

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 AGENTIC JIGSAW INTERACTION LEARNING FOR EN- HANCING VISUAL PERCEPTION AND REASONING IN VISION-LANGUAGE MODELS

Anonymous authors

Paper under double-blind review

ABSTRACT

Although current large Vision-Language Models (VLMs) have advanced in multimodal understanding and reasoning, their fundamental perceptual and reasoning abilities remain limited. Specifically, even on simple jigsaw tasks, existing VLMs perform near randomly, revealing deficiencies in core perception and reasoning capabilities. While high-quality vision-language data can enhance these capabilities, its scarcity and limited scalability impose significant constraints. To address this, we propose **AGILE**, an Agentic jiGsaw Interaction Learning for Enhancing visual perception and reasoning in VLMs. AGILE formulates jigsaw solving as an interactive process, enabling the model to progressively engage with the environment. At each step, the model generates executable code to perform an action based on the current state, while the environment provides fine-grained visual feedback to guide task completion. Through this iterative cycle of observation and interaction, the model incrementally improves its perceptual and reasoning capabilities via exploration and feedback. Experimental results show that AGILE not only substantially boosts performance on jigsaw tasks of varying complexity (e.g., increasing accuracy from 9.5% to 82.8% under the 2×2 setting) but also demonstrates strong generalization across 9 general vision tasks, achieving an average improvement of 3.1%. These results indicate notable enhancements in both perceptual and reasoning abilities. This work opens a new avenue for advancing reasoning and generalization in multimodal models and provides an efficient, scalable solution to the scarcity of multimodal reinforcement learning data. Code and datasets will be released soon.

1 INTRODUCTION

Large Vision-Language Models (VLMs) have recently achieved remarkable success across a wide range of multimodal tasks, including image captioning (Chen et al., 2024a;c; Li et al., 2024b), visual question answering (Bai et al., 2025; Zhu et al., 2025; Hurst et al., 2024; Comanici et al., 2025), and scene understanding (Wang et al., 2025b;a). By effectively associating visual and textual information, these models exhibit strong multimodal perception and reasoning capabilities. However, their performance remains severely limited on tasks that require comprehensive visual understanding and structured reasoning. In particular, we find that current VLMs often perform near random even on relatively simple 2×2 jigsaw tasks (Carlucci et al., 2019; Chen et al., 2023), which demand both perceptual accuracy and logical inference. This suggests that existing pretraining and fine-tuning strategies are insufficient for developing robust perceptual and reasoning abilities.

Recent studies have explored reinforcement learning (RL) (Schulman et al., 2017; Rafailov et al., 2023; Shao et al., 2024) as a way to enhance these capabilities by enabling models to learn through interaction, trial-and-error, and feedback. While RL-based approaches show promise, they are fundamentally constrained by the scarcity and limited scalability of high-quality vision-language RL data. Current methods for constructing multimodal RL datasets (Huang et al., 2025; Yang et al., 2025; Lu et al., 2023) generally fall into two categories: human expert supervision and automated synthesis. Human-supervised datasets are either prohibitively expensive or too small in scale for large-scale training, while automated approaches that rely on closed-source models suffer from lim-

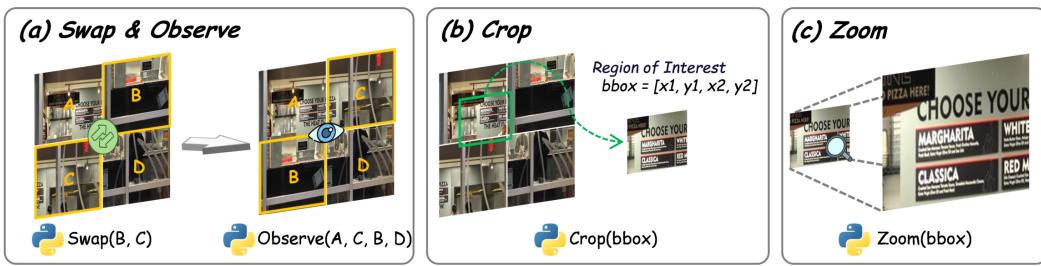


Figure 1: **Description of the action space.** (a) illustrates swapping two jigsaw pieces and observing the updated jigsaw state; (b) shows cropping a specific region of the jigsaw for closer inspection; and (c) depicts zooming into a selected area to examine fine-grained details.

ited quality, capability constraints, and substantial API costs. These challenges collectively hinder the development of VLMs with strong reasoning and generalization abilities.

To address these limitations, we propose **AGILE**, an Agentic jiGsaW Interaction Learning framework for Enhancing visual perception and reasoning in VLMs. Our approach leverages the structured nature of the jigsaw puzzle as a proxy task for perception and reasoning, modeling the jigsaw-solving process as a progressive interaction between the model and its environment. At each step, the model generates Python code to perform actions within a well-defined action space (swapping two image tiles, observing the current jigsaw state, cropping regions for detailed observation, or zooming in for fine-grained analysis). This interaction-driven process enables the model to iteratively refine its perceptual and reasoning capabilities while receiving explicit feedback at every step. By simulating the step-by-step dynamics of jigsaw solving, the model is encouraged to capture structural relationships among visual components and acquire more robust perception and reasoning skills.

A key feature of our approach is the use of code and rule-based data generation, which offers two major advantages. First, the difficulty of the jigsaw task can be precisely controlled by adjusting factors such as the number of correctly placed tiles in the initial state and the overall jigsaw size. Second, because the ground-truth solution is inherently available, the synthetic dataset can be scaled to arbitrary sizes with strict supervision, overcoming the data scarcity inherent in human-annotated or closed-source datasets. The combination of interactive training and scalable data generation yields an efficient and effective framework for advancing visual perception and reasoning in VLMs.

We validate the effectiveness of our approach through comprehensive experiments. Our method substantially improves performance across jigsaw tasks of varying complexity, for instance, raising accuracy from 9.5% to 82.8% on the 2×2 setting, while also demonstrating strong generalization to a wide range of vision tasks, including high-resolution image understanding, real-world scene analysis, fine-grained recognition, visual reasoning, and hallucination benchmarks. Moreover, we show that scaling the training data further boosts performance, and that under equal data budgets, jigsaw-based training achieves results comparable to or even surpassing those obtained with general QA data. These findings underscore the potential of jigsaw tasks in addressing the scarcity of multimodal RL data. By combining interactive training with scalable data generation, our method significantly enhances both perceptual and higher-order reasoning abilities, pointing to a promising new direction for advancing VLMs. