Smooth Operator: Smooth Verifiable Reward Activates Spatial Reasoning Ability of Vision-Language Model

Anonymous ACL submission

Abstract

Vision-Language Models (VLMs) remain limited in high-precision numerical prediction for 3D scene understanding. In continuous numerical tasks, existing reinforcement learning methods such as GRPO are sample-inefficient: binary rewards fail to distinguish near-miss predictions from incorrect ones, while group-relative normalization can further weaken informative reward differences when sample groups are uniformly poor. Consequently, near-miss samples contribute little to policy updates despite carrying valuable learning signal. We propose Smooth Numerical Reward Activation (SNRA) and Absolute-Preserving GRPO (AP-GRPO) to address this issue. SNRA converts raw numerical errors into dense rewards via a dynamically parameterized sigmoid, while AP-GRPO modulates group-relative advantages with an absolute reward factor to emphasize informative near-miss samples and suppress uniformly uninformative groups. This improves the effective training contribution of near-miss predictions and substantially enhances data efficiency. On our Numerical3D-50k dataset, AP-GRPO achieves performance comparable to large-scale supervised learning across five benchmarks and four VLM architectures, without architectural modifications.

1 Introduction

Vision-Language Models (VLMs) excel in 2D visual understanding, often rivaling or exceeding human accuracy across diverse benchmarks. However, in genuine 3D scene understanding, these models reveal a critical weakness: they struggle with precise numerical estimation and geometric reasoning, such as depth perception, inter-object distance calculation, and object sizing (Chen et al., 2024a; Yang et al., 2024; Liu et al., 2025a; Yang et al., 2025a; Cheng et al., 2024). This limitation stems from a reliance on 2D visual encoders that lack explicit geometric priors and depend instead on statistical patterns from planar images.

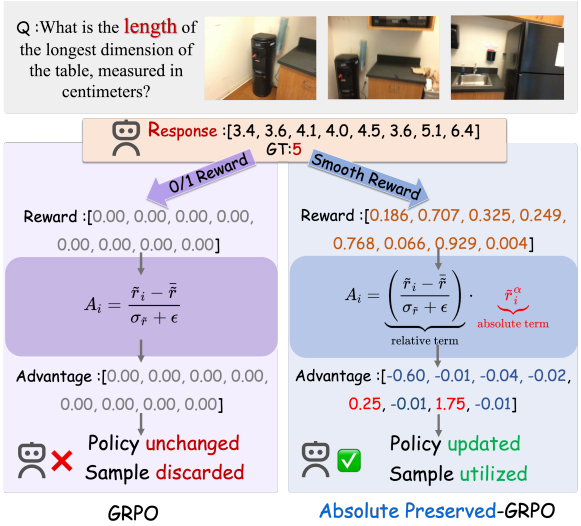


Figure 1: Standard GRPO vs. AP-GRPO (Ours). Standard GRPO (left) assigns binary rewards, yielding zero advantages for near-correct responses and discarding all samples. AP-GRPO (right) uses SNRA to produce dense rewards and modulates advantages by the absolute reward term, recovering near-miss samples' training influence through gradient budget redistribution across groups.

Existing approaches to enhancing 3D capabilities fall into two categories. The first involves feature augmentation, incorporating external geometric representations (e.g., depth maps, point clouds, or features from encoders like VGGT) as additional inputs (Wang et al., 2025; Fan et al., 2025; Cheng et al., 2024; Zhu et al., 2025). While effective, these methods introduce substantial overhead and reliance on auxiliary 3D modalities. The second strategy relies on large-scale training datasets to implicitly acquire 3D understanding (Chen et al., 2024a; Jia et al., 2024; Daxberger et al., 2025; Cheng et al., 2024), but demands enormous data volumes and prohibitive training costs.

In this work, we argue that verifiable numerical signals grounded in physical laws offer a promising pathway for bridging the 2D-to-3D perceptual gap. However, leveraging such signals via rein-

forcement learning (RL) faces a challenge we term the *near-miss waste problem*. Under binary 0-1 rewards, near-miss predictions (e.g., 2.1m when the answer is 2.0m) receive the same zero reward as completely wrong ones (e.g., 50m), discarding all error-magnitude information. A natural fix is to use smooth, dense rewards. However, simply plugging dense rewards into GRPO (Shao et al., 2024) does not help, because GRPO's group-relative normalization produces equal-magnitude gradients from uniformly-poor groups and informative groups alike, drowning out near-miss signals with noise.

To address this, we introduce two complementary components: the Smooth Numerical Reward Activation Operator (SNRA) and Absolute-Preserving GRPO (AP-GRPO). SNRA is a dynamically parameterized sigmoid-based reward transformation that maps raw verifiable signals into a dense reward space within $[0, 1]$, progressively sharpening from tolerant to strict as training proceeds. AP-GRPO then modulates each sample's group-relative advantage by the absolute reward term $\tilde{r}_i^{\alpha}$. The key mechanism operates at the *group level*: uniformly-poor groups (all $\tilde{r}_i \approx 0$) see their entire gradient contribution compressed to near zero, while groups containing near-miss samples retain their contributions. This redistributes gradient budget from noise-dominated groups to informative groups, recovering near-miss samples' effective training influence — not by amplifying individual advantages (which decrease slightly), but by eliminating competing noise from uninformative groups (Appendix B.2).

Through the combined effect of SNRA and AP-GRPO, previously underutilized near-miss samples are reactivated, improving data efficiency. To fully leverage this, we construct Numerical3D-50k, a compact dataset of roughly 50k samples derived from complex 3D spatial queries with verifiable numerical (e.g., depth, distance, size) and relational (e.g., direction, spatial ordering) subtasks. Notably, our method preserves the standard VLM architecture without adding encoders, modalities, or structural modifications.

The main contributions of this work are summarized as follows:

- **Numerical3D-50k Dataset.** A compact dataset of approximately 50k samples derived from existing 3D benchmarks, decomposed into verifiable numerical and discrete relational subtasks with deterministic, physics-grounded supervision.
- **SNRA and AP-GRPO.** A dynamic sigmoid-based reward activation operator for densifying sparse physical signals, and a modified GRPO framework that recovers near-miss samples' training influence through group-level gradient budget redistribution — multiplicatively modulating advantages by the absolute reward to suppress uniformly-poor groups while preserving informative ones (Appendix B.3).
- **Empirical Evidence.** AP-GRPO outperforms six recent RL baselines under matched settings and achieves competitive spatial reasoning across five benchmarks and four VLM architectures, approaching models trained on 80× more supervised data, without architectural modifications.

2 Related Works

2.1 Visual Spatial Understanding and Reasoning

Despite strong progress of LVLMs on many visual tasks (**?**Bai et al., 2025; Chen et al., 2024b; Wang et al., 2024), spatial understanding and reasoning remain challenging (Yang et al., 2025a; Ray et al., 2024; Cheng et al., 2024). Prior work addresses this from two angles (Qian et al., 2025a,d,c,b, 2024). On the understanding side, methods such as SpatialVLM (Chen et al., 2024a), SpatialRGPT (Cheng et al., 2024), SAT (Ray et al., 2024), and SpatialBot (Cai et al., 2025) improve spatial perception through expert-constructed data, depth augmentation, or simulator synthesis, while later works (Ha and Song, 2022; Liu et al., 2025a; Yang et al., 2025a,d; Zhu et al., 2025) further scale perception with larger datasets. On the reasoning side, SpaceR (Ouyang et al., 2025), MindCube (Yin et al., 2025), MVoT (Li et al., 2025a), SpatialReasoner (Ma et al., 2025), and ViLaSR (Wu et al., 2025) strengthen multi-step spatial inference through cognitive maps, multimodal traces, or visual tools. Recent RL frameworks for 3D reasoning (Pan and Liu, 2025; Huang et al., 2025; Liu et al., 2025b; Sun and Chen, 2025) further explore this direction. Unlike prior work relying on relative ranking or massive self-supervision, our approach leverages verifiable numerical signals with absolute-preserving optimization.

2.2 VLM Reinforcement Learning with Verifiable Rewards

Inspired by DeepSeek-R1 (Guo et al., 2025), post-training RL has emerged as a key paradigm to unlock complex reasoning in LVLMs (**?**Bai et al., 2025). Reinforcement Learning with Verifiable Rewards (RLVR) replaces noisy neural reward models with objective, checkable criteria (Thawakar et al., 2025; Pan et al., 2025). Recent work has evolved from simple outcome verification to process-aware mechanisms: Perception-R1 (Xiao et al., 2025) and Vision-SR1 (Li et al., 2025c) introduce perception rewards for visual grounding, while StructVRM (Zhang et al., 2025b) and Argos (Tan et al., 2025) employ agentic verifiers for sub-question-level partial credit. Despite these advances, continuous numerical reasoning remains challenging. Standard algorithms like GRPO (Guo et al., 2025; Zhang et al., 2025c) treat near-miss samples identically to failures under binary rewards, and simply adopting denser rewards does not help — group-relative normalization produces equal-magnitude updates from uninformative and informative groups alike. Our SNRA and AP-GRPO address this by making near-miss samples distinguishable and redistributing gradient budget so that informative groups dominate the policy update.

3 Method

3.1 3D Perceptual Subtasks and Numerical3D-50k Dataset

We decompose 3D scene understanding into two complementary subtasks, each leveraging distinct physical ground-truth signals, and train our method on the curated **Numerical3D-50k** dataset.

Each sample includes a visual input x, a natural-language query q, a subtask identifier $\tau \in \mathcal{T}_{\text{num}} \cup \mathcal{T}_{\text{disc}}$, and verifiable geometric annotations. To ensure a unified optimization landscape during reinforcement learning, we map the feedback from both subtask types into a generalized error space $\mathcal{E}$, where $e_i \in [0, +\infty)$ represents the discrepancy between the prediction and the ground truth.

The subtasks are categorized as follows:

(i) **Numerical subtasks** ($\tau \in \mathcal{T}_{\text{num}}$), requiring precise scalar outputs (e.g., depth, distance, size). The raw error is calculated as the squared physical distance $e_i = (\hat{y}_i - y)^2$. For invalid outputs (e.g., parsing failures), a task-specific maximum penalty $e_{\max}$ is assigned.

(ii) **Discrete relational subtasks** ($\tau \in \mathcal{T}_{\text{disc}}$), involving categorical spatial reasoning (e.g., direction, ordering). Instead of directly assigning a reward score, we define a logical error e_i derived from the task-specific verification $V_\tau(o_i) \in [0, 1]$. Specifically, the error is mapped via $e_i = \Phi(1 - V_\tau(o_i))$, where Φ is a scaling function that ensures discrete failures (score 0) correspond to the high-penalty region of the SNRA operator, while partial credits are mapped to corresponding gradient-sensitive zones.

Subtasks are algorithmically generated for each scene-query pair using deterministic verifiers V_τ, yielding dense and interpretable supervision suitable for reinforcement learning without model modifications. A more detailed definition of the discrete subtasks and verifier design is provided in Appendix F.

The **Numerical3D-50k** dataset is built by integrating spatial reasoning benchmarks (MindCube (Wu et al., 2025), VLM-3R (Fan et al., 2025), VSI-590K (Yang et al., 2025c)) and richly annotated 3D datasets (ScanNet (Dai et al., 2017), ScanNet++ (Yeshwanth et al., 2023), ARKitScenes (Dehghan et al., 2021), SR-91k (Cheng et al., 2024)), from which we generate metric-scale annotations via geometric projection and template synthesis. We select 50k high-quality samples with balanced subtask distribution and verifiable ground truth (see Appendix C for the full breakdown). The dataset will be publicly released upon acceptance.

3.2 Smooth Numerical Reward Activation Operator (SNRA)

In conventional supervised numerical regression, the geometric alignment error is typically measured by the squared Euclidean distance between the predicted coordinate $\hat{y}_i$ and the ground truth y:

$$e_i = |\hat{y}_i - y|^2. \tag{1}$$

However, in reinforcement learning for 3D spatial reasoning, directly using this raw verification signal as reward poses numerical and structural challenges for autoregressive models: e_i is unbounded (while rewards are usually expected to lie in $[0, 1]$) and exhibits high variance across training stages. Large deviations during early exploration can cause gradient spikes, whereas diminishing errors near convergence lead to vanishing gradients, hindering fine-grained precision.

To mitigate these issues, we propose the Smooth Numerical Reward Activation (SNRA) operator,

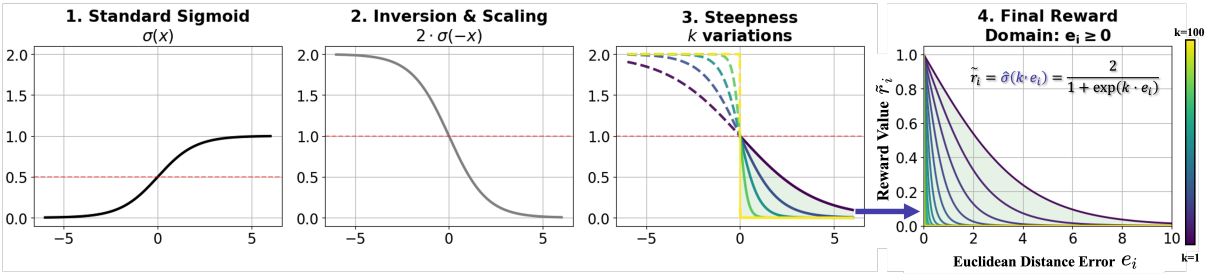


Figure 2: Derivation and mechanism of the SNRA operator. The roadmap depicts the transformation from a standard sigmoid $\sigma(x)$ (Stage 1) to the final reward mapping (Stage 4) via symmetry inversion and amplitude scaling. This formulation anchors the operator at $f(0) = 1$ for perfect predictions. The sharpness k acts as a dynamic soft-threshold: lower k provides smooth gradients to encourage exploration, while higher k enforces strict numerical compliance for precision convergence.

which transforms the raw error into a bounded and smoothly varying reward. As shown in Figure 2 (Stages 1–3), we start from the standard sigmoid $\sigma(x) = (1 + e^{-x})^{-1}$ (Stage 1). We then apply an inversion and amplitude scaling (Stage 2) to obtain

$$f(x) = 2 \cdot \sigma(-x) = \frac{2}{1 + e^{x}}, \tag{2}$$

which is monotonically decreasing and satisfies $f(0) = 1$, assigning the maximal reward to perfect predictions. Next, we introduce a sharpness parameter k (Stage 3) to control the decay rate with respect to the error magnitude: smaller k yields smoother reward shaping to encourage exploration, while larger k produces steeper decay to promote high-precision convergence. By restricting the domain to the physical error space $e_i \geq 0$ and substituting $x = k \cdot e_i$, we define the reward activation as:

$$\hat{\sigma}(k, e_i) = \frac{2}{1 + \exp(k \cdot e_i)}, \quad k > 0, \; e_i \geq 0. \tag{3}$$

In our framework, the numerical reward component $\tilde{r}_i$ enforces numerical precision across subtasks:

$$\tilde{r}_i = \begin{cases} \hat{\sigma}\left(k(t), e_i^{\text{con}}\right), & \text{for continuous subtasks} \\ \hat{\sigma}\left(k(t), e_i^{\text{disc}}\right), & \text{for discrete subtasks} \end{cases} \tag{4}$$

where $k(t)$ is a dynamic sharpness schedule, and e_i^{con} and e_i^{disc} denote the error metrics for continuous and discrete subtasks, respectively. The final reward is constructed as a composite signal:

$$R_i = (1 - \lambda)\tilde{r}_i + \lambda \cdot r_{fmt}, \tag{5}$$

where $r_{fmt} \in \{0, 1\}$ is a binary format reward for structural compliance, and λ is the balancing coefficient.

3.3 Dynamic Sharpness Scheduling

The efficacy of SNRA is governed by the sharpness $k(t)$, which controls reward sensitivity to the error e_i. Instead of treating k as a fixed hyperparameter, we introduce Dynamic Sharpness Scheduling, a non-linear gradient amplifier that gradually shifts optimization from global exploration to local precision refinement. As illustrated in Stage 4 of Figure 2, this mechanism demonstrates the progressive transformation of the reward landscape. Early in training, a small $k(t)$ produces a diffused reward surface with broad gradient support, providing non-zero feedback even when predictions are far from the target and preventing stalls during global coordinate search. As training proceeds, we monotonically increase $k(t)$ to sharpen the reward peak around $e_i = 0$, concentrating gradients on subtle numerical errors and promoting high-precision alignment.

We implement this coarse-to-fine curriculum with a centered, smoothed sigmoid schedule:

$$k(t) = k_{\min} + (k_{\max} - k_{\min}) \cdot \sigma\left(s\left(\frac{t}{T} - \tau\right)\right), \tag{6}$$

where t and T are the current and total steps. $\tau \in (0, 1)$ sets the transition center and $s > 0$ controls its steepness. We use $\tau = 0.5$ and $s = 10$. Thus, $k(t)$ stays near $k_{\min}$ early, rises around $t \approx \tau T$, and approaches $k_{\max}$ late in training. In all experiments, $k_{\min} = 1.0$ and $k_{\max} = 100.0$.

Coupling sharpness to training progress induces **reward hardening**. As $k \to \infty$, the smooth surrogate $\tilde{r}_i$ approaches a binary verification signal:

$$\tilde{r}_i \xrightarrow{k \to \infty} \mathbb{I}_{e_i} = \begin{cases} 1, & \text{if } e_i = 0, \\ 0, & \text{if } e_i > 0. \end{cases} \tag{7}$$

This lets the agent first learn global geometry under

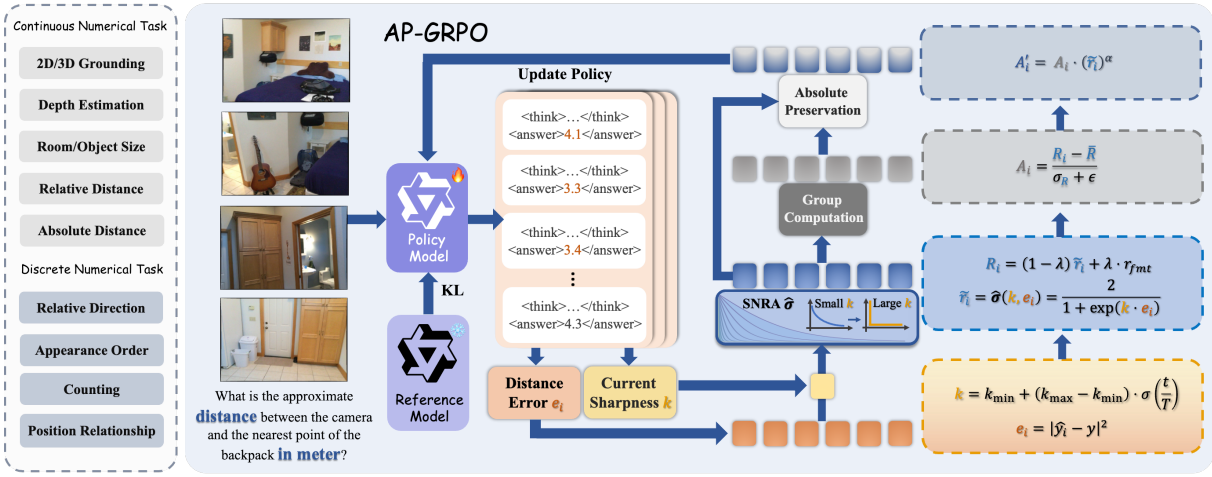


Figure 3: Overview of the AP-GRPO framework. The method processes continuous (e.g., grounding, depth, size, distance) and discrete numerical tasks (e.g., direction, order, counting, position) via a policy model that generates reasoning and answers. Raw rewards are smoothed by SNRA into dense R'_i, then used to compute group-relative advantages modulated by absolute preservation $(R'_i)^\alpha$, yielding A'_i. Policy updates employ a clipped objective with KL regularization for stable, precise 3D perception.

permissive rewards, then face increasingly strict precision requirements, balancing exploration stability with accurate reasoning.

3.4 Absolute-Preserving GRPO (AP-GRPO)

Standard Group Relative Policy Optimization (GRPO) estimates the advantage function through intra-group standardization:

$$A_i = \frac{R_i - \bar{R}}{\sigma_R + \epsilon}, \tag{8}$$

where $\bar{R}$ and σ_R are the mean and standard deviation of rewards within a group of G samples generated from the same query. By enforcing $\sum A_i = 0$, this formulation ensures that the policy update depends only on the relative ranking of samples within the group, increasing the probability of those that outperform the local average.

While relative ranking is effective in high-quality regimes such as language tasks, it encounters a specific pathology in continuous numerical tasks. During early 3D training, most generated predictions are far from the target, so rewards within a group cluster near zero. In this regime, the standardization amplifies noise — tiny random differences among uniformly poor predictions are inflated into large advantages, causing the policy to reinforce the "best of the worst." We call this the **uniformly-poor group trap**: GRPO wastes gradient budget on noisy relative rankings that carry no meaningful precision signal. In a typical early-training scenario, uniformly-poor groups consume nearly half of the total gradient budget while contributing no useful learning signal (see Appendix B.2 for a quantitative analysis).

To address this pathology, we propose **Absolute-Preserving GRPO (AP-GRPO)**, which recovers near-miss samples' effective training influence by redistributing gradient budget across groups. We redefine the advantage function as the product of a relative ranking term and an absolute scaling term:

$$A'_i = \underbrace{\left(\frac{R_i - \bar{R}}{\sigma_R + \epsilon}\right)}_{\text{Relative term}} \cdot \underbrace{\tilde{r}_i^\alpha}_{\text{Absolute term}}, \quad \alpha \geq 1, \tag{9}$$

where α is a hyperparameter controlling the strength of absolute reward modulation. The first term preserves intra-group relative ranking based on the total reward R_i, while the second term $\tilde{r}_i^\alpha$ redistributes gradient budget across groups according to absolute accuracy. The mechanism operates at the group level: in a group where at least one sample achieves high $\tilde{r}_i$ (near-miss), the group's gradient contribution is largely preserved; in a uniformly-poor group where all $\tilde{r}_i \approx 0$, the entire group's gradient is compressed to near zero. Since standard GRPO produces equal-magnitude gradients from both types of groups, this redistribution substantially increases the near-miss samples' share of total training gradient. Importantly, within any given group, the relative ranking is preserved (Proposition A.1): when $\tilde{r}_i$ is close to 1, A'_i behaves similarly to standard GRPO; when $\tilde{r}_i$ is small, the

advantage magnitude is scaled down proportionally, preventing the policy from reinforcing the "best of the worst" in noise-dominated groups.

In our implementation, advantages are computed at the token level, with symmetric clipping applied to maintain numerical stability. Substituting the modulated advantage into the policy objective yields the following clipped surrogate loss:

$$\mathcal{L}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Bigg(\rho_i\left(\frac{R_i-\bar{R}}{\sigma_R+\epsilon}\tilde{r}_i^{\alpha}\right), \text{clip}(\rho_i, 1-\epsilon, 1+\epsilon)\left(\frac{R_i-\bar{R}}{\sigma_R+\epsilon}\tilde{r}_i^{\alpha}\right)\Bigg)\Bigg] - \beta_{\text{KL}}\, D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}), \tag{10}$$

where $\rho_i = \frac{\pi_\theta(o_i|x,q)}{\pi_{\text{old}}(o_i|x,q)}$ denotes the probability ratio between the current and sampling policies. By incorporating absolute quality information from SNRA into the advantage computation, AP-GRPO redistributes gradient budget across groups: updates from uniformly-poor groups are compressed to near zero, so that groups containing genuinely accurate near-miss predictions dominate the policy update. This recovers near-miss samples' effective training influence — not by amplifying their individual advantage, but by eliminating the competing noise from groups that carry no useful precision signal — yielding strong data efficiency and superior metric accuracy on 3D perception tasks.

3.4.1 Design Justification

We justify three key design choices (full analysis in Appendix A–B): (1) AP-GRPO redistributes gradient budget from uniformly-poor groups to informative ones (Appendix B.2); (2) multiplicative fusion is the axiomatically unique modulation preserving advantage sign and gradient decoupling (Appendix B.3, Theorem B.3); and (3) SNRA's sigmoid decay provides a natural non-vanishing reward floor for near-miss samples without additional clipping hyperparameters (Proposition B.5).

4 Experiments

4.1 Experimental Setup

We train on Numerical3D-50K using Qwen2.5-VL-7B as the primary backbone with G=16 rollouts per prompt, α=1.0, and sigmoid sharpness scheduling (k_{min}=1, k_{max}=100). Full hyperparameters and hardware details are in Appendix D.1.

Overall Performance. As shown in Table 1a and Table 1b, our method exhibits competitive performance across multiple spatial benchmarks compared to leading general-purpose models such as Qwen2.5-VL and LLaVA-OneVision. These results suggest that incorporating smooth, verifiable numerical rewards can provide more precise guidance for spatial perception than standard vision-language objectives. Qualitative visualizations and example predictions are provided in Appendix G.

Effectiveness of Sparse Numerical Supervision. While VST (Yang et al., 2025b) requires 4.5M supervised spatial samples, SmoothOp achieves comparable results—and slight improvements on CV-Bench (86.4% vs. 85.5%)—using only **50K** targeted samples, suggesting that SNRA and AP-GRPO effectively extract geometric information from numerical rewards without exhaustive supervised pre-training. We note that VST-RL (Yang et al., 2025b) further improves upon VST-SFT but uses a multi-stage regime (4.1M SFT + 135K RL), making it not directly comparable to our single-stage 50K setting.

Generalization to 3D Tasks. Results on MindCube and BLINK suggest that late-stage numerical alignment via reward hardening improves broader 3D reasoning, boosting occluded reconstruction and perspective-taking. This indicates that precise numerical supervision promotes more robust 3D geometric representations beyond the training objective.

4.2 Roadmap of Reward and Architecture Synergy

Starting from the SFT baseline (S_0, 49.7%), Table 2 and Fig. 4 trace the framework's evolution through three stages. **Stage 1** ($S_1 \rightarrow S_2$): plugging SNRA into standard GRPO actually *hurts* (54.4 $\rightarrow$ 53.8%), because group-relative normalization washes out absolute numerical differences, producing the highest Adv. Var. (0.165). **Stage 2** ($S_2 \rightarrow S_3$): introducing AP-GRPO restores the absolute signal, lifting accuracy to 55.4% and accelerating convergence by 2.6×, confirming that smooth rewards require absolute-preserving optimization. **Stage 3** ($S_3 \rightarrow S_5$): sigmoid scheduling breaks the late-stage plateau, reaching 60.0% with the lowest variance (0.138) by transitioning from coarse exploration to millimeter-level alignment. The sigmoid variant outperforms linear (S_4, 58.9%) by concentrating reward hardening into a narrow phase transition, avoiding premature precision pressure.

Table 1: Comprehensive Evaluation of Spatial Intelligence. Table (a) evaluates fine-grained video-based spatial reasoning on VSI-Bench, while Table (b) summarizes performance across diverse 3D spatial reasoning and general multimodal benchmarks. In the Training Phases column, S denotes Supervised Fine-Tuning (SFT), and R denotes Reinforcement Learning (RL).

