

	CV Bench	V* Bench	MMVP	MMStar-V	MME-RealWorld-V	MMLU-Pro	Total
# Images	2638	191	300	750	6405	-	10284
# Questions	2638	191	300	750	11436	12032	27347

Table 3: **Benchmark statistics.**

A Table of Content

1. Sec. B elaborates the details of the considered five vision-centric benchmarks
2. Sec. C provides the analysis of fine-tuned VLM’s response length versus question difficulties.
3. Sec. D provides additional experimental results including additional comparison with self-training and our efforts to improve VLAA-thinking and Virgo.
4. Sec. E provides the full evaluation results on text-only reasoning benchmark, MMLU-Pro.
5. Sec. F provides implementation details in dataset generation, VLM training, and VLM inference.
6. Sec. G provides additional qualitative results of our dataset generation pipeline.

B Benchmark and evaluation details

We describe the details of each benchmark:

1. CV-Bench (Tong et al., 2024a) is a comprehensive benchmark of over 2k manually-inspected examples, evaluating visual understanding across domains such as object recognition, scene understanding, and visual reasoning. We report the mean accuracy over four categories: count, depth, distance, and relation.
2. V* Bench (Wu & Xie, 2024) targets fine-grained visual reasoning tasks that demand detailed analysis of visual elements. We report the mean accuracy over two categories: attribution and relation.
3. MMVP (Tong et al., 2024b) tests visual pattern recognition using “CLIP-blind pairs”—visually distinct images perceived as similar by CLIP—highlighting systematic limitations in VLMs.
4. MMStar-V includes tasks assessing instance-level reasoning, fine-grained perception (detecting subtle visual details), and coarse perception (understanding overall scene context). In total, it consists of 750 examples.
5. MME-RW-V. MME-RealWorld (Zhang et al., 2024) evaluates real-world visual understanding across domains such as autonomous driving, remote sensing, monitoring, diagrams, tables, and OCR, testing both perception and reasoning. From these, we select three perception tasks—Remote Sensing, Monitoring, and Autonomous Driving—and two reasoning tasks—Monitoring and Autonomous Driving—to form MME-RealWorld-V.
6. MMLU-Pro (Wang et al., 2024a) consists of over 12k questions from academic exams and textbooks, spanning 14 diverse domains. We report the mean accuracies over categories.

As a result, our evaluation provides a comprehensive view on the perceptual capabilities enabled by the training datasets under consideration. Table 3 shows the basic statistics of the considered benchmarks.

C Response length vs. question difficulty

Following prior works (Lightman et al., 2024; Snell et al., 2025), we define question difficulty with respect to a base VLM, *i.e.* Qwen2.5-VL-7B-Instruct. For each question, we

Approach	Avg	CV-Bench	V* Bench	MMVP	MMStar-V	MME-RW-V
Qwen2.5-VL-7B-Instruct	58.47	74.74	48.51	73.67	63.73	31.68
VLAA-thinking	42.32	68.50	53.53	66.67	0.53	22.38
+ only natural images	34.96	61.91	28.86	55.00	6.20	22.86
Virgo	50.87	67.22	44.14	57.67	57.60	27.71
+ improved formatting	52.58	68.94	46.54	66.33	53.47	27.60

Table 4: Attempted improvements on top of VLAA-Thinking and Virgo baselines.

Approach	V* Bench	CV-Bench	MMVP
Visual-CoT-7B-336	53.06	63.52	63
Qwen2.5-VL-7B-Instruct	48.51	74.74	73.67
+ LongPerceptualThoughts- SFT + DPO (Ours)	60.31	76.61	75

Table 5: Comparison with Visual CoT (Shao et al., 2024a) that reasons by explicitly modifying images, *e.g.* through cropping regions of interests.

estimate the model’s accuracy using 11 samples and bin the questions into two quantiles: easy and hard. Our analysis focuses on the outputs of the VLM fine-tuned via DPO on LongPerceptualThoughts. Fig 4 shows the distribution of response lengths across the easy and hard bins for four different tasks. We observe that the model naturally allocates more test-time compute—reflected in longer responses—for harder questions, where its original (pre-fine-tuning) performance was worse.

