

# PDCR: Perception-Decomposed Confidence Reward for Vision-Language Reasoning

## Supplementary Material

### Appendix Contents

10. Broader Impact	1
11. Ethics Statement	1
12. Training Procedure Pseudocode	2
13. Experimental Results on Additional Model Backbone	3
14. Segmentation Detail	3
15. Label Acquisition for Skill Analysis	4
15.1 . Annotation Setup	4
15.2 . Validation of Label Quality	4
15.3 . Qualitative Examples of Skill Decomposition	5
16. Implementation Details	6
16.1 . Training Framework and Hyperparameters	6
16.2 . Prompt Template for Training and Inference	7
17. Ablation Study on Visual Dependence Calculation for Skill Decomposition	8
18. Qualitative Comparisons of Generated Reasoning	9
19. Limitations and Future Works	14

---

### 10. Broader Impact

This work introduces a framework for improving the reasoning capabilities of multimodal Large Language Models. By leveraging the model’s intrinsic confidence dynamics, our method provides fine-grained, step-level supervision, and decomposes this signal to align with the heterogeneous skills of perception and reasoning. This is achieved without the overhead of training separate reward models or requiring manual data annotation.

### 11. Ethics Statement

This research strictly adheres to academic integrity standards, ensuring all prior work is properly cited and acknowledged. Furthermore, our experiments utilize only publicly available datasets and do not involve the collection of sensitive or personally identifiable information.

## 12. Training Procedure Pseudocode

We outline our Perception-Decomposed Confidence Reward (PDCR) training procedure in Algorithm 1. This pseudocode provides a step-by-step specification of the method summarized in Section 5. The **highlighted lines** indicate the additional processing steps introduced in our proposed PDCR compared to PACR [9].

---

### Algorithm 1: Perception-Decomposed Confidence Reward (PDCR) Training

---

```

Input : Current policy  $\pi_\theta$ , Reference policy  $\pi_{\text{ref}}$ , Training batch  $\mathcal{D} = \{(\mathbf{I}_j, \mathbf{q}_j)\}_{j=1}^B$ , White Image  $\mathbf{I}_{\text{white}}$ 
Hyperparameters : Group size  $N$ , Discount  $\gamma$ , Weights  $\lambda_O, \lambda_P$ 

// Phase 1: Data Generation & Decomposition
Initialize buffers:  $\mathcal{T} \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$ ;
Initialize cluster pools:  $\mathcal{I}_{\text{vis}} \leftarrow \emptyset, \mathcal{I}_{\text{text}} \leftarrow \emptyset$ ;
foreach sample  $(\mathbf{I}, \mathbf{q})$  in  $\mathcal{D}$  do
  for  $i \leftarrow 1$  to  $N$  do
    Generate trajectory  $\tau^{(i)} = (h_1^{(i)}, \dots, h_{K_i}^{(i)}, \hat{Y}^{(i)}) \sim \pi_\theta(\cdot | \mathbf{I}, \mathbf{q})$ ;
    Compute outcome  $R^{(i)} \leftarrow \text{is\_equivalent}(\hat{Y}^{(i)}, Y_{gt})$ ;
    for  $k \leftarrow 1$  to  $K_i$  do
       $g_k^{(i)} \leftarrow \log \pi_\theta(Y_{gt} | \mathbf{I}, \mathbf{q}, H_{\leq k}^{(i)}) - \log \pi_\theta(Y_{gt} | \mathbf{I}, \mathbf{q}, H_{< k}^{(i)})$ ; // Conf. Gain
       $p_{\text{vis}} \leftarrow \log \pi_\theta(h_k | \mathbf{I}, \mathbf{q}, H_{< k}^{(i)})$ ;
       $p_{\text{white}} \leftarrow \log \pi_\theta(h_k | \mathbf{I}_{\text{white}}, \mathbf{q}, H_{< k}^{(i)})$ ;
       $V_k^{(i)} \leftarrow p_{\text{vis}} - p_{\text{white}}$ ; // Visual Dependence Score
    // Per-Trajectory Skill Decomposition
     $\mathbf{v}_{\text{sorted}} \leftarrow \text{Sort}(\{V_k^{(i)}\})$ 
     $c^* \leftarrow \text{OtsuMethod}(\mathbf{v}_{\text{sorted}})$ ; // Dynamic Threshold (Eq. 8)
    for  $k \leftarrow 1$  to  $K_i$  do
      if  $V_k^{(i)} \geq c^*$  then
        | Add  $(i, k)$  to  $\mathcal{I}_{\text{vis}}$ ;
      else
        | Add  $(i, k)$  to  $\mathcal{I}_{\text{text}}$ ;
    Store  $(\tau^{(i)}, \{g_k^{(i)}\})$  in  $\mathcal{T}, R^{(i)}$  in  $\mathcal{R}$ ;

// Phase 2: Decomposed Advantage Calculation
Compute sparse advantages  $A_{\text{outcome}}^{(i)}$  using  $\mathcal{R}$ ;
Compute returns  $\mathcal{G} = \{G_k^{(i)}\}$  from step gains  $\{g_k^{(i)}\}$ ;
Compute Stats:  $(\min_{\text{vis}}, \max_{\text{vis}}) \leftarrow \text{Stats}(\{G_k^{(i)} | (i, k) \in \mathcal{I}_{\text{vis}}\})$ ;
Compute Stats:  $(\min_{\text{text}}, \max_{\text{text}}) \leftarrow \text{Stats}(\{G_k^{(i)} | (i, k) \in \mathcal{I}_{\text{text}}\})$ ;
foreach step  $(i, k)$  in  $\mathcal{T}$  do
  if  $(i, k) \in \mathcal{I}_{\text{vis}}$  then
    |  $A_{\text{proc}, k}^{(i)} \leftarrow \text{Normalize}(G_k^{(i)}, \min_{\text{vis}}, \max_{\text{vis}})$ ;
  else
    |  $A_{\text{proc}, k}^{(i)} \leftarrow \text{Normalize}(G_k^{(i)}, \min_{\text{text}}, \max_{\text{text}})$ ;
   $A_{\text{total}, k}^{(i)} \leftarrow \lambda_O A_{\text{outcome}}^{(i)} + \lambda_P A_{\text{proc}, k}^{(i)}$ ;

// Phase 3: Policy Update
 $\mathcal{L}(\theta) \leftarrow -\frac{1}{B \times N} \sum \sum A_{\text{total}, k}^{(i)} \min(\dots)$ ; // GRPO Objective
 $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta)$ ;

