

## 977 A Experiment Settings

### 978 A.1 Pipeline Settings

#### 979 A.1.1 Model Hyperparameter Settings

980 Our base model is Qwen2.5VL 7B[8], which supports dynamic resolution for input images. In all  
981 experiments, we constrained the pixel dimensions of each image to a minimum of 3136 pixels and  
982 a maximum of 1605632 pixels. Because the value of the bounding box is related to the number of  
983 pixels in the input image, the setting of the range of pixels needs to be unified.

#### 984 A.1.2 Zoom Scaling Rule

985 In our pipeline, when a region is selected for closer inspection (e.g., via a "Crop" operation), a zoom  
986 operation is applied. The scaling factor for this zoom, denoted as  $scale$ , is determined dynamically  
987 based on the relative area of the selected bounding box ( $A_{\text{bbox}}$ ) compared to the area of the original  
988 image ( $A_{\text{orig}}$ ). Let  $r = \frac{A_{\text{bbox}}}{A_{\text{orig}}}$  be this area ratio. The scale is calculated using the following piecewise  
989 function:

$$scale = \begin{cases} 2.0, & \text{if } r < 0.125 \\ 1.0, & \text{if } r \geq 0.5 \\ 2.0 - \frac{r - 0.125}{0.375}, & \text{otherwise} \end{cases} \quad (4)$$

990 This rule implies that smaller selected regions (smaller  $r$ ) are scaled up more significantly (up to a  
991 factor of 2.0), while larger regions (larger  $r$ ) are scaled up less, or not at all if they already occupy a  
992 substantial portion of the original image. The intermediate case provides a linear interpolation of the  
993 scaling factor.

### 994 A.2 Training Setting for Supervised Fine-tuning Stage

995 In the supervised fine-tuning stage, we used the complete VLIR dataset. Our experiments were con-  
996 ducted on 4 NVIDIA A100 GPUs, each equipped with 80GB of memory, leveraging DeepSpeed[41]  
997 for efficient training. We used a batch size of 2 with a gradient accumulation of 8, a learning rate of  
998  $2 \times 10^{-7}$ , and trained for 3 epochs. During this phase, the vision encoder and MLP projector were  
999 frozen, and only the Large Language Model (LLM) component was trained.

### 1000 A.3 Training Setting for R-GRPO Stage

1001 For the R-GRPO stage, we sampled approximately 5,000 data points from TextVQA[47], GQA[17],  
1002 VSR[25], DocVQA[32] and M<sup>3</sup>CoT[9] datasets. Regarding the hyperparameters for the GRPO  
1003 formulation(2), we set  $M = 5$ . Following the experience of related studies, we set  $\beta = 0.0$ , i.e., we  
1004 eliminate the KL divergence constraint.

1005 Our experiments for R-GRPO were performed on 6 NVIDIA A100 GPUs, each with 80GB of  
1006 memory, also utilizing DeepSpeed[41]. The batch size per device was set to 1, with a gradient  
1007 accumulation of 16. The learning rate was  $1 \times 10^{-6}$ , and training continued for 300 steps. We employ  
1008 a rule-based reinforcement learning approach, where the correctness of the final answer was judged  
1009 using an exact match criterion. Similar to the supervised fine-tuning stage, the vision encoder and  
1010 MLP projector were frozen, and only the LLM component was trained.

## 1011 B Prompt Templates for VLIR Dataset Construction and Filtering

### 1012 B.1 Data Construction Prompts

1013 Given an {image, question, answer} triplet, the following prompt was used to construct the interleaved  
1014 visual-linguistic chain of thought:

1015 

1016 You are performing "Multimodal Interleaved Reasoning". During the thinking

```

1017 process, you need to keep an eye on the visual cues in the original image,
1018 find regions of the image that help answer the question, and use the "Crop"
1019 tool to crop and zoom in for detailed analysis.
1020 When using the tool, you must output a JSON object in the following format:
1021 {"bbox_2d": [x1, y1, x2, y2]}
1022 Ensure that you "Crop" at least once.
1023 Continue thinking after each operation until you reach the final answer.
1024 Output the thinking process within a pair of <think> </think> tags and then
1025 output the final answer within a pair of <answer> </answer> tags.
1026 {question}

```

Listing 1: Prompt for dataset construction.