(a) Evaluation Results on VSI-Bench with Data Scale and Training Phases

Methods	Data Scale	Train. Phase	Avg.	Numerical Question				Multiple-Choice Question			
				Obj. Cnt.	Abs. Dist.	Obj. Size	Room Size	Rel. Dist.	Rel. Dir.	Route Plan	Appr. Ord.
Proprietary and Open-source General Models											
GPT-4o (Hurst et al., 2024)	-	-	34.0	46.2	5.3	43.8	38.2	37.0	41.3	31.5	28.5
Gemini-1.5 Pro (Team et al., 2024)	-	-	45.4	56.2	30.9	64.1	43.6	51.3	46.3	36.0	34.6
LLaVA-OV-7B (?)	-	-	32.4	47.7	20.2	47.4	12.3	42.5	35.2	29.4	24.4
LLaVA-Video-7B (?)	-	-	35.6	48.5	14.0	47.8	24.2	43.5	42.4	34.0	30.6
Qwen2.5-VL-7B (Wang et al., 2024)	-	-	32.7	34.5	19.4	47.6	40.8	32.8	24.5	32.5	29.4
*Open-source Models with **Standard Architecture** Finetuning*											
SAT-7B (Ray et al., 2024)	175K	1 (S)	-	-	-	-	-	47.3	41.1	37.1	36.1
SpatialLadder-3B (Li et al., 2025b)	26K	3 (S+R)	45.7	-	-	-	-	-	-	-	-
InternVL-Sp.-8B (Deng et al., 2025)	291K	1 (S)	52.3	68.7	40.9	63.1	54.3	47.7	-	29.9	60.5
SpaceR-7B (Ouyang et al., 2025)	151K	2 (S+R)	43.5	61.9	28.6	60.9	35.2	38.2	46.0	31.4	45.6
ViLaSR-7B (Wu et al., 2025)	81K	2 (S+R)	45.4	63.5	34.4	60.6	30.9	48.9	45.2	30.4	49.2
VST-3B-SFT (Yang et al., 2025b)	4.1M	1 (S)	57.9	69.3	45.4	71.8	62.4	59.0	46.0	38.7	**70.2**
VST-7B-SFT (Yang et al., 2025b)	4.1M	1 (S)	**60.6**	**72.0**	**44.4**	**74.3**	68.3	59.7	**55.8**	**44.9**	65.2
SmoothOp-3B	**50K**	1 (R)	53.9	70.1	42.8	69.9	38.3	60.4	45.4	38.7	65.5
SmoothOp-7B	**50K**	1 (R)	60.0	71.1	41.4	72.2	**70.3**	**61.7**	55.1	42.8	65.4

(b) Evaluation Results on Other Spatial Intelligence Benchmarks.

Model	Data Scale	Train. Phase	CV-Bench	MMSI	MindCube	BLINK
Proprietary Models and Open-source General Models						
Gemini-2.5-Pro (Comanici et al., 2025)	–	–	53.5	38.0	57.6	70.6
GPT-4o (Hurst et al., 2024)	–	–	76.0	30.3	53.4	65.9
LLaVA-OV-7B (?)	–	–	61.9	31.0	34.7	48.2
Qwen2.5-VL-3B (Wang et al., 2024)	–	–	71.4	28.6	37.6	47.6
Qwen2.5-VL-7B (Wang et al., 2024)	–	–	75.4	26.8	36.0	56.4
InternVL3-8B (Chen et al., 2024b)	–	–	42.1	28.0	41.5	55.5
*Open-source Models with **Standard Architecture** Finetuning*						
SpatialLadder-3B (Li et al., 2025b)	26K	3 (S+R)	73.7	26.1	**43.4**	52.9
SPAR-8B (Zhang et al., 2025a)	7M	2 (S)	80.7	–	–	43.9
SpaceR-7B (Ouyang et al., 2025)	151K	2 (S+R)	74.8	20.1	37.9	55.4
ViLaSR-7B (Wu et al., 2025)	81K	2 (S+R)	81.8	30.2	35.1	56.2
VST-3B-SFT (Yang et al., 2025b)	4.1M	1 (S)	84.4	30.2	35.9	59.1
VST-7B-SFT (Yang et al., 2025b)	4.1M	1 (S)	85.5	32.0	39.7	62.1
SmoothOp-3B	**50K**	1 (R)	**84.9**+18.9	**32.2**+12.6	39.4+4.8	**56.9**+19.5
SmoothOp-7B	**50K**	1 (R)	**86.4**+14.6	**33.7**+25.7	39.6+10.0	**62.6**+11.0

Table 2: **Roadmap of optimization mechanisms on VSI-Bench (Qwen2.5-VL-7B).** Conv. Steps = iterations to reach 95% of max accuracy; Adv. Var. = policy gradient stability.

ID	Optimization Mechanism	Convergence Steps (↓)	Advantage Variance (↓)	Average Accuracy (%)
S_0	SFT	N/A	N/A	49.7
S_1	GRPO with Binary 0-1 Reward	310	0.152	54.4
S_2	GRPO with SNRA (Fixed k)	390	0.165	53.8
S_3	**AP-GRPO** with SNRA (Fixed k)	250	0.148	55.4
S_4	**AP-GRPO** with SNRA (Linear Scheduling)	150	0.142	58.9
S_5	**AP-GRPO** with **SNRA (Sigmoid Scheduling)**	**130**	**0.138**	**60.0**

4.3 Ablation Studies

We conduct ablations on the VSI-Bench validation set. Additional ablation studies, including hyperparameter sensitivity analyses for k_{min}, k_{max}, α, and group size G, are reported in Appendix E.

Data Efficiency. As shown in Table 4, AP-GRPO at 10k samples (51.4%) already approaches Standard GRPO at 50k (54.4%), demonstrating

Table 3: (a) Comparison with RL baselines on VSI-Bench (**Qwen2.5-VL-7B**). All methods use the same backbone, data, verifier, and budget. (b) Comparison of advantage estimation strategies.

Method	Avg. Acc (%)
Reinforce++	51.95
DAPO	53.47
Standard GRPO	54.40
ReMax	54.36
GSPO	55.41
Dr. GRPO	56.12
CISPO	57.58
AP-GRPO (Ours)	**59.97**

(a) RL baseline comparison.

Advantage Formulation	Sync. Type	Avg. Acc (%)
Standard GRPO	Relative only	54.4
Pure Absolute Scalar	Absolute only	43.1
AP-GRPO (Ours)	**Rel. + Abs.**	**60.0**

(b) Advantage estimation strategies.

Table 4: Data efficiency and scaling comparison on VSI-Bench.

Method	Acc. (%) @ Training Samples		
	10k	20k	50k
Standard GRPO	44.9	48.6	54.4
AP-GRPO (Ours)	51.4	57.6	**60.0**

that gradient budget redistribution substantially improves sample utilization by recovering near-miss training signals that would otherwise be wasted.

RL Baseline Comparison. Under matched settings (same backbone, data, verifier, and budget), AP-GRPO (59.97%) outperforms all recent RL/RLVR baselines (Table 7), including Dr. GRPO (56.12%) and CISPO (57.58%). Methods like Reinforce++ and DAPO underperform standard GRPO because their designs target binary-reward settings: Reinforce++'s global batch baseline (Hu et al., 2025) cannot adapt to the prompt-wise reward scale variation inherent in continuous numerical tasks, while DAPO's Clip-Higher exploration (Yu et al., 2025) amplifies noise under dense graded rewards.

Cross-Architecture Generalization. To verify that the proposed method is not specific to one architecture, we evaluate SmoothOp on four VLM backbones under the same protocol (Table 5). SmoothOp consistently improves spatial reasoning: Qwen2.5-VL-3B (+21.2), Qwen2.5-VL-7B (+27.3), LLaVA-OneVision-7B (+14.1), and Qwen3-VL-8B (+3.9). The modest gain on Qwen3-VL is expected as it already incorporates RL-based optimization during pre-training.

Multiplicative vs. Alternative Fusion. The multiplicative formulation (59.97%) significantly outperforms additive (56.63%) and mixed (55.44%) alternatives, consistent with the theoretical analysis in Appendix B.3 showing that multiplicative fusion uniquely preserves advantage sign and gradient decoupling (Table 8, Appendix D).

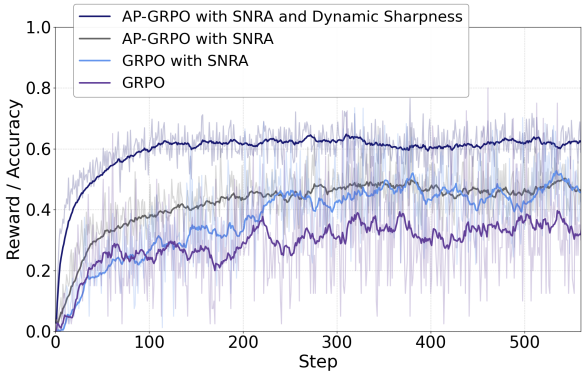


Figure 4: Component-level ablation study. The synergy of SNRA and Dynamic Sharpness allows AP-GRPO to surpass standard GRPO variants, validating the necessity of geometric priors and reward landscape refinement for precise 3D reasoning.

Table 5: Cross-architecture generalization on VSI-Bench. All models trained with Numerical3D-50K and AP-GRPO.

Backbone	Base	+ SmoothOp	Δ
Qwen2.5-VL-3B	32.7	53.9	+21.2
Qwen2.5-VL-7B	32.7	60.0	+27.3
LLaVA-OneVision-7B	32.4	46.5	+14.1
Qwen3-VL-8B-Instruct	57.9	61.8	+3.9

Advantage Formulation and Data Efficiency. Tables 3b and 4 present two additional ablations. Table 3b compares advantage estimation strategies: Pure Absolute Scalar (43.1%, absolute only) fails catastrophically without relative ranking, while our hybrid AP-GRPO (60.0%, Rel. + Abs.) significantly outperforms both extremes. Table 4 shows that AP-GRPO achieves 51.4% with only 10K samples, already exceeding standard GRPO at 50K (54.4% vs. 48.6%), directly demonstrating the data efficiency of gradient budget redistribution.

5 Conclusion

We present AP-GRPO, a data-efficient RL framework that addresses the near-miss gradient waste problem in 3D spatial reasoning. By redistributing gradient budget so that near-miss predictions dominate the policy update, our approach achieves competitive spatial reasoning performance with only 50K targeted samples, generalizing consistently across four VLM backbones without task-specific architectural changes.

Limitations

Our study has several limitations that we plan to address in future work.

Scope of evaluation. We evaluate on four VLM backbones (Qwen2.5-VL-3B/7B, LLaVA-OneVision-7B, Qwen3-VL-8B) and five spatial reasoning benchmarks. While cross-architecture generalization is encouraging (Table 5), evaluation on larger models (>13B) and non-spatial continuous tasks (e.g., scientific measurement, financial forecasting) remains for future work. Additionally, we have not yet measured the impact on general-purpose VLM capabilities (e.g., MMMU, MM-Bench); verifying that spatial RL training does not degrade general performance is important before deployment.

Verifiable reward requirement. AP-GRPO and SNRA rely on the availability of deterministic verifiable signals with known ground truth. This limits direct applicability to tasks where such signals are unavailable (e.g., aesthetic judgment, open-ended generation). Extending to softer verification signals or learned reward models is a promising direction.

Hyperparameter sensitivity. While the framework is robust to k_{min} within $(0, 1]$ and $\alpha = 1$ is clearly optimal (Appendix E), the sharpness schedule parameters (k_{max}, transition midpoint τ, steepness s) have not been extensively tuned across diverse task domains. We provide sensitivity analysis for the key hyperparameters but acknowledge that optimal settings may vary for non-3D applications.

Computational cost. Our method requires generating G samples per prompt for group-relative optimization. With $G = 16$ (our default), this increases sampling cost by $16\times$ compared to standard SFT. While the overall training remains data-efficient (50K samples), the per-step compute overhead may be a concern for resource-constrained settings.

References

Adel Ahmadyan, Liangzhe Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. 2021. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*.

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

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. 2025. Spatialbot: Precise spatial understanding with vision language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9490–9498. IEEE.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *International Conference on 3D Vision (3DV)*.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. 2024a. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14455–14465.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

An-Chieh Cheng, Hongxu Yin, Yang Fu, Qiushan Guo, Ruihan Yang, Jan Kautz, Xiaolong Wang, and Sifei Liu. 2024. Spatialrgpt: Grounded spatial reasoning in vision-language models. *Advances in Neural Information Processing Systems*, 37:135062–135093.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and 1 others. 2025. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7395–7408.

Gerard Debreu. 1960. Topological methods in cardinal utility theory. In Kenneth J. Arrow, Samuel Karlin, and Patrick Suppes, editors, *Mathematical Methods in the Social Sciences*, pages 16–26. Stanford University Press.

Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and 1 others. 2021. Arkitscenes: A diverse real-world dataset for

3d indoor scene understanding using mobile rgb-d data. *NeurIPS Datasets and Benchmarks*, 2(6):16.

Nianchen Deng, Lixin Gu, Shenglong Ye, Yinan He, Xizhou Zhu, and Jifeng Dai. 2025. Internspatial: A comprehensive dataset for spatial reasoning in vision-language models. *arXiv preprint arXiv:2506.18385*.

Zhiwen Fan, Jian Zhang, Renjie Li, Junge Zhang, Runjin Chen, Hezhen Hu, Kevin Wang, Huaizhi Qu, Dilin Wang, Zhicheng Yan, and 1 others. 2025. Vlm-3r: Vision-language models augmented with instruction-aligned 3d reconstruction. *arXiv preprint arXiv:2505.20279*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Huy Ha and Shuran Song. 2022. Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. *arXiv preprint arXiv:2207.11514*.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*.

Ting Huang, Zeyu Zhang, and Hao Tang. 2025. 3d-r1: Enhancing reasoning in 3d vlms for unified scene understanding. *arXiv preprint arXiv:2507.23478*.

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

Baoxiong Jia, Yixin Chen, Huangyue Yu, Yan Wang, Xuesong Niu, Tengyu Liu, Qing Li, and Siyuan Huang. 2024. Sceneverse: Scaling 3d vision-language learning for grounded scene understanding. In *European Conference on Computer Vision*, pages 289–310. Springer.