```

---

Table 3. **Experimental results on the Qwen3-VL-8B-Instruct backbone.** We report the accuracy across seven evaluation benchmarks. We compare our method, **PDCR (ours)**, against strong baselines, including sparse-reward (GRPO), stabilized (DAPO), and naive dense-reward (PACR) methods. The **best** score in each column is in bold, and the second-best is underlined.

Base model + Method	General Visual Understanding				Visual Math & Hallucination			Average
	MMMU-Pro	MMMU	RealWorldQA	VisNumBench	MathVerse	MathVision	HallusionBench	
<i>Backbone model: Qwen3-VL-8B</i>								
Zero-shot Inference	49.1	55.2	70.1	29.1	68.1	48.3	74.9	56.4
GRPO [6]	49.6	55.0	<u>71.2</u>	<u>37.2</u>	69.3	49.9	75.6	58.3
DAPO [10]	50.0	55.5	<b>71.3</b>	37.0	69.2	50.3	<u>75.8</u>	58.4
PACR [9]	<u>49.9</u>	<u>56.1</u>	70.6	37.1	<u>69.9</u>	<u>50.4</u>	75.5	<u>58.5</u>
<b>PDCR (ours)</b>	<b>50.7</b>	<b>57.1</b>	70.7	<b>37.6</b>	<b>70.6</b>	<b>51.0</b>	<b>76.0</b>	<b>59.1</b>

### 13. Experimental Results on Additional Model Backbone

We further evaluate PDCR on the recently released [Qwen3-VL-8B-Instruct](#) (implementation details are outlined in Appendix 16). As shown in Table 3, PDCR demonstrates generalization to this stronger backbone, achieving a final average score of **59.1**. This performance outperforms the sparse GRPO baseline (58.3, **+0.8%**) and also improves upon the naive dense-reward baseline, PACR (58.5, **+0.6%**).

### 14. Segmentation Detail

A prerequisite for a process-based reward framework is the segmentation of the reasoning trajectory  $\tau^{(i)}$  into a discrete sequence of steps  $\{h_k^{(i)}\}_{k=1}^{K_i}$ . The step is the fundamental unit to which a reward or advantage is assigned.

Previous work in process-reward modeling has adopted several strategies to define this unit:

- **Supervised Formatting via SFT:** Some approaches utilize Supervised Fine-Tuning (SFT) to enforce a highly structured output format, such as training the model to generate explicit step delimiters (e.g., “Step 1:”, “Step 2:”). While this simplifies segmentation, it necessitates an SFT stage to instill these formatting constraints. As detailed in our implementation (Appendix 16), our training adopts a R1-style pipeline [1] that bypasses SFT to initiate reasoning training directly via RLVR. Consequently, strategies relying on pre-learned formatting are not applicable to our setting.
- **Confidence-based Segmentation:** Some approaches leverage intrinsic model signals to dynamically segment reasoning trajectories. For instance, Liu et al. [4] proposes, a method that divides reasoning steps based on the model’s confidence in predicting the next token. This approach aims to identify true decision points by observing how the model’s uncertainty changes, avoiding arbitrary segmentation based on fixed lengths or rule-based heuristics that may not align with the model’s internal reasoning flow.
- **Rule-Based Segmentation:** A more common, lightweight, and model-agnostic approach is to use simple heuristics to parse the generated text. This is the strategy adopted by the original PACR framework [9] and other process-reward based methods [3, 7, 11]. The most common heuristic is to define a new step at each newline character ( $\backslash n$ ), or after sentence-terminating punctuation (e.g., a period followed by a space). **In this work, we adopt this strategy due to its simplicity and to ensure a fair comparison with PACR [9].**

## 15. Label Acquisition for Skill Analysis

To empirically validate the heterogeneous nature of V-L reasoning ([**Observation 1**] in Section 4) and the effectiveness of our unsupervised skill decomposition (Section 5.1), we required a set of ground truth skill labels. Since no existing dataset provides step-level distinctions between perception and reasoning, we constructed a high-quality annotated set using a state-of-the-art Large Language Model as a programmatic annotator.

### 15.1. Annotation Setup

We sampled 10,000 reasoning trajectories generated by the `Qwen2.5-VL-7B-Instruct` model from our validation set. Each trajectory was first segmented into discrete steps following Appendix 14. Subsequently, we utilized `gpt-5-2025-08-07` to classify every individual step within these trajectories. To ensure the labels captured true visual dependency rather than just semantic relevance, we designed a rigorous prompt with a specific exclusion criterion. The model was provided with the full context (Image, Question, Full Reasoning Trace) and the specific Candidate Step, and instructed to output a boolean classification in strict JSON format. The exact system instruction provided to the annotator is as follows:

#### System Instruction for Skill Annotation

**Role:** You are an expert annotator. Decide whether a SPECIFIC step in a chain-of-thought is a “visual perception” step.

**Definition:** A visual perception step should primarily involve reading or describing visual content from the provided image (e.g., recognizing objects, numbers, colors, relative positions, text in the image, bars in a chart, etc.). Steps that perform logical deductions, arithmetic, or generic reasoning without referencing observed visual features are NOT visual-perception steps.

**Exclusion Criterion:** If a step refers to visual details that could already be inferred from earlier reasoning (i.e., it does not require directly inspecting the image again), classify that step as NOT visual perception.

**Output Format:** Return STRICT JSON with keys exactly:

- `is_visual_perception` (boolean)
- `reason` (string)

Do not include extra keys or text outside JSON.

---

#### Input Template:

Question: `{question}`

Full thinking process (all steps concatenated): `{full_process}`

Candidate step to classify: `{step_text}`

Now output JSON only.

**Distinguishing Perception from Contextual Recall.** A critical challenge in defining these labels is separating *active perception* from *visual memory*. As noted in the **Exclusion Criterion** above, a reasoning step might reference a visual object without actually looking at the image, relying instead on facts established in previous steps. We explicitly instructed the model to classify such instances as **Textual Reasoning**, ensuring that the **Visual Perception** label is reserved strictly for steps that extract *new* information from the raw visual signal.

### 15.2. Validation of Label Quality

To verify the reliability of this automated annotation, we performed a rigorous inter-annotator agreement study:

1. **Human Inter-Annotator Agreement:** Two human experts independently annotated a random subset of 100 steps. They achieved a Cohen’s Kappa of  $\kappa = 0.82$ , indicating that the binary distinction between perception and reasoning is well-defined and unambiguous to humans.
2. **Model-Human Alignment:** We compared the primary `gpt-5` annotations against the human consensus on the same subset. The model achieved a Kappa score of  $\kappa = 0.79$  (Table 4). This high alignment confirms that the model effectively acts as a reliable proxy for human judgment, correctly adhering to the strict definitions provided in the prompt.