1028 Given an {image, question, answer, bounding box annotation} quadruplet, the following prompt was  
 1029 used:

```

1030 I will now provide you with an image, a question, and a "Crop" operation
1031 string. Your task is to write the reasoning process used to answer the
1032 question as instructed. During the reasoning process, the respondent
1033 utilizes a "Crop" operation to assist with reasoning. The format of
1034 the operation is as follows:
1035 {"bbox_2d": [x1, y1, x2, y2]}
1036 This bounding box indicates the key region that needs to be focused
1037 on to correctly answer the question.
1038 You must think step by step from the perspective of the respondent,
1039 using the "Crop" operation at appropriate moments in your reasoning
1040 process to eventually reach the correct answer. Important notes:
1041 1. You must not modify the content or format of the "Crop" operation
1042 in any way.
1043 2. In a real setting, the respondent only has access to the image and
1044 the question. This bounding box indicates the area where the correct
1045 answer information is located. In this task, they are provided to ensure
1046 the correctness of your reasoning process. When writing the reasoning,
1047 pretend you are the respondent who independently identifies when to use
1048 the "Crop" operation and how to reach the answer step by step.
1049 3. Make sure the reasoning is fluent, logical, and concise.
1050 4. Format of the reasoning process: <think>...</think><answer>...</answer>
1051
1052 Here is an example:
1053 Question: Are there any black numbers or letters?
1054 "Crop" operation: {"bbox_2d": [247, 384, 307, 444]}
1055 Reasoning: <think>
1056 Step 1: To determine if there are black numbers or letters, I need to
1057 focus on the text visible in the image. The dog is wearing a heart-shaped
1058 tag that has some text on it. I will crop and zoom in on the tag for a
1059 closer look at the text details. {"bbox_2d": [247, 384, 307, 444]}
1060 Step 2: After cropping, I can see that the letters "G PLUS" are in red,
1061 and the numbers "6 223 13" are also in red. There are no black numbers
1062 or letters on the tag. Review the rest of the image, there are no black
1063 numbers or letters either.</think>
1064 <answer>no</answer>
1065
1066 Question:{question}
1067 "Crop" operation:{crop}
1068 Now Output the reasoning process:
1069

```

Listing 2: Prompt for dataset construction.

## 1071 B.2 Data Filtering Prompts

1072 The prompt for assessing the recognizability of the cropped images is as follows:

1073

1074 You need to determine whether the content in a picture is a complete and  
 1075 semantically meaningful visual unit. Please look carefully at this cropped  
 1076 image and determine whether it contains a recognizable object, block of text,  
 1077 or specific part of a diagram. If it is recognizable, answer 'yes'; if not,  
 1078 answer 'no'.  
 1079 Now output 'yes' or 'no' directly.

Listing 3: Prompt for assessing cropped image recognizability.

1081 The prompt for assessing the quality of the reasoning process is as follows:

1082 You need to make an in-depth assessment of this reasoning process. First,  
 1083 determine whether its logic is rigorous and whether each step of reasoning leads  
 1084 naturally and smoothly to the next; second, check whether the reasoning process  
 1085 progresses gradually towards arriving at the final answer; and lastly, check  
 1086 whether there is any false information or repetitive redundancy in the text  
 1087 that is not relevant to the reasoning. If this textual reasoning meets the  
 1088 requirements in terms of logic, advancement and content streamlining, output  
 1089 'yes'; whenever one of these is not met, output 'no'.  
 1090 {question}  
 1091 {ground-truth answer}  
 1092 {reasoning process}  
 1093 Now output 'yes' or 'no' directly.

Listing 4: Prompt for assessing reasoning process quality.

## 1096 C Demonstrations for VLM-R<sup>3</sup>

1097 This section provides qualitative examples illustrating the capabilities of VLM-R<sup>3</sup>.

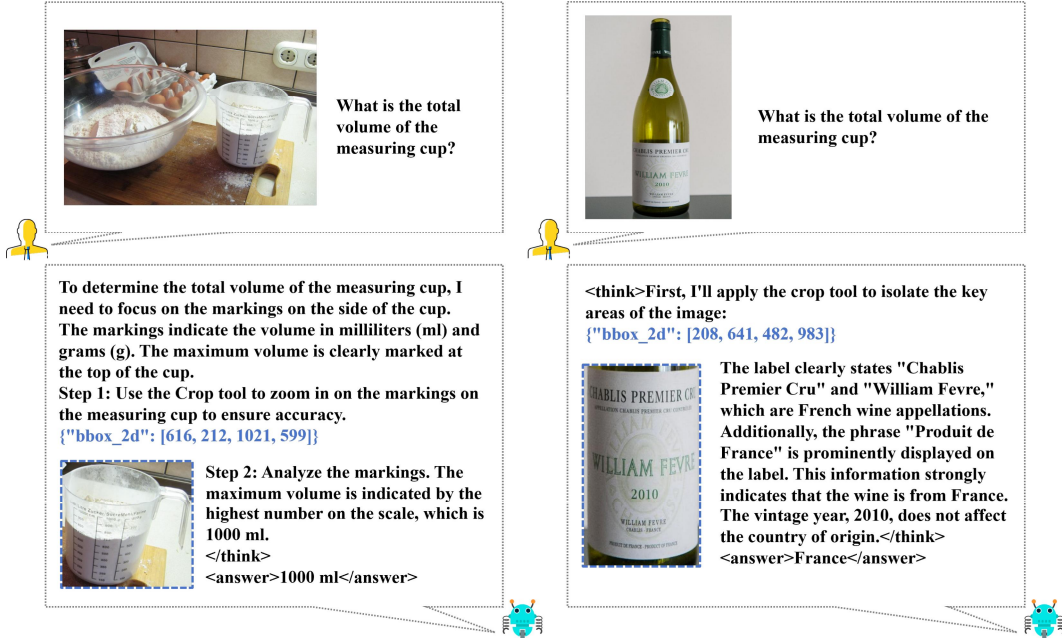


Figure 5: This figure presents how VLM-R<sup>3</sup> captures details in a natural image and reasons about them.

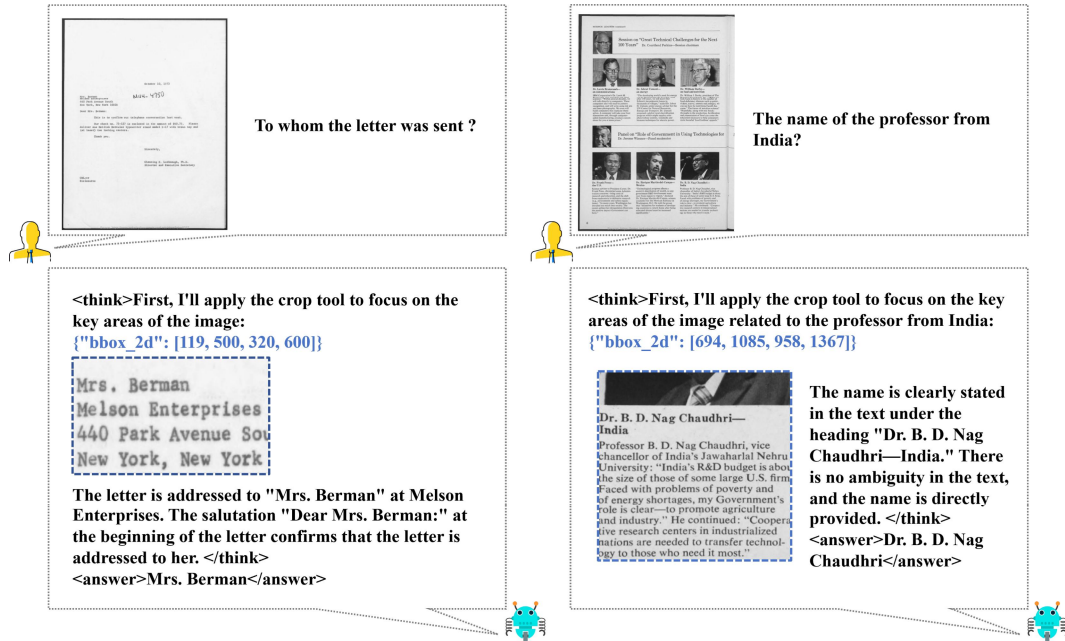


Figure 6: This figure presents how VLM-R<sup>3</sup> gets the text details from the document image.



Figure 7: This figure presents how VLM-R<sup>3</sup> iteratively tracks multiple visual cues in an image.

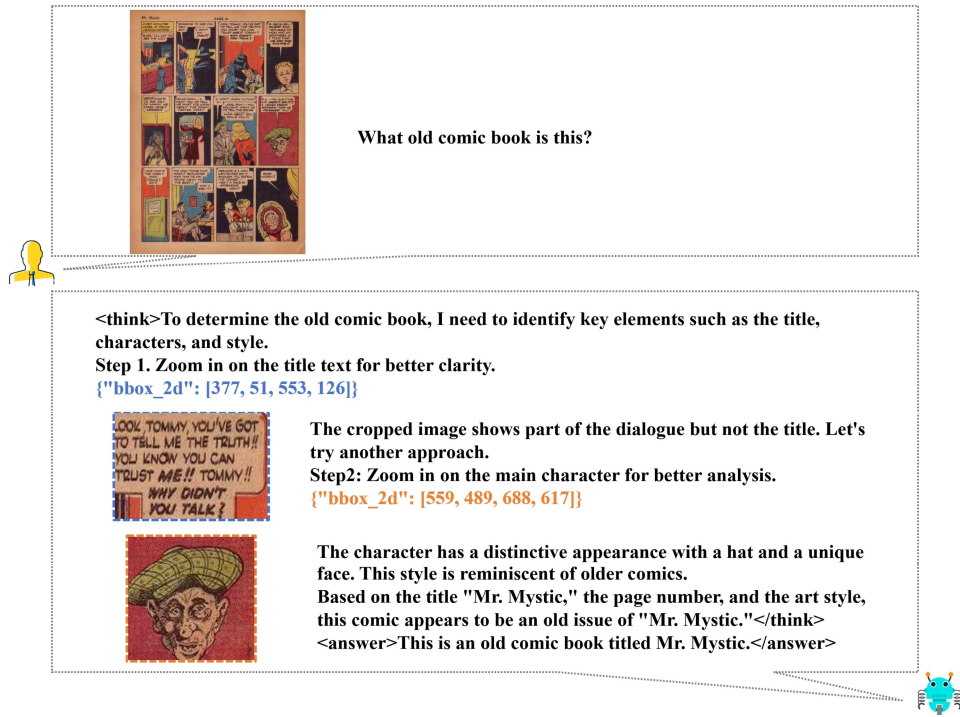


Figure 8: This figure presents how VLM-R<sup>3</sup> performs complex interleaved text-image CoT reasoning.

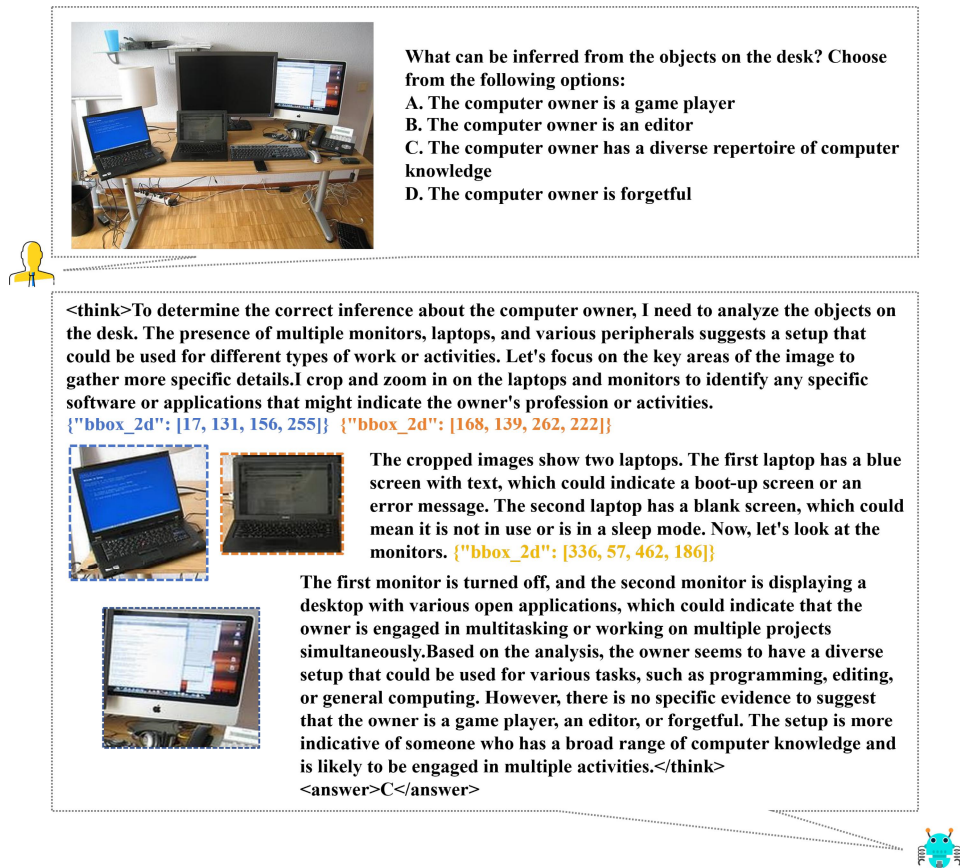


Figure 9: This figure presents how VLM-R<sup>3</sup> performs complex interleaved text-image CoT reasoning.