Chengzu Li, Wenshan Wu, Huanyu Zhang, Yan Xia, Shaoguang Mao, Li Dong, Ivan Vulić, and Furu Wei. 2025a. Imagine while reasoning in space: Multimodal visualization-of-thought. *arXiv preprint arXiv:2501.07542*.

Hongxing Li, Dingming Li, Zixuan Wang, Yuchen Yan, Hang Wu, Wenqi Zhang, Yongliang Shen, Weiming Lu, Jun Xiao, and Yueting Zhuang. 2025b. Spatialladder: Progressive training for spatial reasoning in vision-language models. *arXiv preprint arXiv:2510.08531*.

Zongxia Li, Wenhao Yu, Chengsong Huang, Zhenwen Liang, Rui Liu, Fuxiao Liu, Jingxi Chen, Dian Yu, Jordan Boyd-Graber, Haitao Mi, and Dong Yu. 2025c. Self-rewarding vision-language model via reasoning decomposition. *arXiv preprint arXiv:2508.19652*.

Kangcheng Liu, Yong-Jin Liu, and Baoquan Chen. 2025a. General 3d vision-language model with fast rendering and pre-training vision-language alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025b. Spatial-ssrl: Enhancing spatial understanding via self-supervised reinforcement learning. *arXiv preprint arXiv:2510.27606*.

Wufei Ma, Yu-Cheng Chou, Qihao Liu, Xingrui Wang, Celso de Melo, Jianwen Xie, and Alan Yuille. 2025. Spatialreasoner: Towards explicit and generalizable 3d spatial reasoning. *arXiv preprint arXiv:2504.20024*.

Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. 2025. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. 2025. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning. *arXiv preprint arXiv:2502.19634*.

Zhenyu Pan and Han Liu. 2025. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. *arXiv preprint arXiv:2503.18470*.

Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, and 1 others. 2025a. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. *arXiv preprint arXiv:2505.15298*.

Kangan Qian, Ziang Luo, Sicong Jiang, Zilin Huang, Jinyu Miao, Zhikun Ma, Tianze Zhu, Jiayin Li, Yangfan He, Zheng Fu, and 1 others. 2025b. Fasionad++: Integrating high-level instruction and information bottleneck in fat-slow fusion systems for enhanced safety in autonomous driving with adaptive feedback. *arXiv preprint arXiv:2503.08162*.

Kangan Qian, Zhikun Ma, Yangfan He, Ziang Luo, Tianyu Shi, Tianze Zhu, Jiayin Li, Jianhui Wang, Ziyu Chen, Xiao He, and 1 others. 2024. Fasionad: Fast and slow fusion thinking systems for human-like autonomous driving with adaptive feedback. *arXiv preprint arXiv:2411.18013*.

Kangan Qian, Jinyu Miao, Xinyu Jiao, Ziang Luo, Zheng Fu, Yining Shi, Yunlong Wang, Kun Jiang, and Diange Yang. 2025c. Priormotion: Generative class-agnostic motion prediction with raster-vector motion field priors. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27284–27294.

Kangan Qian, Jinyu Miao, Ziang Luo, Zheng Fu, Yining Shi, Yunlong Wang, Kun Jiang, Mengmeng Yang, Diange Yang, and 1 others. 2025d. Lego-motion: Learning-enhanced grids with occupancy instance modeling for class-agnostic motion prediction. *arXiv preprint arXiv:2503.07367*.

Arijit Ray, Jiafei Duan, Ellis Brown, Reuben Tan, Dina Bashkirova, Rose Hendrix, Kiana Ehsani, Aniruddha Kembhavi, Bryan A Plummer, Ranjay Krishna, and 1 others. 2024. Sat: Dynamic spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*.

Yirong Sun and Yanjun Chen. 2025. Prism: Preference refinement via implicit scene modeling for 3d vision-language preference-based reinforcement learning. *arXiv preprint arXiv:2503.10177*.

Reuben Tan, Baolin Peng, Zhengyuan Yang, Hao Cheng, Oier Mees, Theodore Zhao, Andrea Tupini, Isar Meijier, Qianhui Wu, Yuncong Yang, Lars Liden, Yu Gu, Sheng Zhang, Xiaodong Liu, Lijuan Wang, Marc Pollefeys, Yong Jae Lee, and Jianfeng Gao. 2025. Multimodal reinforcement learning with adaptive verifier for ai agents. *arXiv preprint arXiv:2512.03438*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, Hisham Cholakkal, Ivan Laptev, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*.

Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Niessner. 2019. Rio: 3d object instance re-localization in changing indoor environments. In *ICCV*.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. 2025. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Junfei Wu, Jian Guan, Kaituo Feng, Qiang Liu, Shu Wu, Liang Wang, Wei Wu, and Tieniu Tan. 2025. Reinforcing spatial reasoning in vision-language models with interwoven thinking and visual drawing. *arXiv preprint arXiv:2506.09965*.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. 2025. Perception-r1: Advancing multimodal reasoning capabilities of mllms via visual perception reward. *arXiv preprint arXiv:2506.07218*.

Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025a. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 10632–10643.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911.

Rui Yang, Ziyu Zhu, Yanwei Li, Jingjia Huang, Shen Yan, Siyuan Zhou, Zhe Liu, Xiangtai Li, Shuangye Li, Wenqian Wang, and 1 others. 2025b. Visual spatial tuning. *arXiv preprint arXiv:2511.05491*.

Shusheng Yang, Jihan Yang, Pinzhi Huang, Ellis Brown, Zihao Yang, Yue Yu, Shengbang Tong, and 1 others. 2025c. Cambrian-s: Towards spatial supersensing in video. *arXiv preprint arXiv:2511.04670*.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, and 1 others. 2025d. Mmsi-bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*.

Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. 2023. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12–22.

Baiqiao Yin, Qineng Wang, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, and 1 others. 2025. Spatial mental modeling from limited views. In *Structural Priors for Vision Workshop at ICCV'25*.

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

Jiahui Zhang, Yurui Chen, Yanpeng Zhou, Yueming Xu, Ze Huang, Jilin Mei, Junhui Chen, Yu-Jie Yuan, Xinyue Cai, Guowei Huang, Xingyue Quan, Hang Xu, and Li Zhang. 2025a. From flatland to space: Teaching vision-language models to perceive and reason in 3d. *arXiv preprint arXiv:2503.22976*.

Xiangxiang Zhang, Jingxuan Wei, Donghong Zhong, Qi Chen, Caijun Jia, Cheng Tan, Jinming Gu, Xiaobo Qin, Zhiping Liu, Liang Hu, Tong Sun, Yuchen Wu, Zewei Sun, Chenwei Lou, Hua Zheng, Tianyang Zhan, Changbao Wang, Shuangzhi Wu, Zefa Lin, and 15 others. 2025b. Structvrm: Aligning multimodal reasoning with structured and verifiable reward models. *arXiv preprint arXiv:2508.05383*.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. 2025c. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*.

Chenming Zhu, Tai Wang, Wenwei Zhang, Jiangmiao Pang, and Xihui Liu. 2025. Llava-3d: A simple yet effective pathway to empowering lmms with 3d capabilities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4295–4305.

Appendix

A Design Justification

This section and the following §B together provide a comprehensive justification for the AP-GRPO design. We first analyze the near-miss waste problem (§A.1) and then present the full theoretical foundations: gradient budget redistribution via three-scenario analysis (§B.2), a comparative analysis showing why multiplicative fusion is preferred (§B.3), its axiomatic uniqueness proof (§B.4), a formal demonstration that additive alternatives introduce sign-flipping pathology (§B.5), and a characterization of SNRA's non-vanishing reward floor (§B.6).

A.1 The Near-Miss Waste Problem

We provide a detailed three-stage analysis of how standard GRPO fails to utilize near-miss samples in continuous numerical tasks, and how AP-GRPO recovers their training influence.

Stage 1: Binary rewards discard error-magnitude information. Standard 0-1 rewards assign identical zero reward to both near-miss predictions (e.g., predicting 2.1m when the answer is 2.0m) and completely wrong predictions (e.g., predicting 50m). This "reward annihilation" deprives the policy of incremental feedback during early training.

Stage 2: Dense rewards alone do not help under GRPO. A natural fix is to use smooth, graded rewards — which is exactly what our SNRA operator provides. However, simply plugging dense rewards into standard GRPO does not improve performance (Table 2: S_1 54.4 $\rightarrow$ S_2 53.8). The reason is that GRPO computes advantages via group-relative normalization $A_i = (R_i - \bar{R})/\sigma_R$. When all samples in a group are poor — common in early 3D training — the group mean is low and the standard deviation is small, so the normalization inflates tiny absolute differences into artificially large advantages. The policy then chases noisy relative rankings within uniformly poor groups, wasting gradient budget on "the best of the worst."

Stage 3: AP-GRPO recovers near-miss training influence via gradient budget redistribution. AP-GRPO multiplies the relative advantage by $\tilde{r}_i^{\alpha}$. This does *not* amplify individual low-reward samples — indeed, each sample's advantage magnitude decreases slightly. The key mechanism operates at the *group level*: in a uniformly-poor group (all $\tilde{r}_i \approx 0$), the entire group's gradient contribution is compressed to near zero; in a group containing near-miss samples (some $\tilde{r}_i$ substantially above zero), gradients are preserved. Since standard GRPO produces equal-magnitude gradients from both types of groups, AP-GRPO effectively redirects gradient budget from noise-dominated groups to informative groups, substantially increasing the near-miss samples' share of total training gradient. In groups with mixed quality, the ordering is preserved (Proposition B.1), so the standard GRPO benefit is retained.

Worked Example. Consider two groups generated from the same prompt (GT = 2.0m). *Group A* (informative): predictions 2.1m, 3.0m, 5.0m, 10.0m — the 2.1m sample is a near-miss with SNRA reward $\tilde{r} = 0.95$. *Group B* (uniformly poor): predictions 10m, 15m, 20m, 50m — all far from target, all $\tilde{r} \approx 0$. Under standard GRPO, both groups produce *equal-magnitude* gradient updates ($\Sigma|A| \approx 3.8$), because normalization operates within each group independently. Under AP-GRPO, Group B's gradients are compressed by $\sim$3000$\times$ (since all $\tilde{r}_i^\alpha \approx 0$), while Group A's near-miss sample retains 95% of its advantage. The near-miss sample's share of total training gradient rises from $\sim$21% to $\sim$83%. See Section B.2 for the full numerical derivation.

B Theoretical Analysis of AP-GRPO

We provide rigorous theoretical analysis of AP-GRPO's properties: gradient directionality preservation, gradient budget redistribution quantification, variance suppression, and the axiomatic uniqueness of multiplicative fusion.

Assumptions

B.1 Gradient Directionality and Policy Reweighting

A key concern when modifying the advantage function in reinforcement learning is whether the resulting update still preserves the qualitative directionality of policy improvement. Although AP-GRPO introduces a non-uniform weighting by the absolute reward term r^α, we show that it maintains the sign structure of the group-relative advantage and hence preserves the local preference ordering of trajectories within a group.

B.1.1 Setup

Let the standard RL objective be

$$\mathcal{J}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R(\tau)],$$

and denote by $\phi(\tau) = \nabla_\theta \log \pi_\theta(\tau)$ the score function.

Let $A_{GRPO}(\tau)$ be the group-normalized advantage used in GRPO, constructed from reward $r(\tau)$ within a group $\mathcal{G}$ as

$$A_{GRPO}(\tau) = \frac{r(\tau) - \bar{r}_\mathcal{G}}{\sigma_{r,\mathcal{G}} + \epsilon},$$

so that within each group we approximately have $\mathbb{E}_\mathcal{G}[A_{GRPO}] \approx 0$ and $\mathbb{E}_\mathcal{G}[A_{GRPO}^2] \approx 1$.

The GRPO update direction (ignoring clipping and other regularization) can be written as

$$g_{GRPO} \;=\; \mathbb{E}_{\tau\sim\pi_\theta}\big[\phi(\tau)\, A_{GRPO}(\tau)\big].$$

AP-GRPO introduces an absolute reward weighting

$$w(r) = r^\alpha, \quad r \in [0,1],\ \alpha \geq 1,$$

and uses the modified gradient

$$g_{AP} \;=\; \mathbb{E}_{\tau\sim\pi_\theta}\big[\phi(\tau)\, A_{GRPO}(\tau)\, w(r(\tau))\big].$$

B.1.2 Monotone Reweighting and Local Ordering Preservation

We first clarify what AP-GRPO can be guaranteed to preserve.

Proposition B.1 (Sign preservation and positive ordering)**.** *Let $\mathcal{G}$ be a GRPO group, and for $\tau \in \mathcal{G}$ define*

$$A(\tau) = A_{GRPO}(\tau), \qquad \tilde{A}(\tau) = A_{GRPO}(\tau)\, r(\tau)^\alpha.$$

Assume that reward $r(\tau)$ is monotone in the original advantage, i.e.

$$A(\tau_1) > A(\tau_2) \quad \Rightarrow \quad r(\tau_1) \geq r(\tau_2).$$

Then for any $\tau \in \mathcal{G}$:

1. $\operatorname{sign}\big(\tilde{A}(\tau)\big) = \operatorname{sign}\big(A(\tau)\big)$.
2. *If $A(\tau_1) > A(\tau_2) > 0$, then $\tilde{A}(\tau_1) \geq \tilde{A}(\tau_2) > 0$.*

In particular, AP-GRPO guarantees that trajectories with positive relative advantages maintain their local ranking, ensuring that the best-performing samples within a group receive the strongest positive reinforcement.

Proof. (1) For any τ, $r(\tau) \in [0,1]$ and $\alpha \geq 1$ imply $r(\tau)^\alpha \geq 0$. Therefore

$$\tilde{A}(\tau) = A(\tau)\, r(\tau)^\alpha$$

is obtained by multiplying $A(\tau)$ by a non-negative scalar. Hence $\operatorname{sign}(\tilde{A}(\tau)) = \operatorname{sign}(A(\tau))$.

(2) Consider the case $A(\tau_1) > A(\tau_2) > 0$. By the monotonicity assumption on $r(\cdot)$, we have $r(\tau_1) \geq r(\tau_2)$, and since $x \mapsto x^\alpha$ is non-decreasing on $[0,1]$, this implies

$$r(\tau_1)^\alpha \geq r(\tau_2)^\alpha.$$

Thus

$$\tilde{A}(\tau_1) = A(\tau_1)\, r(\tau_1)^\alpha$$

$$\geq A(\tau_1)\, r(\tau_2)^\alpha$$
$$\geq A(\tau_2)\, r(\tau_2)^\alpha = \tilde{A}(\tau_2),$$

where the last inequality uses $A(\tau_1) > A(\tau_2) > 0$ and $r(\tau_2)^\alpha \geq 0$. Therefore $\tilde{A}(\tau_1) \geq \tilde{A}(\tau_2) > 0$.

Hence, AP-GRPO applies a non-negative reweighting that strictly preserves the ordering of positive advantages. □

Within each GRPO group, the AP-GRPO update preserves the direction of relative policy improvement for high-performing trajectories. For trajectories with negative advantages ($A(\tau) < 0$), the weighting term $r(\tau)^\alpha$ scales down the penalty in proportion to absolute quality: as rewards decrease ($r \to 0$), the effective penalty $\tilde{A}(\tau)$ vanishes. The key mechanism operates at the *group level*: uniformly-poor groups see their entire gradient contribution compressed to near zero, while groups containing near-miss samples retain their contributions. This redistributes gradient budget from noise-dominated groups to informative groups, recovering near-miss samples' effective training influence.

B.2 Gradient Budget Redistribution: A Three-Scenario Static Analysis

We provide a concrete numerical analysis demonstrating how AP-GRPO recovers near-miss samples' training influence through group-level gradient budget redistribution. We fix SNRA sharpness $k = 1$ (early training), $\alpha = 1$, group size $G = 4$, and ground truth $\mathrm{GT} = 2.0\mathrm{m}$. For clarity of exposition, we use the absolute error $e_i = |\hat{y}_i - y|$ in this worked example; the mechanism is identical under the squared error $e_i = |\hat{y}_i - y|^2$ used in our implementation, differing only in the numerical scale of k. We compare three reward–optimization configurations across two types of groups.

B.2.1 Setup

Group A (Informative): Predictions $\{2.1, 3.0, 5.0, 10.0\}\mathrm{m}$. Sample 1 (2.1m) is a *near-miss* with error $e_1 = 0.1$.

Group B (Uniformly Poor): Predictions $\{10.0, 15.0, 20.0, 50.0\}\mathrm{m}$. All samples are far from the target.

Three scenarios:

1. **Binary + GRPO**: $R_i \in \{0, 1\}$ with tolerance 0.05m. All samples in both groups receive $R_i = 0$.
2. **SNRA + GRPO**: Dense reward $\tilde{r}_i = 2/(1 + \exp(k \cdot e_i))$, standard GRPO advantage $A_i = (R_i - \bar{R})/(\sigma_R + \epsilon)$.
3. **SNRA + AP-GRPO**: Same dense reward, but $A'_i = A_i \cdot \tilde{r}_i^\alpha$.

B.2.2 Group A: Informative Group

Under SNRA with $k = 1$, the rewards are:

Sample	Pred.	e_i	$\tilde{r}_i$	GRPO A_i	AP-GRPO A'_i	$\lvert A'_i\rvert/\lvert A_i\rvert$
S1 (near-miss)	2.1m	0.1	0.950	+1.53	+1.45	95.0%
S2	3.0m	1.0	0.538	+0.39	+0.21	53.8%
S3	5.0m	3.0	0.095	−0.83	−0.08	9.5%
S4	10.0m	8.0	0.001	−1.09	−0.001	0.1%
			Total gradient $\Sigma\lvert A\rvert$:	3.84	1.74	
			Near-miss share $\lvert A_1\rvert/\Sigma\lvert A\rvert$:	39.8%	**83.3%**	

Observation. Under AP-GRPO, the near-miss sample (S1) retains 95% of its original advantage magnitude, while far-miss samples (S3, S4) are effectively gated out. The near-miss's share of total gradient within this group rises from 39.8% to **83.3%**.

B.2.3 Group B: Uniformly-Poor Group

Sample	Pred.	e_i	$\tilde{r}_i$	GRPO A_i	AP-GRPO A'_i
S1	10.0m	8.0	6.7×10^{-4}	+1.72	$+1.2 \times 10^{-3}$
S2	15.0m	13.0	≈ 0	−0.56	≈ 0
S3	20.0m	18.0	≈ 0	−0.58	≈ 0
S4	50.0m	48.0	≈ 0	−0.58	≈ 0
			Total gradient $\Sigma\lvert A\rvert$:	3.44	≈ 0.001

Observation. Under standard GRPO, Group B produces gradient of magnitude 3.44 — nearly identical to Group A's 3.84. This means *half the total training gradient comes from a group that carries no useful precision signal*. Under AP-GRPO, Group B's gradient is compressed by $\sim$3000×.

B.2.4 Cross-Group Comparison and Near-Miss Signal-to-Noise Ratio

Metric	Binary+GRPO	SNRA+GRPO	SNRA+AP-GRPO
Near-miss reward	0	0.950	0.950
Near-miss $\lvert A\rvert$	0	1.53	1.45
Group A gradient $\Sigma\lvert A\rvert$	0	3.84	1.74
Group B gradient $\Sigma\lvert A\rvert$	0	3.44	≈ 0.001
Informative/Poor ratio	N/A	**1.12**×	**∼1500**×
Near-miss global SNR	0%	21.0%	**83.3%**

where the **near-miss global SNR** is defined as the near-miss sample's gradient magnitude divided by the total gradient from both groups: $\mathrm{SNR} = |A_1^{(A)}|/(\Sigma|A^{(A)}| + \Sigma|A^{(B)}|)$.

Key conclusions:

1. **Binary + GRPO**: All rewards are zero. All advantages are zero. Near-miss samples are *completely wasted*.

2. **SNRA + GRPO**: SNRA makes near-miss samples distinguishable ($\tilde{r}_1 = 0.95$), but GRPO normalization produces equal-magnitude gradients from both groups (ratio $1.12\times$). The near-miss signal is *drowned* by noise from Group B. This explains why SNRA + GRPO actually *hurts* performance (Table 2: S_1 54.4 $\rightarrow$ S_2 53.8).

3. **SNRA + AP-GRPO**: Group B's gradient is compressed by $\sim$3000$\times$. The near-miss sample's global SNR rises from 21% to 83%. Near-miss training influence is *recovered* — not by amplifying individual advantages (which decrease slightly: $1.53 \rightarrow 1.45$), but by eliminating competing noise from uniformly-poor groups.

This analysis resolves the apparent paradox that multiplying by $\tilde{r}_i^\alpha < 1$ "suppresses" near-miss samples: the suppression is real at the *individual sample level*, but the recovery is achieved at the *group level* through gradient budget redistribution.

B.3 Why Multiplicative Fusion: A Comparative Analysis

A natural question is why we choose multiplicative fusion $A'_i = A_i \cdot \tilde{r}_i^\alpha$ over alternative combination strategies. We analyze three candidate formulations and show that the multiplicative form is uniquely suited for our goal.

B.3.1 Three Candidate Formulations

1. **Multiplicative (AP-GRPO):** $A'_i = A_i \cdot \tilde{r}_i^\alpha$

2. **Additive:** $A'_i = A_i + \beta \cdot \tilde{r}_i^\alpha$, where $\beta > 0$ is a scaling coefficient

3. **Mixed (Multiplicative + Additive):** $A'_i = A_i \cdot \tilde{r}_i^\alpha + \beta \cdot \tilde{r}_i^\alpha$

B.3.2 Desirable Properties

We consider four properties that a well-designed advantage modulation should satisfy:

P1. Sign Preservation. $\text{sign}(A'_i) = \text{sign}(A_i)$ for all $\tilde{r}_i \in (0, 1]$. The modulation should not flip the direction of the policy update: samples that GRPO ranks as below-average should still be penalized, not reinforced.

P2. Zero-Mean Approximate Invariance. If $\sum_i A_i = 0$ (as guaranteed by GRPO normalization), then A'_i should remain approximately balanced, i.e., $|\sum_i A'_i|$ should be small. This prevents the modulation from introducing a systematic bias toward increasing or decreasing the policy's entropy.

P3. Monotone Absolute Scaling. $|A'_i|$ should be monotonically increasing in $\tilde{r}_i$ for fixed $|A_i|$. Higher absolute reward quality should translate to larger update magnitude.

P4. Gradient Decoupling. The ranking direction (determined by GRPO's relative ordering) and the precision magnitude (determined by $\tilde{r}_i^\alpha$) should not interfere: modulation should scale the magnitude without altering the sign or relative ordering of positive advantages.

B.3.3 Analysis

Multiplicative fusion satisfies all four properties.

- **P1** (✓): Since $\tilde{r}_i^\alpha \geq 0$, the sign of $A'_i = A_i \cdot \tilde{r}_i^\alpha$ equals the sign of A_i.

- **P2** (✓): $\sum_i A'_i = \sum_i A_i \cdot \tilde{r}_i^\alpha$. When rewards within a group are similar (common within a group), this is approximately $\bar{r}^\alpha \cdot \sum_i A_i \approx 0$.

- **P3** (✓): $|A'_i| = |A_i| \cdot \tilde{r}_i^\alpha$, which is monotonically increasing in $\tilde{r}_i$ since x^α is monotone for $\alpha \geq 1$.

- **P4** (✓): The multiplicative structure ensures that the sign structure of GRPO is untouched, and the magnitude scaling depends only on the absolute reward. These two contributions are decoupled.

Additive fusion violates P1 (sign preservation).

- Consider a sample with $A_i = -0.5$ (below group average) and $\tilde{r}_i = 0.8$ (decent absolute quality). With $\beta = 1$: $A'_i = -0.5 + 1 \times 0.8 = +0.3$ The advantage *flips sign*: the policy now reinforces a sample that GRPO correctly identified as below-average. This causes the policy to learn from conflicting signals — GRPO's relative ranking says "penalize," but the additive term says "reinforce."

- **P2** ($\times$): $\sum_i A'_i = \sum_i A_i + \beta \sum_i \tilde{r}_i^\alpha = 0 + \beta \sum_i \tilde{r}_i^\alpha > 0$. The additive term introduces a systematic positive bias, pushing the policy toward higher entropy regardless of quality.

Mixed fusion inherits the additive defect.

- $A'_i = A_i \cdot \tilde{r}_i^\alpha + \beta \cdot \tilde{r}_i^\alpha = (A_i + \beta) \cdot \tilde{r}_i^\alpha$. This is equivalent to multiplicative fusion with a shifted advantage $A_i + \beta$. The shift can flip signs when $A_i \in (-\beta, 0)$, violating P1.

- The additional β parameter introduces unnecessary complexity without clear benefit, since the multiplicative term alone already provides the desired properties.

B.3.4 Empirical Validation

The theoretical analysis is confirmed by experiments on VSI-Bench (Qwen2.5-VL-7B, 50K data):

Formulation	P1	P2	P4	VSI-Bench (%)
Multiplicative (AP-GRPO)	✓	✓	✓	**59.97**
Additive	×	×	×	56.63
Mixed	×	×	partial	55.44

The multiplicative formulation outperforms both alternatives by a significant margin (+3.34 over additive, +4.53 over mixed), consistent with the theoretical analysis. The additive and mixed forms' sign-flipping behavior leads to conflicting gradient signals that destabilize training, explaining their inferior performance.

We now formalize this result: Theorem B.3 below proves that multiplicative fusion is the unique linear advantage modulation satisfying a natural set of axioms, followed by a formal proof that additive alternatives introduce sign-flipping pathology (Theorem B.4), and a characterization of SNRA's non-vanishing reward floor (Proposition B.5).

B.4 Axiomatic Uniqueness of Multiplicative Fusion

We establish that multiplicative fusion is the unique linear modulation satisfying a set of natural axioms for advantage modification.

Definition B.2 (Linear Advantage Modulation)**.** A *linear advantage modulation* is a function $F : \mathbb{R} \times [0, 1] \to \mathbb{R}$ of the form $F(A, r) = g(r) \cdot A + h(r)$, where $g, h : [0, 1] \to \mathbb{R}$ are continuous functions.

We require F to satisfy the following axioms:

A1 Sign Preservation: $\text{sign}(F(A, r)) = \text{sign}(A)$ for all $A \neq 0$, $r \in (0, 1]$.

A2 Zero-Mean Invariance: If $\sum_i A_i = 0$, then $\sum_i F(A_i, r_i) = 0$ whenever $r_1 = \cdots = r_G$.

A3 Monotone Scaling: $r_1 > r_2 \Rightarrow g(r_1) \geq g(r_2)$.

A4 Full Gating: $F(A, 0) = 0$ for all A.

A5 Identity Recovery: $F(A, 1) = A$.

A6 Non-Degeneracy: g is not identically zero on $(0, 1)$.

Theorem B.3 (Axiomatic Characterization)**.** *Let $F(A, r) = g(r) \cdot A + h(r)$ satisfy axioms A1–A6, and additionally assume:*

A7 Scale-Free Composition: *$g(r_1 \cdot r_2) = g(r_1) \cdot g(r_2)$ for all $r_1, r_2 \in [0, 1]$.*

Then $h \equiv 0$ and $g(r) = r^\alpha$ for some $\alpha \geq 1$, yielding $F(A, r) = A \cdot r^\alpha$.

Proof. **Step 1: $h \equiv 0$.** From A1, for any $A > 0$, $g(r)A + h(r) > 0$ must hold for arbitrarily small $A > 0$, requiring $h(r) \geq 0$. For $A < 0$, the same argument gives $h(r) \leq 0$. Hence $h(r) = 0$ for $r \in (0, 1]$. From A4, $g(0) \cdot A + h(0) = 0$ for all A, so $g(0) = 0$ and $h(0) = 0$.

Step 2: g is a monotone map $[0, 1] \to [0, 1]$. From A5, $g(1) = 1$. Combined with A3 (monotonicity), A6 (non-degeneracy), and Step 1, g maps $[0, 1] \to [0, 1]$ non-decreasingly with $g(0) = 0$ and $g(1) = 1$.

Step 3: Power-law form via A7. From A7, g satisfies the Cauchy functional equation on the multiplicative semigroup $[0, 1]$: $g(r_1 r_2) = g(r_1)g(r_2)$. Since g is continuous and monotone (from A3), the unique solution is $g(r) = r^\alpha$ for some $\alpha > 0$ (see, e.g., Aczél, *Lectures on Functional Equations*, Thm. 3). The constraint $\alpha \geq 1$ is imposed as a design requirement: for $\alpha < 1$, $r^\alpha > r$ when $r \in (0, 1)$, meaning the modulation would *amplify* advantages in low-reward regimes rather than suppress them, defeating the purpose of gating uniformly-poor groups. Thus, for gradient budget redistribution to function as intended, we restrict to $\alpha \geq 1$. □

Remark. Axiom A7 (scale-free composition) captures the intuition that the gating effect of two successive quality filters should compose multiplicatively — a natural requirement when rewards represent independent quality dimensions. This is analogous to the power-law characterization in utility theory (Debreu, 1960).

B.5 Gradient Decoupling: Multiplicative vs. Additive

Theorem B.4 (Additive Sign Flip)**.** *Let $A_i = (R_i - \bar{R})/(\sigma_R + \epsilon)$ be the GRPO advantage. The additive modulation $\tilde{A}_i^{\text{add}} = A_i + \beta \tilde{r}_i^\alpha$ ($\beta > 0$) can flip the advantage sign: there exist samples with $A_i < 0$ for which $\tilde{A}_i^{\text{add}} > 0$. The multiplicative modulation $\tilde{A}_i^{\text{mul}} = A_i \cdot \tilde{r}_i^\alpha$ preserves $\text{sign}(\tilde{A}_i^{\text{mul}}) = \text{sign}(A_i)$ for all $\tilde{r}_i > 0$.*

Proof. Additive: Take $A_i = -\delta < 0$ and $\tilde{r}_i = 1$. Then $\tilde{A}_i^{\text{add}} = -\delta + \beta > 0$ whenever $\beta > \delta$, flipping the sign.

Multiplicative: $\tilde{A}_i^{\text{mul}} = A_i \cdot \tilde{r}_i^{\alpha}$. Since $\tilde{r}_i^{\alpha} > 0$, $\text{sign}(\tilde{A}_i^{\text{mul}}) = \text{sign}(A_i)$. □

Multiplicative fusion thus achieves *sample-level gradient decoupling*: the ranking direction (from GRPO) and the precision magnitude (from $\tilde{r}_i^{\alpha}$) operate independently.

B.6 Non-Vanishing Reward Floor

Proposition B.5 (Non-Vanishing Reward Floor)**.** *With $k_{min} = 1$, the SNRA reward $\tilde{r}_i = 2/(1 + \exp(k \cdot e_i))$ satisfies: for $e_i \leq 1$, $\tilde{r}_i \geq 0.538$; for $e_i \leq 2$, $\tilde{r}_i \geq 0.238$; for $e_i \geq 5$, $\tilde{r}_i < 0.02$. Consequently, without requiring an explicit reward clipping threshold, SNRA's sigmoid decay naturally maintains a non-vanishing reward floor for near-miss samples (small e_i) while effectively zeroing out rewards for large errors, providing a smooth separation between informative and uninformative samples.*

Proof. Direct computation for $k = 1$: $\tilde{r}(1) = 2/(1 + e^1) = 0.538$, $\tilde{r}(2) = 2/(1 + e^2) = 0.238$, $\tilde{r}(5) = 2/(1 + e^5) = 0.013$. Since $\tilde{r}$ is monotonically decreasing in e_i, the bounds follow for all e_i in the stated ranges. □

B.7 Variance Suppression in Sparse Reward Regimes

In high-precision 3D tasks and other challenging reasoning problems, the model initially generates many "near-miss" samples with very low absolute rewards. Standard GRPO normalizes these low-reward samples to have unit variance within each group, which can introduce significant noise in the policy updates when the absolute reward signal is weak. AP-GRPO mitigates this by scaling the effective variance of the advantage by the absolute performance.

Theorem B.6. *In sparse reward regimes where the absolute rewards r_i are small, AP-GRPO reduces the variance of the advantage estimator by a factor proportional to $\epsilon^{2\alpha}$, where ϵ is the mean reward in the group.*

Proof. Consider a group of G trajectories, with rewards modeled as

$$r_i = \epsilon + \xi_i, \quad \mathbb{E}[\xi_i] = 0, \quad \text{Var}(\xi_i) = \sigma^2,$$

where $\epsilon \ll 1$ is a small baseline reward (sparse reward regime), and ξ_i represents small fluctuations around ϵ. For simplicity, assume G is large enough that the empirical group mean and variance satisfy

$$\bar{r} \approx \epsilon, \quad \sigma_r^2 \approx \sigma^2.$$

In standard GRPO, the within-group advantage is computed as

$$A_i = \frac{r_i - \bar{r}}{\sigma_r} \approx \frac{\xi_i}{\sigma_r}.$$

Thus, within the group,

$$\text{Var}(A_i) \approx \frac{\mathbb{E}[\xi_i^2]}{\sigma_r^2} \approx 1,$$

where we used $\sigma_r^2 \approx \sigma^2$. This unit variance persists even as $\epsilon \to 0$, implying that the magnitude of policy updates remains large, despite the absolute reward signal being tiny.

In AP-GRPO, the advantage is modulated by the absolute term:

$$A_i^{AP} = A_i \, r_i^{\alpha} \approx \frac{\xi_i}{\sigma_r} (\epsilon + \xi_i)^{\alpha}.$$

To approximate $\text{Var}(A_i^{AP})$ for small ϵ, we use a first-order Taylor expansion (Delta method) of $g(x) = x^{\alpha}$ around $x = \epsilon$:

$$(\epsilon + \xi_i)^{\alpha} = \epsilon^{\alpha} + \alpha\epsilon^{\alpha-1}\xi_i + O(\xi_i^2).$$

Substituting into A_i^{AP}, we obtain

$$\begin{aligned} A_i^{AP} &= \frac{\xi_i}{\sigma_r}\left(\epsilon^{\alpha} + \alpha\epsilon^{\alpha-1}\xi_i + O(\xi_i^2)\right) \\ &= \underbrace{\frac{\xi_i}{\sigma_r}\epsilon^{\alpha}}_{\text{leading term}} \\ &\quad + \underbrace{\alpha\frac{\xi_i^2}{\sigma_r}\epsilon^{\alpha-1}}_{\text{higher order}} + O(\xi_i^3). \end{aligned}$$

Assuming that ξ_i has finite moments up to order 4 (a mild regularity condition), the contribution of the higher-order terms to the variance is of smaller order in ϵ. In particular, for fixed noise scale σ and $\alpha \geq 1$, as $\epsilon \to 0$ the leading contribution to the variance comes from the first term:

$$\text{Var}(A_i^{AP}) = \epsilon^{2\alpha}\text{Var}\left(\frac{\xi_i}{\sigma_r}\right) + o(\epsilon^{2\alpha}) \approx \epsilon^{2\alpha},$$

where we have used $\mathrm{Var}(\xi_i/\sigma_r) \approx 1$ and absorbed higher-order contributions into $o(\epsilon^{2\alpha})$.

Thus, in the sparse reward regime,

$$\mathrm{Var}(A^{AP}) \approx \epsilon^{2\alpha} \quad \text{as} \quad \epsilon \to 0.$$

For $\alpha \geq 1$, we have

$$\lim_{\epsilon \to 0} \mathrm{Var}(A^{AP}) = 0,$$

showing that AP-GRPO suppresses the variance of the advantage estimator in proportion to the absolute reward level. This effectively down-weights noisy gradient updates when the model has not yet discovered reliably high-reward trajectories. □

B.8 Recovery of Sensitivity in High-Precision Regimes

While the suppression of variance is essential during the early stages of training, the model must eventually recover its sensitivity to subtle differences in reward in order to achieve high-precision alignment. We now show that as the policy improves and rewards approach the unit maximum ($r \to 1$), AP-GRPO asymptotically recovers the full optimization power of standard GRPO.

Theorem B.7. *In the high-precision regime where the absolute rewards r_i approach 1, the variance of the AP-GRPO advantage $\mathrm{Var}(A^{AP})$ converges to the variance of the standard GRPO advantage $\mathrm{Var}(A)$, allowing for fine-grained relative optimization.*

Proof. Consider a regime where the policy has learned most of the coarse structure of the task, and rewards lie in a small neighborhood of the maximum, i.e., $r_i \in [1-\eta, 1]$ for some small $\eta > 0$. Write

$$r_i = 1 - \delta_i, \quad \delta_i \geq 0, \quad \delta_i \to 0,$$

where δ_i represents the residual error of the i-th trajectory.

Let A_i denote the standard GRPO advantage within the group, which is centered and normalized so that (up to numerical constants) $\mathbb{E}[A_i] \approx 0$ and $\mathrm{Var}(A_i) \approx 1$. We assume A_i has finite second moment, i.e., $\mathbb{E}[A_i^2] < \infty$, which is standard in policy gradient analysis.

Under AP-GRPO, the weighted advantage is

$$A_i^{AP} = A_i\, r_i^{\alpha} = A_i\,(1-\delta_i)^{\alpha}.$$

For small δ_i, a first-order Taylor expansion of $(1-\delta_i)^{\alpha}$ around $\delta_i = 0$ yields

$$(1-\delta_i)^{\alpha} \approx 1 - \alpha\delta_i,$$

so that

$$A_i^{AP} \approx A_i(1-\alpha\delta_i) = A_i - \alpha\delta_i A_i.$$

We now examine the variance of A_i^{AP}. We have

$$\begin{aligned}\mathrm{Var}(A_i^{AP}) &= \mathrm{Var}(A_i - \alpha\delta_i A_i) \\ &= \mathrm{Var}(A_i) + \alpha^2\mathrm{Var}(\delta_i A_i) - 2\alpha\,\mathrm{Cov}(A_i, \delta_i A_i).\end{aligned}$$

Assume that as training progresses and the policy converges, the residual error δ_i shrinks in such a way that $\delta_i A_i \to 0$ in mean square, i.e.,

$$\lim_{t\to\infty} \mathbb{E}[(\delta_i A_i)^2] = 0,$$

where t denotes training time or iteration. This is a mild regularity condition: as rewards concentrate near 1, both δ_i and fluctuations in A_i within the group become small.

Under this assumption,

$$\lim_{\delta\to 0} \mathrm{Var}(\delta_i A_i) = 0 \quad \text{and} \quad \lim_{\delta\to 0} \mathrm{Cov}(A_i, \delta_i A_i) = 0.$$

Therefore,

$$\lim_{r\to 1} \mathrm{Var}(A^{AP}) = \lim_{\delta\to 0} \mathrm{Var}(A_i^{AP}) = \mathrm{Var}(A_i) \approx 1.$$

Thus, in the high-precision regime where rewards approach 1, the AP-GRPO advantage recovers the same variance level as standard GRPO. Intuitively, as $r_i^{\alpha} \to 1$, the absolute weighting becomes negligible, and AP-GRPO behaves like pure GRPO, allowing the optimizer to fully exploit fine-grained relative differences in reward. □

B.9 Optimization Dynamics of Dynamic Sharpness Scheduling

The SNRA operator $\sigma(k, e_i)$ is parameterized by the sharpness k. In this section, we provide a theoretical justification for the dynamic scheduling of $k(t)$, demonstrating how it addresses the gradient vanishing problem in the early stages and the precision bottleneck in the late stages of training.

Lemma B.8 (Gradient Extremum Localization)**.** *The gradient of the SNRA reward $\sigma(k, e_i) = \frac{2}{1+\exp(ke_i)}$ with respect to the numerical error e_i is given by:*

$$\nabla_{e_i}\sigma(k, e_i) = -\frac{2k\exp(ke_i)}{(1+\exp(ke_i))^2}.$$

The magnitude of this gradient $|\nabla_{e_i}\sigma|$ attains its maximum value when the error e_i satisfies

$$e_i = 0,$$

at which point the gradient magnitude is $|\nabla_{e_i}\sigma|_{\max} = \frac{k}{2}$.

Proof. We consider the magnitude of the gradient

$$\left|\nabla_{e_i}\sigma(k, e_i)\right| = \frac{2k\exp(ke_i)}{(1+\exp(ke_i))^2}.$$

Let $x = \exp(ke_i) > 0$. Then

$$f(x) = \frac{2kx}{(1+x)^2}.$$

Maximizing $|\nabla_{e_i}\sigma|$ over e_i is equivalent to maximizing $f(x)$ over $x > 0$. The derivative is

$$\begin{aligned} f'(x) &= 2k \cdot \frac{(1+x)^2 - x \cdot 2(1+x)}{(1+x)^4} \\ &= 2k \cdot \frac{1-x^2}{(1+x)^4}. \end{aligned}$$

Setting $f'(x) = 0$ gives $1 - x^2 = 0 \Rightarrow x = 1$, i.e., $\exp(ke_i) = 1$ and hence $e_i = 0$.