Table 4. **Inter-Annotator Agreement Scores.** The strong correlation ( $\kappa > 0.75$ ) validates that the labels are reliable.

Comparison Pair	Metric Interpretation	Cohen’s $\kappa$
Human Expert 1 vs. Expert 2	Task Definition Quality	0.82
GPT-5 vs. Human Consensus	Proxy Reliability	0.79

### 15.3. Qualitative Examples of Skill Decomposition

To further illustrate the functional distinction between visual perception and textual reasoning (discussed as Observation 1 in the main text), we provide a set of representative reasoning steps in Table 5. Specifically, we show examples of individual steps that we labeled as either **Visual Perception** or **Textual Reasoning** based on the annotation process described in Appendix 15.1. As shown below, perception steps are characterized by explicit grounding in the image, whereas reasoning steps rely on logical deduction or internal knowledge.

Table 5. **Qualitative Examples of Skill Decomposition.** We show examples of steps that we labeled as **Visual Perception** or **Textual Reasoning**. This separation illustrates the heterogeneous nature of the V-L reasoning task.

Sample Step Content
<p><b>EXAMPLES OF VISUAL PERCEPTION STEPS</b>  <i>(Extracting attributes, counting, reading text, identifying coordinates)</i></p>
<ul style="list-style-type: none"> <li>• The image shows a group of people in red and white attire that resembles soccer jerseys, who appear to be in distress or disappointment.</li> <li>• The use of impasto technique, where thickly applied paint creates texture, is also evident.</li> <li>• The label ‘G’ is positioned in the middle of the solar system, closer to the sun than Venus and further from the sun than Earth.</li> <li>• The individual appears to be actively paddling and navigating the river, which suggests a sense of engagement and energy.</li> <li>• Looking at the diagram, the area that is light gray and surrounds the darker gray area (umbra) is the penumbra.</li> <li>• The diagram shows a plant cell with various labeled parts.</li> <li>• This person seems to be looking at a smaller, darker figure that is separate from the main silhouette, suggesting a sense of isolation or separation.</li> <li>• The image shows a group of players standing in front of a goalpost, and there is a ball present on the field.</li> <li>• The Moon is at stage 1, and the diagram indicates that it is at the position labeled as “Midnight.” Therefore, the time of day at stage 1 is midnight.</li> </ul>
<p><b>EXAMPLES OF TEXTUAL REASONING STEPS</b>  <i>(Deduction, arithmetic, applying external knowledge, conclusion drawing)</i></p>
<ul style="list-style-type: none"> <li>• Since the question asks for the total sum of the values, we need to add the value of the blue bar (45) to the value of the red bar (30), resulting in <math>45 + 30 = 75</math>.</li> <li>• Antarctica is the only continent in the world that is entirely below the Antarctic Circle, making it stand out as the only one often highlighted in this way on world maps.</li> <li>• Substituting these into the formula for the average value, we get:</li> <li>• Combining these observations, the time shown on the clock is 5:30.</li> <li>• To find the total area of the composite shape, we must calculate the area of the square and add it to the area of the triangle.</li> <li>• Therefore, based on the architectural style and location, the most likely location of the image is Xi’an.</li> <li>• In nature, this type of relationship is known as mutualism, which is a form of symbiotic relationship.</li> <li>• Applying the Pythagorean theorem to the right-angled triangle formed by the ladder and the wall, we can solve for the unknown length of the ladder.</li> <li>• Thus, we can infer that the vehicle entering the intersection has committed a traffic violation.</li> </ul>

## 16. Implementation Details

### 16.1. Training Framework and Hyperparameters

We perform all experiments using the EasyR1 framework [8]. Consistent with the R1-Zero style training [1], we apply Reinforcement Learning with Verifiable Rewards (RLVR) directly on the base model, bypassing any Supervised Fine-Tuning (SFT) stage. This ensures that the reasoning behaviors we observe are genuine results of the reinforcement learning process, rather than artifacts inherited from supervised fine-tuning.

All models (including baselines) are trained on the `Vision-SR1-47K` dataset for 1 epoch. To ensure a strictly fair comparison, we maintain a consistent configuration across all methods (GRPO [6], DAPO [10], PACR [9], and our PDCR). Common hyperparameters for the optimizer, RL process, and evaluation are detailed in Table 6. All models were trained on a single node with  $8 \times$  NVIDIA A100 80GB GPUs.