D Additional Results

Comparison with Visual-CoT Our work focuses on equipping VLMs with an internal search mechanism that unfolds through a textual inner monologue. Prior studies have incorporated intermediate representations by explicitly modifying images, *e.g.* cropping (Shao et al., 2024a). Here, we compare our approach with **Visual-CoT-7b-336**. Visual-CoT-7b-336 is trained in two stages, leveraging Vicuna-7B () and ViT-L/14-336 (). Its training data includes 438k question-answer pairs annotated with intermediate bounding boxes (Shao et al., 2024a).

As shown in Tab. 5, Visual-CoT-7b-336 underperforms even our base instruction-tuned VLM, Qwen2.5-VL-7B. Our method consistently surpasses baselines across diverse benchmarks, highlighting its strong generalizability.

VLAA-Thinking and Virgo adjustments. As we saw degradation in performance when training on both, Virgo and VLAA-Thinking, we spent additional time investigating the datasets and the model behavior they are causing which lead to these results.

We found that VLAA-Thinking consists of large proportions of math questions whereas natural image data is dominating the considered benchmarks as we focus on perceptual tasks. We hypothesize that this distribution shift might lead to lower performance. To investigate, we consider a version of VLAA-Thinking where we only keep the image subsets containing natural images, *i.e.*, ALLaVA-LAION and VizWiz, and randomly sample a subset of the same size. For Virgo, we found that predictions would not consistently respect the system prompt when formatting answers leading to inconsistencies with our regex-based evaluation. We thus explore a version of the dataset where we only copy the answer provided inside `\boxed{}` into `<answer>` tags, discarding the justification part of the answer, while keeping the thinking part of the dataset the same.

The results of both adjustments can be found in Table 4. We observe that training on only natural images in VLAA-Thinking hurts performance further, likely due to the limited data diversity. On the other hand, when applying improved answer formatting the results on

	Avg	Biology	Business	Chemistry	CompSci.	Econ.	Engin.	Health	History	Law	Math	Phil.	Physics	Psych.	Other
Qwen2.5-VL-7B-Instruct	48.07	68.62	55.77	44.79	49.51	61.26	34.26	47.68	43.57	24.89	50.41	38.88	47.19	60.65	45.56
DOCCI	32.99	51.60	42.33	22.61	37.32	43.48	18.89	32.76	22.31	10.26	40.19	29.46	25.56	51.13	33.98
VLAA-Thinking	21.56	25.24	27.76	15.11	20.73	25.47	7.64	24.45	29.40	13.35	26.72	20.04	17.78	21.43	26.73
Virgo	37.95	64.02	44.36	28.98	36.59	50.36	10.63	38.63	37.27	21.16	41.67	33.07	33.18	53.88	37.45
Ours - SFT	50.77	71.83	56.78	50.35	51.22	62.68	38.49	50.86	42.78	25.07	64.25	40.88	50.65	60.78	44.16
Ours - SFT + DPO	50.20	73.08	55.26	45.94	48.29	62.09	37.98	51.10	45.41	28.25	59.07	40.68	48.73	62.28	44.70

Table 6: Results for all categories of the MMLU-Pro dataset.

Virgo improve slightly from 50.87% to 52.58%. However, despite these adjustments, the datasets still fail to improve beyond the base model.

E Full MMLU-Pro Evaluation Results

We provide the detailed results on all MMLU-Pro categories in Table 6. We observe that the model fine-tuned on our LongPerceptualThoughts dataset consistently outperforms the baselines and provides improvements on top of the base model except for the Other category.

F Implementation Details

F.1 LongPerceptualThoughts

Data generation. Our framework consists of three stages: generates verifiable multiple-choice questions using \mathcal{M}_{LLM} , extracts simple chains of thought (CoTs) from vision-language models \mathcal{M}_{VLM} , and expands them into rich, long-form reasoning traces using frontier reasoning models \mathcal{M}_{Reason} .

1. In Stage 1, we use gpt-4o-mini-2024-07-18 with temperature 0.7. First, we prompt GPT-4o using the prompt in Fig. 5 to generate multiple-choice questions. Then, we parse the outputs by prompting GPT-4o again using the prompt in Fig. 6.
2. In Stage 2, we use Qwen2.5-VL-7B-Instruct with temperature 0.7, top_p 0.8, repetition_penalty, 1.05, and set number of samples per input to 3.
3. In Stage 3, we use R1-Distill-Qwen-32B with temperature 0.7, top_p 0.8, top_k 50, repetition_penalty, 1.05, and set number of samples per input to 3. To avoid outputs include phrases like “As the description says”, we explicitly define bad_words as “describe, description, described, describes, descriptions, mention, mentions, mentioned, misread, text, stated, says, mental”. The prompt is provided in Fig. ??

F.2 Training details

SFT Training. We fine-tune the language decoder with a batch size of 256, sweeping learning rates over $\{10^{-5}, 8 \times 10^{-6}, 6 \times 10^{-6}\}$. Training runs for up to 5 epochs with early stopping based on the average validation accuracy. We set the maximum image resolution to 512×512 and the input cutoff length to 1024.

DPO Training. We fine-tune the language decoder with a batch size of 256, sweeping learning rates over $\{1 \times 10^{-6}, 5 \times 10^{-7}, 1 \times 10^{-7}\}$. Training runs for up to 3 epochs with early stopping based on the average validation accuracy. We set the maximum image resolution to 512×512 and the input cutoff length to 1024. For DPO, we set β to 1. and following Pang et al. (2024), we include SFT loss with a weight of 0.5.

F.3 DOCCI Captions

We select the same 500 images used to generate our dataset. Next, we format the training dataset with the user prompt “Provide a detailed description of the image.”, prepending the image token and use the dense description provided in the dataset as the target answer of the model without further processing. We train the model using learning rate 8×10^{-6}

with batch size 256 for a maximum of 20 epochs. The training reaches maximum average accuracy on the validation set in the third epoch and we subsequently use this checkpoint to report results in the main paper.

F.4 VLAA-thinking

We preprocess the dataset into two different versions, discarding samples where no reasoning trace could be extracted. The first version uses 24,035 randomly selected samples from the original dataset containing 158,827 samples. The second version also 24,035 random samples, however, we filter the dataset for images from ALLaVA-LAION and VizWiz. The latter specifically contains natural images - similar to the setting we train and evaluate on. We use the official dataset³ provided and apply some minor processing to the dataset to format the samples into a similar format as ours. In particular, we extract the thinking process and the answer from the *ds_answer* column of the dataset and place these into `<think>` and `<answer>` tags respectively. We use the same system prompt as for our model (see Sec. F.7).

Training. We finetune the language decoder using batch size 256. For both versions, we perform hyper parameter tuning by sweeping learning rates $\{10^{-5}, 8 \times 10^{-6}, 6 \times 10^{-6}\}$. We train for a maximum of 5 epochs and perform early stopping based on the average accuracy on the validation datasets.

F.5 Virgo

We use the dataset introduced in Virgo (Du et al., 2025) as D_{SD} ⁴ as other versions are not publicly available and it provides the best average performance in their experiments. As instructed on the webpage we use the "conversation" column of the dataset which the authors report to be the final data used to train the Virgo-7B model. The conversation column is constructed as the correct response with the shortest length of 5 samples given each prompt.

We apply minor processing to the dataset to follow our format by replacing the `<|begin_of_solution|>` and `<|end_of_solution|>` with `<answer>` and `</answer>`. Similarly, we replace `<|begin_of_thought|>` and `<|end_of_thought|>` with `<think>` and `</think>`. Finally, we append "Format the answer with the letter of the correct option in parentheses." to the system prompt if the question is a multiple choice question. Overall, the resulting training dataset contains 14,540 samples.

Training. For training, we follow the setup described in F.4, i.e., performing basic hyper parameter tuning, with the only change to limit training to 3 epochs as we found that the model performance peaks early during training. Surprisingly, we achieve the best validation performance before the first epoch ends.

F.6 Evaluation

Inference setup. We use vLLM (Kwon et al., 2023) for inferencing all models with greedy decoding. Detailed settings can be found in Tbl. 7. Further, we resize images' longer side to 512 pixels preserving the aspect ratio if necessary. As the reasoning traces for MMLU-Pro tend to be longer for all models due to the difficulty of the task, we double the maximum number of new tokens generated. We use four NVIDIA RTX6000.

F.7 Prompt design

We provide the prompts used in formulating training and evaluation data:

1. **VLM training:** Inspired by the prompt provided by DeepSeek-R1 (DeepSeek-AI et al., 2025), we design the prompt used to evaluate VLMs to provide thoughts and answers, as shown in Fig. 10. For DOCCI descriptions, we use the prompt in Fig. 8
2. **VLM evaluation:** We evaluate VLMs that answer directly using the prompt in Fig. 9.