Evaluating at $e_i = 0$ (i.e., $x = 1$), we obtain

$$\left|\nabla_{e_i}\sigma(k, 0)\right| = \frac{2k}{(1+1)^2} = \frac{k}{2}.$$

Thus, the gradient magnitude attains its maximum $\frac{k}{2}$ at $e_i = 0$. □

Theorem B.9 (Global Search via Low Sharpness)**.** *For large initial errors $e_i \gg 0$, a high sharpness value k leads to exponential gradient vanishing, whereas a low sharpness k preserves the learning signal.*

Proof. Consider an initial error $E > 0$. The gradient magnitude at E for a given k is

$$\begin{aligned} \left|\nabla_{e_i}\sigma(k, E)\right| &= \frac{2k\exp(kE)}{(1+\exp(kE))^2} \\ &= \frac{2k}{\exp(kE) + 2 + \exp(-kE)}. \end{aligned}$$

When $k \to \infty$ for a fixed $E > 0$, the denominator grows exponentially as $O(\exp(kE))$, while the numerator grows only linearly as $O(k)$. Thus:

$$\lim_{k\to\infty}\left|\nabla_{e_i}\sigma(k, E)\right| = 0.$$

In other words, a fixed high-sharpness reward would fail to provide any meaningful gradient for samples that are far from the target.

By initializing the sharpness at a small value $k_{\min}$ at $t = 0$, we keep $\exp(k_{\min}E)$ at a moderate scale even for large E, thereby avoiding exponential suppression of the gradient. This maintains a non-vanishing learning signal that enables the model to begin orienting its predictions toward the ground truth from a wide range of initial errors. □

Theorem B.10 (Precision Refinement via High Sharpness)**.** *As the model converges ($e_i \to 0$), the reward surface becomes effectively flat for low k, leading to insufficient optimization pressure for high-precision tasks. Increasing k restores useful gradient magnitudes in this regime.*

Proof. Let $\delta > 0$ represent a small residual error in the late stage of training. The gradient magnitude at $e_i = \delta$ is

$$\left|\nabla_{e_i}\sigma(k, \delta)\right| = \frac{2k\exp(k\delta)}{(1+\exp(k\delta))^2}.$$

For $k\delta \ll 1$, a Taylor expansion $\exp(k\delta) = 1 + k\delta + O((k\delta)^2)$ gives

$$\begin{aligned} \left|\nabla_{e_i}\sigma(k, \delta)\right| &= \frac{2k(1 + k\delta + O((k\delta)^2))}{(2 + k\delta + O((k\delta)^2))^2} \\ &= \frac{k}{2} + O(k^2\delta). \end{aligned}$$

Thus, near $e_i = 0$, the gradient magnitude is approximately $\frac{k}{2}$ and does not vanish as $\delta \to 0$; instead, it is controlled primarily by k.

However, the *reward difference* between a nearly correct prediction and a perfect one is

$$\begin{aligned} |\sigma(k, \delta) - \sigma(k, 0)| &= \left|\frac{2}{1+\exp(k\delta)} - 1\right| \\ &= \frac{2}{1+\exp(k\delta)} - 1, \end{aligned}$$

where the last equality uses $\sigma(k, 0) = 1$. For $k\delta \ll 1$, we have

$$\begin{aligned} \exp(k\delta) &= 1 + k\delta + O((k\delta)^2), \\ \frac{2}{1+\exp(k\delta)} &= \frac{2}{2 + k\delta + O((k\delta)^2)} \\ &= 1 - \frac{k}{2}\delta + O((k\delta)^2), \end{aligned} \tag{11}$$

so that

$$|\sigma(k, \delta) - \sigma(k, 0)| \approx \frac{k}{2}\delta \quad \text{for } k\delta \ll 1,$$

which is $O(k\delta)$.

For a fixed low k, as $\delta \to 0$ this difference becomes extremely small and can be easily overshadowed by numerical noise or by regularization terms such as the KL-divergence penalty β_{KL}. In other words, with small k the reward surface becomes too flat to reliably distinguish between errors of size δ and 0 at high precision.

By increasing $k(t)$ to a larger value $k_{\max}$ in the late stage of training, both the local slope near $e_i = 0$ (which scales like $k/2$) and the reward contrast between $e_i = \delta$ and $e_i = 0$ (which scales like $k\delta$) are amplified by orders of magnitude. This compensates for the diminishing residual error δ, ensuring that the effective gradient signal on the policy parameters $\nabla_\theta \mathcal{L}$ remains large enough to continue driving the model toward the exact optimum $e_i = 0$. □

This coarse-to-fine curriculum guarantees that the model is neither stalled by gradient vanishing at the beginning nor limited by insufficient gradient resolution at the end.

C Numerical3D-50k Dataset Details

C.1 Data Sources

The Numerical3D-50K dataset is curated from multiple publicly available 3D scene understanding datasets. Table 6 lists the raw data sources.

Table 6: Raw data sources used to construct Numerical3D-50K.

Source Dataset	Primary Modality / Subtasks
ScanNet (Dai et al., 2017)	Room Size, Obj. Size, Depth
ARKitScenes (**?**)	Distance, Depth
Matterport3D (Chang et al., 2017)	Room Size, Route Planning
3RScan (Wald et al., 2019)	Obj. Count, Spatial Relations
SUN RGB-D (Song et al., 2015)	Depth, Obj. Size
Objectron (Ahmadyan et al., 2021)	Obj. Size, Distance
LLaVA-Video-178k (Zhang et al., 2024)	Dir., Order, Dist. (video)
SPAR-7M (Zhang et al., 2025a)	Spatial Relations, Position

C.2 Construction Pipeline

We construct the dataset through the following pipeline:

1. **Source aggregation.** We aggregate scene-level 3D annotations (point clouds, depth maps, object bounding boxes, camera poses) from the above sources.

2. **Question generation.** For continuous numerical tasks (distance, depth, room size, object size), we programmatically generate question–answer pairs using ground-truth 3D annotations. For discrete relational tasks (direction, order, counting, position), we use template-based generation with deterministic verification.

3. **RLVR conversion.** Both continuous and discrete tasks are converted into verifier-compatible dense rewards within the SNRA framework. For continuous tasks, the verifier computes scalar prediction error (e.g., absolute or normalized deviation), which is directly mapped to a dense reward via SNRA. For discrete tasks, we first compute a graded verifier score $V_\tau(o) \in [0, 1]$, define the logical miss $u = 1 - V_\tau(o)$, and map it through the scaling function Φ_τ into a unified error space (see Appendix F).

4. **Filtering and balancing.** We filter and subsample to obtain the final 50K samples, balancing across continuous numerical and discrete relational subtasks as shown in Fig. 5.

C.3 Overlap Check

We verify that no scene ID or video ID in Numerical3D-50K overlaps with the test splits of any evaluation benchmark (VSI-Bench, CV-Bench, MMSI, MindCube, BLINK). The filtering is performed at the scene/video level rather than the question level, ensuring that even unseen question variants from seen scenes are excluded.

D Extended Experimental Results

D.1 Experimental Setup and Hyperparameters

All experiments are conducted on 8×H100 (80GB) GPUs using DeepSpeed ZeRO-3 for memory-efficient training. We adopt the following default hyperparameters unless otherwise stated:

- **Base model:** Qwen2.5-VL-7B-Instruct (primary), with cross-architecture validation on Qwen2.5-VL-3B, LLaVA-OneVision-7B, and Qwen3-VL-8B-Instruct.
- **Training data:** Numerical3D-50K.
- **Learning rate:** 1×10^{-6} with cosine decay and 50-step warmup.
- **Batch size:** 128 prompts per step, each generating $G = 16$ rollout samples.

- **Compute:** 8×H100 GPUs, approximately 48 hours per full training run.
- **SNRA parameters:** $k_{\min} = 1.0$, $k_{\max} = 100.0$, schedule midpoint $\tau = 0.5$, steepness $s = 10$, format weight $\lambda = 0.1$.
- **AP-GRPO parameters:** absolute scaling exponent $\alpha = 1.0$, clipping ratio $\epsilon = 0.2$, KL coefficient $\beta_{\text{KL}} = 0.001$.
- **Generation:** temperature 1.0, top-p 0.95, max new tokens 1024.
- **Evaluation:** greedy decoding (temperature 0) on all benchmarks.

For RL baseline comparisons (Table 7), all methods use identical backbone, dataset, verifier, and total training budget to ensure fair comparison.

D.2 Comparison with Recent RL Baselines

To establish that the gain of AP-GRPO is not merely over vanilla GRPO but reflects a genuine algorithmic advantage for continuous numerical tasks, we reproduce six recent RL/RLVR baselines under matched settings: same Qwen2.5-VL-7B backbone, same Numerical3D-50K dataset, same verifier, and same training budget. As shown in Table 7, AP-GRPO outperforms all baselines by a clear margin. Methods designed for discrete logic tasks (e.g., Reinforce++, DAPO) struggle with continuous numerical precision, while Dr. GRPO — which addresses GRPO optimization pathologies — achieves 56.12 but remains 3.85 points below AP-GRPO. This confirms that the gradient budget redistribution mechanism provides a benefit specific to verifiable continuous numerical reasoning that general GRPO improvements do not capture.

Table 7: Comparison with recent RL baselines on VSI-Bench (**Qwen2.5-VL-7B**). All methods use the same backbone, data, verifier, and training budget.

Method	VSI-Bench Avg. (%)
Reinforce++	51.95
DAPO	53.47
Standard GRPO	54.40
ReMax	54.36
GSPO	55.41
Dr. GRPO	56.12
CISPO	57.58
AP-GRPO (Ours)	**59.97**

D.3 Advantage Fusion Strategies

We compare the multiplicative formulation of AP-GRPO against additive and mixed alternatives. The multiplicative form uniquely satisfies sign preservation, zero-mean approximate invariance, monotone absolute scaling, and gradient decoupling (see Appendix B.3 for formal analysis).

Table 8: Advantage fusion strategies on VSI-Bench (**Qwen2.5-VL-7B**).

Fusion Strategy	Avg. Acc (%)
Additive ($A_i + \beta \tilde{r}_i^{\alpha}$)	56.63
Mixed (multiply + add)	55.44
Multiplicative ($A_i \cdot \tilde{r}_i^{\alpha}$)	**59.97**

E Extended Ablation Studies and Sensitivity Analysis

To further validate the robustness and internal mechanisms of the *Smooth Operator* framework, we provide a series of controlled experiments focusing on hyperparameter sensitivity and operator design.

E.1 Dynamics of Sharpness Schedule (k_{min} and k_{max})

The SNRA operator relies on a dynamic sharpness schedule. We analyze the impact of the starting point (k_{min}) and the terminal bound (k_{max}) separately.

Initial Smoothness k_{min}. As shown in Table 9, the framework is highly robust to k_{min} within the $(0, 1]$ range. However, performance degrades when $k_{min} > 1$, as a prematurely sharp reward surface creates a "cold start" problem, penalizing early-stage exploration.

Terminal Sharpness k_{max}. Table 10 illustrates a parabolic performance trend. While increasing k enhances precision, an excessively large k_{max} (e.g., 400) leads to reward sparsity and high-variance gradients. $k_{max} \approx 100$ serves as the optimal "sweet spot" for fine-grained 3D reasoning.

E.2 Sensitivity of Absolute Scaling Factor (α)

We investigate the impact of the integer-scaled factor α in AP-GRPO. Since $R_i \in [0, 1]$, the absolute reward component R_i^{α} diminishes quadratically when $\alpha = 2$. This leads to *advantage vanishing*, where the gradients become too small to guide the model beyond the SFT baseline (see Table 11).

Table 9: Sensitivity analysis of initial k_{min} on VSI-Bench accuracy.

Hyperparameter	**Varying Initial k_{min}**				
Value	0.1	0.5	**1.0 (Default)**	2.0	5.0
VSI-Bench Acc. (%)	59.8	60.1	**60.0**	57.2	54.5
Training Status	Stable	Stable	**Stable**	Unstable	Degenerated

Table 10: Sensitivity analysis of peak sharpness k_{max} on VSI-Bench accuracy.

Hyperparameter	**Varying k_{max}**				
Value	25	50	**100 (Default)**	200	400
VSI-Bench Acc. (%)	54.8	57.6	**60.0**	59.1	57.9
Δ Performance	-5.2	-2.4	-	-0.9	-2.1

Table 11: Impact of scaling factor α on training stability.

Scaling Factor	$\alpha = 0$ (GRPO)	$\alpha = 1$ (AP-GRPO)	$\alpha = 2$
Final Accuracy (%)	54.4	**60.0**	49.7
Training Status	Stable	**Optimal**	**Vanishing**
Gradient Norm	Normal	Normal	≈ 0

E.3 Sensitivity to Group Size G

Since both the mean and standard deviation in AP-GRPO depend on the group size G, we evaluate how performance varies across different group sizes under the same training budget and backbone (Qwen2.5-VL-7B).

Table 12: Sensitivity to group size G on VSI-Bench accuracy.

Group Size (G)	4	8	**16 (Default)**
VSI-Bench Acc. (%)	57.36	58.84	**59.97**

A clear monotonic trend is observed: larger group size yields better performance, consistent with the intuition that larger G provides more reliable within-group mean/variance estimates and thus more stable relative ranking. We did not test larger G values due to GPU memory constraints. From a mechanism perspective, when G is smaller, relative statistics become noisier; however, the absolute modulation term $\tilde{r}_i^{\alpha}$ still provides some robustness by preventing the update from being determined purely by relative normalization, which explains why performance degrades gradually rather than collapsing sharply as G decreases.

F Discrete Verification Scores and the Scaling Function Φ

This section details how we construct (i) task-specific verifier scores $V_\tau(o) \in [0, 1]$ that provide partial credit for discrete relational subtasks, and (ii) a scaling function Φ_τ that maps these scores into a unified non-negative error space compatible with SNRA.