**Format reward.** Given that our training pipeline bypasses SFT, the model does not inherently possess the structural priors to format its reasoning steps. To address this, we incorporate a format reward alongside the outcome accuracy reward. A trajectory is assigned a format score  $R_{\text{fmt}}$  based on the valid enclosure of the reasoning process within `<think>` and `</think>` tags, and the final answer within `\boxed{\}`. Specifically, we augment the outcome reward  $R^{(i)}$  (Eq. 1) with a fixed bonus of  $R_{\text{fmt}} = 0.1$  for compliant trajectories, while assigning  $R_{\text{fmt}} = 0$  to invalid ones.

**Details on DAPO baseline.** For the DAPO baseline [10], we adopted a specific configuration to isolate the impact of reward stability. The full DAPO framework originally introduces three modifications to GRPO: (1) *Clipping* to promote diversity and prevent entropy collapse, (2) *Dynamic Sampling* to solve the vanishing advantage problem (where a batch has 0 variance in rewards), and (3) a *Token-Level Policy Gradient* loss to mitigate length bias.

As discussed in Section 7, our primary goal in comparing with DAPO is to determine if PDCR’s performance gains are merely a result of stabilizing the training signal against the vanishing advantage problem. Therefore, our implementation of the DAPO baseline specifically applies *only* Dynamic Sampling on top of the standard GRPO framework. This controlled configuration allows for a direct, isolated comparison regarding reward signal stability.

Table 6. **Key hyperparameters** standard to the RLVR framework in EasyR1 library [8] used for training and evaluation.

Hyperparameter	Value
<i>Common Hyperparameters</i>	
Optimizer	AdamW [5]
Learning Rate	1e-6
Weight Decay	1e-2
LR Schedule	Constant
Epochs	1
Freeze Vision Encoder	False
Global Batch Size	128
Rollout Batch Size	512
Rollout Size	8
Rollout Temperature	1.0
Rollout Top-p	0.99
Max Input Prompt Length	12800
Max Response Length	2048
Max Pixels	4194304
Min Pixels	262144
Use Kl Loss	True
KL Coef	1e-2
<i>PACR &amp; PDCR Specific</i>	
$\lambda_O$	0.7
$\lambda_P$	0.3
<i>Evaluation Generation</i>	
Temperature	0.0
Max New Tokens	2048

## 16.2. Prompt Template for Training and Inference

To ensure a fair comparison, we use the same system prompt template across GRPO, DAPO, PACR, and our PDCR method as follows:

### Reasoning Template for GRPO, DAPO, PACR, and PDCR

**SYSTEM:**

You are a helpful assistant.

**USER:**

{image}

{question}

You FIRST analyze the image/video and think about the detailed, step-by-step reasoning process as an internal monologue and then provide the final answer. The reasoning process MUST BE enclosed within <think> </think> tags. The final answer MUST BE put in \boxed{ }.

## 17. Ablation Study on Visual Dependence Calculation for Skill Decomposition



Figure 7. Visual Perturbation Strategies Evaluated for Skill Decomposition. To calculate the Visual Dependence Score ( $V_k^{(i)}$ , Eq. 7), we compare the model’s probability on the (a) **Original** image against four baselines: (b) **White** (Strategy adopted in main text), (c) **Gaussian Blur**, (d) **Gaussian Noise**, and (e) **Rotate**. Our analysis confirms that strategies which effectively destroy semantic information (b, c, d) yield high decomposition accuracy, whereas simple spatial transformation (e) preserves the visual content, leading to poor separation.

In Eq. 7 of the main text, we introduced the **Visual Dependence Score** ( $V_k^{(i)}$ ) as the log-likelihood ratio between the model’s probability given the original image  $\mathbf{I}$  versus a non-informative baseline  $\mathbf{I}_{\text{white}}$ . In this section, we examine whether a blank white image the optimal baseline for measuring visual reliance.