³<https://huggingface.co/datasets/UCSC-VLAA/VLAA-Thinking>

⁴<https://huggingface.co/datasets/RUC-AIBOX/Virgo-Visual-Long-Thought-Dataset>

Setting	Value
cutoff_length	2048
max_new_tokens	1024 (2048 for MMLU-Pro)
temperature	0.0
top_p	1.0
dtype	half

Table 7: vLLM inference settings.

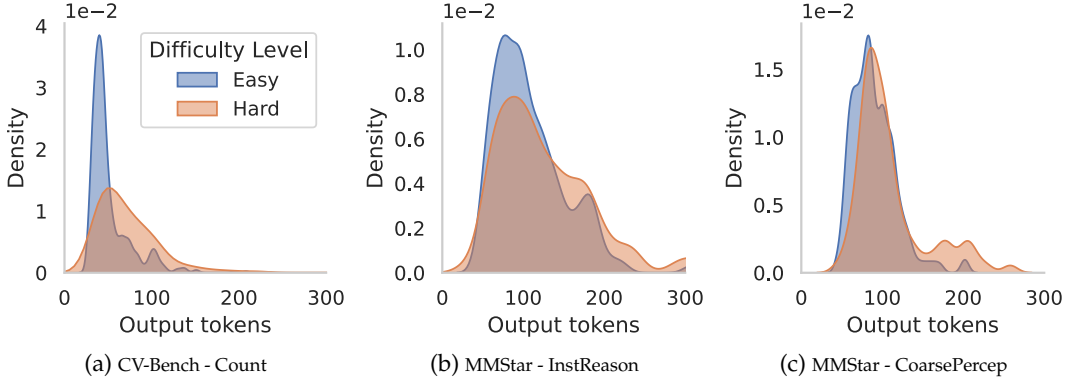


Figure 4: **Response lengths vs. question difficulties.** We analyze the responses of the VLM fine-tuned on LongPerceptualThoughts via DPO. Interestingly, we find that the model finetuned in our data naturally allocates more test-time compute for hard questions. We follow [Lightman et al. \(2024\)](#); [Snell et al. \(2025\)](#) and determine question complexity using rollouts on the base model.

G Qualitative dataset example

We provide an example of our dataset in Fig. 11.

```
System: You are an assistant that converts image descriptions to
↳ multi-choice visual questions.
User: Task:
You are given a detailed description of an image. Your goal is to
↳ generate diverse vision-centric, detailed questions that require a
↳ careful examination of the image for subtle visual details. Each
↳ question should be answerable in a brief sentence or single phrase.
Instructions:
- Focus on Visual Detail:
  - Ask questions that require examining fine details such as textures,
    ↳ patterns, and small or hidden elements.
  - Encourage the reader to analyze spatial relationships like object
    ↳ overlap, perspective, and layout.
  - Include aspects of lighting, shadows, and color gradients that
    ↳ affect the image's mood and depth.
- Comprehensive Coverage:
  - Ensure that the questions, as a group, address the majority of
    ↳ important details mentioned in the image description.
- Design for Multiple-Choice:
  - For each question, provide 4 answer options labeled A, B, C, and D.
  - Include one correct answer and three plausible distractors.
- Encourage Careful Inspection:
  - Design each question so that it cannot be answered without a close,
    ↳ careful visual inspection of the image.
  - Avoid generic or overly broad questions; each should target specific
    ↳ visual cues mentioned or implied in the description.
- Clarity, Specificity, and Brevity in Answers:
  - Formulate questions that are clear and focused on visual elements.
  - Ensure each question is detailed enough to challenge the reader to
    ↳ look beyond the surface.
  - Avoid questions that can be answered with general knowledge or
    ↳ assumptions.
  - Each question should be answerable in a brief sentence or even a
    ↳ single phrase.
- Structured Output:
  - Provide the questions in a numbered list.
  - Example Format: 1. <question> question here </question> <choices>
    ↳ (A) ... (B) ... (C) ... (D) ... </choices> <answer> short answer
    ↳ here </answer>

Image Description:
{{ image descriptions }}
Assistant:
```

Figure 5: Text prompt converting descriptions to multi-choices questions.

User: You are given a text containing multiple multi-choice questions.

