Jingyuan Liu, Junjie Yan, Kun Ouyang, Liang Chen, Lin Sui, Longhui Yu, Mengfan Dong, Mengnan Dong, Nuo Xu, Pengyu Cheng, Qizheng Gu, Runjie Zhou, Shaowei Liu, Sihan Cao, Tao Yu, Tianhui Song, Tongtong Bai, Wei Song, Weiran He, Weixiao Huang, Weixin Xu, Xiaokun Yuan, Xingcheng Yao, Xingzhe Wu, Xinxing Zu, Xinyu Zhou, Xinyuan Wang, Y. Charles, Yan Zhong, Yang Li, Yangyang Hu, Yanru Chen, Yejie Wang, Yibo Liu, Yibo Miao, Yidao Qin, Yimin Chen, Yiping Bao, Yiqin Wang, Yongsheng Kang, Yuanxin Liu, Yulun Du, Yuxin Wu, Yuzhi Wang, Yuzi Yan, Zaida Zhou, Zhaowei Li, Zhejun Jiang, Zheng Zhang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Zijia Zhao, and Ziwei Chen. Kimi-VL technical report, 2025b. URL https://arxiv.org/abs/2504.07491.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025a.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*, 2025b.

Xinyu Wang, Yuliang Liu, Chunhua Shen, Chun Chet Ng, Canjie Luo, Lianwen Jin, Chee Seng Chan, Anton van den Hengel, and Liangwei Wang. On the general value of evidence, and bilingual scene-text visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10126–10135, 2020.

Lilian Weng. Reward hacking in reinforcement learning. *lilianweng.github.io*, Nov 2024. URL https://lilianweng.github.io/posts/2024-11-28-reward-hacking/.

Renqiu Xia, Bo Zhang, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Peng Ye, Min Dou, Botian Shi, et al. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *arXiv preprint arXiv:2402.12185*, 2024.

Shijie Xia, Yiwei Qin, Xuefeng Li, Yan Ma, Run-Ze Fan, Steffi Chern, Haoyang Zou, Fan Zhou, Xiangkun Hu, Jiahe Jin, et al. Generative ai act ii: Test time scaling drives cognition engineering. *arXiv preprint arXiv:2504.13828*, 2025.

Chi Xie, Zhao Zhang, Yixuan Wu, Feng Zhu, Rui Zhao, and Shuang Liang. Described object detection: Liberating object detection with flexible expressions. *Advances in Neural Information Processing Systems*, 36:79095–79107, 2023.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

Yiyang Yao, Peng Liu, Tiancheng Zhao, Qianqian Zhang, Jiajia Liao, Chunxin Fang, Kyusong Lee, and Qing Wang. How to evaluate the generalization of detection? a benchmark for comprehensive open-vocabulary detection. *arXiv preprint arXiv:2308.13177*, 2023.

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*, 2025a.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025b.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024. URL https://arxiv.org/abs/2407.12772.

Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025.