- Following the analysis in Huang et al. [2], which examined *token-level* visual dependency, we examine the optimal strategy to divide the visual grounding steps using our dynamic thresholding. Specifically, we evaluate four distinct perturbation methods to compute the reference probability  $p_{w,k}^{(i)} = \log \pi_{\theta}(h_k^{(i)} | \mathbf{I}_{\text{perturbed}}, q, H_{<k})$ . As illustrated in Figure 7, these include:
- (b) **White Image (Strategy adopted in main text)**: Replacing the visual input with a white image to remove all information.
  - (c) **Gaussian Blur**: Applying a Gaussian blur kernel ( $\sigma = 5.0$ ) to remove high-frequency details while preserving color statistics.
  - (d) **Gaussian Noise**: Adding random Gaussian noise ( $\mu = 0, \sigma = 1$ ) to corrupt the signal structure.
  - (e) **Rotate**: Rotating the input image by  $90^\circ$  clockwise to alter spatial orientation.

**Results and Analysis.** We evaluate each perturbation strategy on the Qwen2.5-VL-7B-Instruct backbone by measuring the skill decomposition accuracy achieved using our dynamic thresholding (Eq. 9). All results are validated against the ground-truth labeled dataset described in Appendix 15. Our findings offer a clear insight into the nature of visual perception in large multimodal models:

- **Information Destruction is Necessary:** The strategies that effectively destroy visual semantics (i.e., **White**, **Gaussian Blur**, and **Gaussian Noise**) perform similarly well, achieving decomposition accuracies in the range of **75.8% - 76.2%**. This suggests that the specific *type* of noise matters less than the fact that the visual information is removed. We opted for the **White Image** in PDCR simply because it is the most computationally efficient and reproducible method to achieve this ‘blinding’ effect.
- **Spatial Transformation is Insufficient:** In contrast, the **Rotation** strategy yields significantly poorer decomposition accuracy (**65.9%**). This indicates that modern VLMs possess a degree of rotation invariance; even when the image is turned  $90^\circ$ , the model can still extract sufficient features to maintain a high likelihood for perception steps. Consequently, the log-likelihood ratio  $V_k^{(i)}$  remains low for all steps, failing to distinguish perception from reasoning.

Thus, to accurately identify if a model is ‘seeing’, we should distort the visual signal entirely rather than merely transforming it. This confirms that information-destroying baselines are essential for robust skill decomposition, validating our choice of the White Image strategy in our experiments.

## 18. Qualitative Comparisons of Generated Reasoning

In this section, we present side-by-side comparisons between our proposed PDCR and baselines (GRPO [6], PACR [9]) on V-L reasoning tasks. All examples presented were generated by the final policies trained on the `Qwen2.5-VL-7B-Instruct` backbone. These examples highlight the core benefit of our decomposed reward signal: the mitigation of visual hallucinations. As observed in the figures below, baseline models often exhibit a specific failure mode where they generate logically coherent reasoning chains that are premised on incorrect visual extraction (highlighted in **red**). In contrast, PDCR guides the model to accurately ground its reasoning in the visual input (highlighted in **green**). This accurate perception serves as a robust foundation for subsequent logic, leading to correct final answers in tasks involving spatial estimation, counting, and geometric reasoning.

Generated Sample 1 by GRPO



**Question:** Which of the following options is a reasonable estimate of the angle between the ground surfaces in the figure?

**Choices:** (a) 130 degrees (b) 75 degrees (c) 92 degrees (d) 42 degrees

**Ground truth:** (b) 75 degrees

**Generation:**

<think>

To estimate the angle between the ground surfaces, we need to observe the intersection of the two surfaces. The angle appears to be quite sharp, suggesting a smaller angle. Let's analyze the options:

(a) 130 degrees: This is a very large angle, which is not typical for a ground surface intersection.

(b) 75 degrees: This is a moderate angle, but it's still relatively large.

(c) 92 degrees: This is a smaller angle, but it's still larger than what we typically see in ground surface intersections.

(d) 42 degrees: This is a very small angle, which is more plausible for the intersection of two ground surfaces.

Given the visual observation, the angle appears to be closer to 42 degrees. </think>  d

Generated Sample 1 by PDCR (ours)



**Question:** Which of the following options is a reasonable estimate of the angle between the ground surfaces in the figure?