- Each question includes a question statement, several choices, and an
- answer. Your task is to reformat the text so that each multi-choice
- question follows the structure below:

1. <question> question text here </question> <choices> (A) choice A text
 → (B) choice B text (C) choice C text (D) choice D text </choices>
 → <answer> answer text here </answer>

Please ensure that:

- Each question is numbered sequentially (e.g., 1., 2., 3., ...).
- The question portion is enclosed within the `<question>` tags.
- All answer options are enclosed within the `<choices>` tags, with each
 → option clearly labeled (A), (B), (C), (D).
- The answer is enclosed within the `<answer>` tags.
- The original content is preserved, but any formatting issues are
 → corrected according to the template above.

Here is the original content: `{{ raw outputs in Fig.5 }}`

Assistant:

Figure 6: Text prompt to parse the response of Fig. 5 to multi-choices questions.

User: You are a large language model that answers visual questions by

- generating a vivid mental image from a text description. Given a
- visual question along with an image description, create a detailed
- internal visualization of the image. Then, use this mental image to
- spatially reason through and answer the question.

- After building the mental image from the text description, you should
 → not explicitly referencing the text description in your internal
 → reasoning. e.g., Avoid saying "The description states ..." within
 → <think>...</think> block.
- Ensure your reasoning is logically sound and leads coherently to the
 → final answer. The steps you follow should clearly support the
 → conclusion you reach.
- Please provide your answer as (X), where X is the letter of the correct
 → option.
- Enclose your final answer within <answer> and </answer> tags.

Assistant: <think> `{{ simple CoT }}` `{{ cognitive phrase }}`

Figure 7: Text prompt to generate R1 thoughts.

System: A conversation between User and Assistant. The user asks a visual

- question, and the Assistant solves it. The answer are enclosed within
- <answer> </answer> tags, respectively, i.e., <answer> answer here
- </answer>. Format the answer with the letter of the correct option in
- parentheses.

User: <image>Provide a detailed description of the image.

Assistant: `{{ image descriptions }}`

Figure 8: Training prompt for training on DOCCI descriptions.

System: A conversation between User and Assistant. The user asks a visual
→ question, and the Assistant solves it. The answer are enclosed within
→ <answer> </answer> tags, respectively, i.e., <answer> answer here
→ </answer>. Format the answer with the letter of the correct option in
→ parentheses.
User: <image> {{ question }}
Select from the following choices.
{{ choices }}
Assistant:

Figure 9: Evaluation prompt for direct answers.

System: A conversation between User and Assistant. The user asks a visual
→ question, and the Assistant solves it. The assistant first thinks
→ about the reasoning process in the mind and then provides the user
→ with the answer. The reasoning process and answer are enclosed within
→ <think> </think> and <answer> </answer> tags, respectively, i.e.,
→ <think> reasoning process here </think> <answer> answer here
→ </answer>. Format the answer with the letter of the correct option in
→ parentheses.
User: <image>{{ question }}
Select from the following choices.
{{ choices }}
Assistant:

Figure 10: Evaluation prompt for thoughts and answers.



Question: How is the smaller tennis racket decorated within its strings? Select from the following choices.

- (A) A smiley face
 - (B) A floral design
 - (C) A logo
 - (D) A pattern of stars
- % Correct answer: (A) A smiley face

Simple CoT

<think>

Upon examining the smaller tennis racket, I can see that there is a
→ distinct pattern of stars within its strings. This pattern is
→ consistent across the strings, making it clear what the design is.
</think>

<answer> (D) </answer>

Ours

<think>

Upon examining the smaller tennis racket, I can see that there is a
→ distinct pattern of stars within its strings. This pattern is
→ consistent across the strings, making it clear what the design is.
→ Wait, no, hold on. Actually, the image shows a smiley face inside the
→ strings, but it's upside down. So that must be the decoration.

Wait, maybe I got confused. Let me double-check. The smaller racket has a
→ smiley face printed in black within the strings. So the correct
→ answer should be A smiley face. But wait, the options include both A
→ and C. Hmm, could it be a logo? No, because the Wilson racket has a
→ logo, but the smaller one has a smiley face. So the decoration is a
→ smiley face, which is option A.

</think>

<answer> (A) </answer>

Figure 11: Qualitative example of our LongPerceptualThoughts dataset performing verification.