F.1 From Discrete Scores to a Unified Error

For a discrete subtask $\tau \in \mathcal{T}_{\text{disc}}$, the verifier produces a graded score $V_\tau(o) \in [0, 1]$ for a model output o. We convert this score into a *logical miss*:

$$u \triangleq 1 - V_\tau(o) \in [0, 1], \tag{12}$$

and map it into an error scalar:

$$e_{\text{disc}} = \Phi_\tau(u) \in [0, +\infty). \tag{13}$$

The mapping is designed to satisfy:

$$\Phi_\tau(0) = 0, \quad \lim_{u \to 1} \Phi_\tau(u) = +\infty, \quad \Phi_\tau \text{ is monotone increasing.} \tag{14}$$

The first condition ensures that a fully correct discrete prediction ($V_\tau = 1$) corresponds to zero error; the divergence at $u \to 1$ ensures that complete failures ($V_\tau \to 0$) can be mapped to arbitrarily large penalties; the monotonicity ensures that higher verification scores always yield smaller errors.

We define a unified error space $E = [0, \infty)$. In practice, we employ numerical stabilizers (e.g., ϵ-clipping for logs and $e_{\max}$ for invalid parses) which may upper-bound realized error values while preserving non-negativity and the intended ordering.

F.2 Log-Scaled Φ_τ and Calibration

We adopt a log-based unbounded scaling:

$$\Phi_\tau(u) = \eta_\tau \left[-\log\left(1 - u + \epsilon\right)\right]^{\gamma_\tau}, \quad u \in [0, 1), \tag{15}$$

where $\eta_\tau > 0$ is a scale factor, $\gamma_\tau > 0$ controls curvature, and $\epsilon \in (0, 1)$ is a numerical stabilizer.

Since $-\log(1-u+\epsilon) \geq 0$ and grows without bound as $u \to 1$ (for small ϵ), Φ_τ maps $u \in [0,1)$ to $[0,+\infty)$, meeting the unified error-domain requirement in theory.

In implementation, we clip u to avoid evaluating $\log(0)$:

$$u \leftarrow \min(u,\, 1-\epsilon)\,, \tag{16}$$

which yields a finite but very large error for exact failures, while preserving numerical stability.

To ensure that (near-)failures fall into the high-penalty region of SNRA at the end of training, we calibrate η_τ using the terminal sharpness $k_{\max}$ and a target near-zero reward level $\epsilon_r \in [10^{-3}, 10^{-2}]$. Recall SNRA:

$$\hat{\sigma}(k,e) = \frac{2}{1+\exp(ke)}. \tag{17}$$

We first compute the target error $e^\star$ that corresponds to reward ϵ_r under $k_{\max}$:

$$\hat{\sigma}(k_{\max}, e^\star) = \epsilon_r \;\Rightarrow\; e^\star = \frac{1}{k_{\max}} \ln\left(\frac{2}{\epsilon_r} - 1\right). \tag{18}$$

Let $V_{\min}$ denote the score floor treated as "complete failure" (typically $V_{\min} = 0$), so that $u_{\text{fail}} = 1 - V_{\min}$. With the clipped implementation, this becomes:

$$u_{\text{fail}} = \min(1 - V_{\min},\, 1-\epsilon) = 1-\epsilon. \tag{19}$$

We calibrate η_τ by enforcing $\Phi_\tau(u_{\text{fail}}) = e^\star$:

$$\eta_\tau = \frac{e^\star}{[-\log(1-u_{\text{fail}}+\epsilon)]^{\gamma_\tau}} = \frac{e^\star}{[-\log(2\epsilon)]^{\gamma_\tau}}. \tag{20}$$

This guarantees that a score at the failure floor is mapped to an SNRA reward approximately ϵ_r when $k(t)$ reaches $k_{\max}$.

Unless stated otherwise, we share a single η across all discrete tasks to keep the reward scale consistent, and tune γ_τ mildly across task types (typically $\gamma_\tau \in [1,2]$).

F.3 Verifier Designs with Partial Credit

We now specify $V_\tau(o)$ for the subtasks within our dataset. As illustrated in Fig. 5, our **Numerical3D-50k** corpus is strategically balanced between metric-scale estimation and logical spatial reasoning. Specifically, we detail the verifiers for the four major discrete subtasks: Relative Direction, Appearance Order, Counting, and Position Relationship.

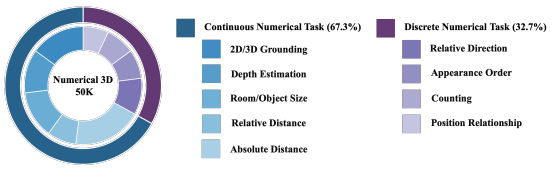


Figure 5: **Composition of the Numerical3D-50k dataset.** The hierarchical distribution illustrates the primary split between continuous numerical tasks ($\mathcal{T}_{\text{num}}$) and discrete relational tasks ($\mathcal{T}_{\text{disc}}$). The inner ring details the specific subtasks for which we design deterministic verifiers and partial credit functions as described in Sec. F.3.

F.3.1 Relative Direction

The goal is to predict the direction of object A relative to B (e.g., 4-way or 8-way direction bins). Let the ground-truth class be c and the predicted class be $\hat{c}$, both mapped to indices $i, j \in \{0, \ldots, K-1\}$, where $K \in \{4, 8\}$. We define the circular distance on the direction ring:

$$d(c,\hat{c}) = \min(|i-j|,\, K - |i-j|). \tag{21}$$

We then assign a graded score:

$$V_{\text{dir}}(o) = \begin{cases} 1, & d(c,\hat{c}) = 0, \\ \eta_{\text{dir}}, & d(c,\hat{c}) = 1, \\ 0, & d(c,\hat{c}) \geq 2, \end{cases} \tag{22}$$

where $\eta_{\text{dir}} \in (0,1)$ is a fixed near-miss credit (we use $\eta_{\text{dir}} = 0.5$ by default). This scheme gives partial credit to predictions that are off by one adjacent direction bin, while assigning zero credit to larger angular mistakes.

When continuous geometry is available (optional), we can compute the ground-truth relative bearing angle θ and the predicted bearing $\hat{\theta}$, and use a smooth angular score:

$$\Delta \triangleq \text{wrap}(|\theta - \hat{\theta}|) \in [0, \pi], \tag{23}$$

$$V_{\text{dir}}(o) = \exp\left(-\frac{\Delta^2}{2\sigma_\theta^2}\right), \tag{24}$$

where σ_θ controls angular tolerance, yielding fully continuous partial credits.

F.3.2 Appearance Order

This task evaluates whether the model correctly predicts the appearance/visibility order of objects. We support both pairwise and listwise formulations.

For a pairwise query on objects A and B, the verifier extracts their first-visible indices (or timestamps) t_A and t_B from the annotated sequence. Let

the predicted relation be $\widehat{\text{ord}} \in \{A \prec B,\ B \prec A\}$. The ground-truth relation is induced by $\text{sign}(t_A - t_B)$. We define:

$$V_{\text{ord}}(o) = \mathbb{I}\Big[\widehat{\text{ord}} = \text{sign}(t_B - t_A)\Big] \cdot \left(1 - \exp\left(-\frac{|t_A - t_B|}{\beta}\right)\right), \quad (25)$$

where $\beta > 0$ sets a margin scale. The term $1 - \exp(-|t_A - t_B|/\beta)$ down-weights nearly-tied cases, reflecting that these are intrinsically ambiguous even when the predicted order matches.

For listwise ordering with n objects, the verifier compares the predicted permutation $\hat{\pi}$ against the ground-truth permutation π using the Kendall inversion count $K(\pi, \hat{\pi})$. The normalized score is:

$$V_{\text{ord}}(o) = 1 - \frac{K(\pi, \hat{\pi})}{\binom{n}{2}}, \quad (26)$$

which lies in $[0, 1]$ and provides partial credit proportional to how many pairwise order constraints are satisfied.

F.3.3 Counting

The counting task requires predicting an integer $\hat{n}$ for a ground-truth count n. We parse $\hat{n}$ from the model output and define the absolute deviation:

$$\Delta \triangleq |\hat{n} - n|. \quad (27)$$

We use an exponentially decaying correctness score:

$$V_{\text{cnt}}(o) = \exp\left(-\frac{\Delta}{\tau}\right), \quad (28)$$

where $\tau > 0$ controls tolerance. With $\tau = 1$, an off-by-one prediction receives e^{-1} partial credit, and larger errors decay smoothly, avoiding binary sparsity.

When count ranges vary significantly across tasks (optional), we can use a normalized linear score:

$$V_{\text{cnt}}(o) = \max\left(0,\ 1 - \frac{|\hat{n} - n|}{\max(n, 1) + c}\right), \quad (29)$$

where $c > 0$ prevents instability when n is small.

F.3.4 Position Relationship

This task evaluates spatial relations between objects (e.g., `left-of`, `in-front-of`, `inside`, `overlap`, `near`). Since multiple relations can simultaneously hold, we treat this as a multi-label verification problem whenever applicable.

Let S be the ground-truth set of relations and $\hat{S}$ be the predicted set extracted from the output. We use Jaccard similarity as the verifier score:

$$V_{\text{pos}}(o) = \frac{|\hat{S} \cap S|}{|\hat{S} \cup S|}. \quad (30)$$

This yields partial credit for predicting a correct subset, penalizes missing relations, and also penalizes spurious relations.

If a dataset defines mutually exclusive relation labels, we can optionally define a relation graph with shortest-path distance $d(c, \hat{c})$ between the predicted and ground-truth labels, and use:

$$V_{\text{pos}}(o) = 1 - \frac{d(c, \hat{c})}{d_{\max}}, \quad (31)$$

where $d_{\max}$ is the maximum graph distance. This assigns partial credit when the predicted relation is semantically close to the correct one.

F.4 Final Discrete Error for SNRA

Given $V_\tau(o)$ from the task-specific verifiers above, we compute $u = 1 - V_\tau(o)$ and apply the log-scaled mapping:

$$e_{\text{disc}} = \Phi_\tau(1 - V_\tau(o)) = \eta_\tau \left[-\log\left(V_\tau(o) + \epsilon\right)\right]^{\gamma_\tau}. \quad (32)$$

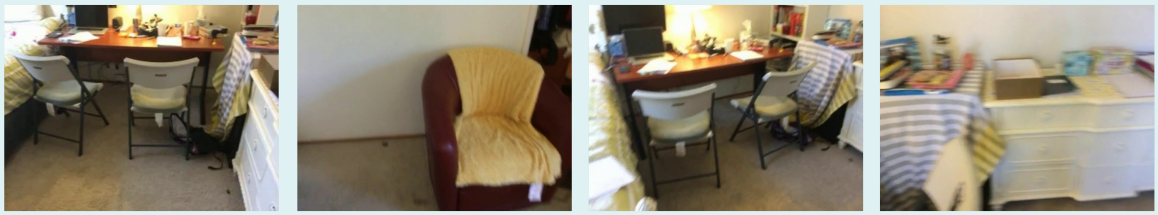
This mapping is non-negative, monotone decreasing in V_τ, and unbounded as $V_\tau(o) \to 0$, hence $e_{\text{disc}} \in [0, +\infty)$ in theory.

The resulting e_{disc} is fed into SNRA to produce the smooth numerical reward for discrete tasks:

$$\tilde{r} = \hat{\sigma}(k(t), e_{\text{disc}}). \quad (33)$$

With the calibration in Eq. (18)–(20), fully incorrect (or near-zero) discrete predictions receive near-zero rewards at late training (high k), while partially correct predictions remain within gradient-sensitive regions, improving sample utilization and stabilizing AP-GRPO updates.

G Case Visualization



Question: (Object Absolute Distance)
Measuring from the closest point of each object, what is the **distance** between the door and the window (in meters)?

Response: <think>The video shows one small bedroom from multiple angles, featuring a desk, chairs, armchair, and dresser. The door is near the dresser (left images), and the window is behind the desk (right images). From closest points, the distance along the wall appears approximately 2.7 meters based on typical room proportions and visible layout.</think> <answer> **2.7** </answer>

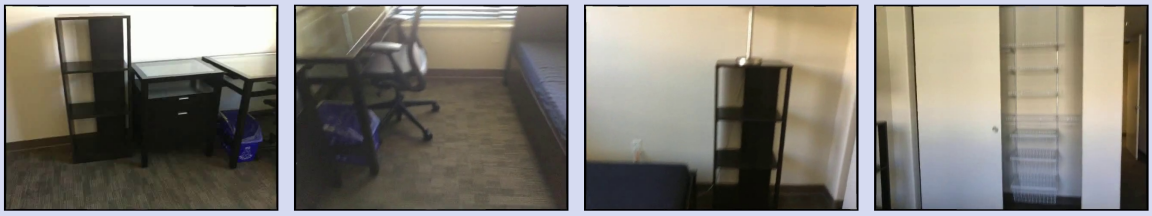
GT: 3.1 **MRA:.5:.95:.05: 0.9**

Question: (Object Size Estimation)
What is the length of the longest dimension (length, width, or height) of the tv, measured in centimeters?

Response: <think>The images show a living room with a wall-mounted TV above an arched niche. Comparing the TV size to nearby frames (~50–60 cm) and shelf proportions, its longest dimension (diagonal) is estimated at approximately 97 cm, consistent with a typical 43-inch screen including bezel.</think><answer> **97** </answer>

GT: 98 **MRA:.5:.95:.05: 1**

Figure 6: Cases of VSI-Bench



Question: (Room Size Estimation)
What is the size of this room (in square meters)? If multiple rooms are shown, estimate the size of the combined space.

Response: <think>The video shows a single compact bedroom with a bed, desk, chair, shelving unit, and closet. The layout is narrow and elongated. Visually estimating, the bed is about 2m long, and the room dimensions appear roughly 3 m wide by 5 m long, suggesting an area of about 15 m². Considering typical compact single-room layouts and furniture density, 15.9 square meters is a reasonable estimate. </think> <answer> **15.9** </answer>

GT: 14.7 **MRA:.5:.95:.05: 0.9**

Question: (Object Count)
How many trash bin(s) are in this room?

Response: <think>The video shows a single compact bedroom with a bed, desk, chair, shelving unit, and closet. The layout is narrow and elongated. Visually estimating, the bed is about 2 m long.</think> <answer> **2** </answer>

GT: 2 **MRA:.5:.95:.05: 1**

Figure 7: Cases of VSI-Bench