**Choices:** (a) 130 degrees (b) 75 degrees (c) 92 degrees (d) 42 degrees

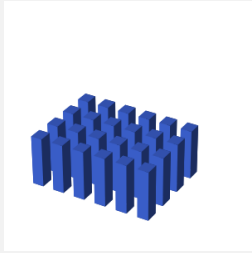
**Ground truth:** (b) 75 degrees

**Generation:**

<think>

To estimate the angle between the ground surfaces, we need to observe the intersection of the two surfaces. The angle appears to be less than 90 degrees but more than 45 degrees. The angle looks closer to 75 degrees than to 130 degrees or 42 degrees. Therefore, the most reasonable estimate is 75 degrees. </think>  b

### Generated Sample 2 by GRPO



**Question:** Which of the following options is a reasonable estimate of the number of cubes in the figure?

**Choices:** (a) 17 (b) 24 (c) 7 (d) 39

**Ground truth:** (b)

**Generation:**

<think>

To estimate the number of cubes in the figure, let's break it down step by step:

1. Visual Inspection: The figure appears to be a 3D structure with cubes stacked in a somewhat organized manner. It looks like a pyramid-like structure with a base and layers decreasing upwards.

2. Counting Layers:

- The base layer looks like it has 6 cubes.
- The next layer up has 5 cubes.
- The layer above that has 4 cubes.
- The top layer has 3 cubes.

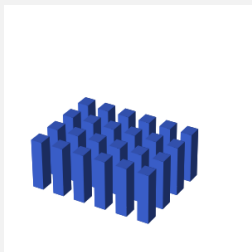
3. Summing Up: Adding these together gives us  $6 + 5 + 4 + 3 = 18$  cubes.

4. Comparison with Options:

- (a) 17 is close but slightly less.
- (b) 24 is significantly more.
- (c) 7 is much less.
- (d) 39 is much more.

Given the visual inspection and the summing up, the closest reasonable estimate is 17. </think>

### Generated Sample 2 by PDCR (ours)



**Question:** Which of the following options is a reasonable estimate of the number of cubes in the figure?

**Choices:** (a) 17 (b) 24 (c) 7 (d) 39

**Ground truth:** (b)

**Generation:**

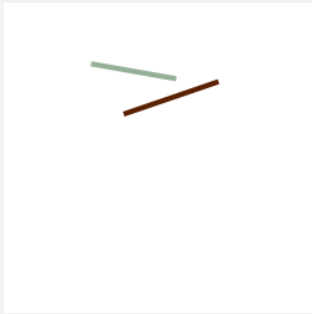
<think>

To estimate the number of cubes in the figure, we can break it down into smaller, more manageable sections. The figure appears to be a 3D structure with a grid-like pattern. By counting the cubes in one row and one column, we can estimate the total number.

- The figure has 6 rows.
- Each row appears to have 4 cubes.

So, the total number of cubes is approximately  $6 \text{ rows} * 4 \text{ cubes per row} = 24$  cubes. </think>

Generated Sample 3 by PACR



**Question:** Which of the following options is a reasonable estimate of the sharp angle between two lines in the figure?

**Choices:** (a) 67 degrees (b) 51 degrees (c) 29 degrees (d) 85 degrees

**Ground truth:** (c) 29 degrees

**Generation:**

<think>

To estimate the sharp angle between the two lines, we need to visually assess the angle formed by the intersection of the two lines.

The angle appears to be less than 90 degrees but more than 45 degrees.

- Option (a) 67 degrees is close to 90 degrees, which is not sharp. - Option (b) 51 degrees is a reasonable estimate as it is less than 90 degrees and more than 45 degrees. - Option (c) 29 degrees is too small for a sharp angle. - Option (d) 85 degrees is close to 90 degrees, which is not sharp.

Therefore, the most reasonable estimate for the sharp angle between the two lines is 51 degrees. </think>  b

Generated Sample 3 by PDCR (ours)



**Question:** Which of the following options is a reasonable estimate of the sharp angle between two lines in the figure?

**Choices:** (a) 67 degrees (b) 51 degrees (c) 29 degrees (d) 85 degrees

**Ground truth:** (c) 29 degrees

**Generation:**

<think>

To estimate the sharp angle between the two lines, we need to consider the relative orientation of the lines. The lines appear to be intersecting at a point, and the angle between them is not a right angle (90 degrees) but rather a smaller angle less than 45 degrees.

Looking at the options provided, the angle that seems to be smaller than 45 degrees and thus the most reasonable for a sharp angle between two intersecting lines is 29 degrees.

Therefore, the most reasonable estimate of the sharp angle between the two lines is 29 degrees. </think>  c

Generated Sample 4 by PACR



**Question:** Which of the following options is a reasonable estimate of the area percentage of the orange portion of the pie chart?

**Choices:** (a) (60%, 70%) (b) (80%, 90%) (c) (10%, 20%) (d) (30%, 40%)

**Ground truth:** (d) (30%, 40%)

**Generation:**

<think>

The image shows a pie chart with a single orange section. The orange section appears to cover less than a quarter of the entire circle, which is approximately 25%. Given the options, the orange section is clearly less than 30% of the pie chart. Therefore, the most reasonable estimate for the area percentage of the orange portion is (c) (10%, 20%). </think>  c

Generated Sample 4 by PDCR (ours)



**Question:** Which of the following options is a reasonable estimate of the area percentage of the orange portion of the pie chart?

**Choices:** (a) (60%, 70%) (b) (80%, 90%) (c) (10%, 20%) (d) (30%, 40%)

**Ground truth:** (d) (30%, 40%)

**Generation:**

<think>

The pie chart shows a large orange section that appears to cover more than a quarter but less than half of the circle. A quarter of a circle is 25%, and half is 50%. The orange section looks to be between 30% and 40% of the circle. Therefore, the most reasonable estimate for the area percentage of the orange portion is (30%, 40%). </think>  d

## 19. Limitations and Future Works

Our work presents a promising direction for self-rewarding V-L models, but has limitations that offer avenues for future research.

**Computational Cost.** The primary limitation of our framework is the computational overhead during the training phase. As shown in Figure 6-(b) of main paper, both PDCR and the dense-reward baseline (PACR) [9] require additional computation to estimate the step-level confidence gains (and in our case, an additional visual dependence scores). This results in a higher per-step training cost compared to standard sparse-reward methods like GRPO [6]. However, since our decomposition mechanism is lightweight, the cost of PDCR is comparable to that of PACR.

Crucially, we argued that this training overhead is a favorable trade-off for inference efficiency. As shown in Figure 6-(c)&(d), PDCR induces significantly more concise reasoning traces compared to GRPO. Consequently, while PDCR requires more GPU-hours to train, it yields a more efficient model for deployment. Future work could explore internal activation-based approximations to obtain these confidence scores without requiring full additional forward passes, potentially bridging the training cost gap.

**Binary Skill Decomposition.** Our method makes a simplifying assumption that V-L reasoning steps are a heterogeneous mixture of two *distinct* and *separable* skills. We then use Otsu’s method to enforce a single, hard threshold  $c^*$  between them. In reality, some steps may be hybrid (e.g., complex spatial reasoning that is both perceptual and logical). Our binary decomposition forces these hybrid steps into one of the two clusters, which may not be optimal. Future research could explore soft-clustering assignments or the identification of three or more distinct skill types.

## References

- [1] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 3, 6
- [2] Siyuan Huang, Xiaoye Qu, Yafu Li, Yun Luo, Zefeng He, Daizong Liu, and Yu Cheng. Spotlight on token perception for multimodal reinforcement learning. *arXiv preprint arXiv:2510.09285*, 2025. 8
- [3] Dong Bok Lee, Seanie Lee, Sangwoo Park, Minki Kang, Jinheon Baek, Dongki Kim, Dominik Wagner, Jiongdoo Jin, Heejun Lee, Tobias Bocklet, et al. Rethinking reward models for multi-domain test-time scaling. *arXiv preprint arXiv:2510.00492*, 2025. 3
- [4] Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, et al. Adaptivestep: Automatically dividing reasoning step through model confidence. *arXiv preprint arXiv:2502.13943*, 2025. 3
- [5] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6
- [6] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 3, 6, 9, 14
- [7] Zhaohui Yang, Chenghua He, Xiaowen Shi, Linjing Li, Qiyue Yin, Shihong Deng, and Daxin Jiang. Beyond the first error: Process reward models for reflective mathematical reasoning. *arXiv preprint arXiv:2505.14391*, 2025. 3
- [8] Shenzhi Wang Zhangchi Feng Dongdong Kuang Yuwen Xiong Yaowei Zheng, Junting Lu. Easyr1: An efficient, scalable, multi-modality rl training framework. <https://github.com/hiyouga/EasyR1>, 2025. 6
- [9] Eunseop Yoon, Hee Suk Yoon, Jaehyun Jang, SooHwan Eom, Qi Dai, Chong Luo, Mark A Hasegawa-Johnson, and Chang D Yoo. Pacr: Progressively ascending confidence reward for llm reasoning. *arXiv preprint arXiv:2510.22255*, 2025. 2, 3, 6, 9, 14
- [10] Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiase Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. 3, 6
- [11] Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, et al. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*, 2025. 3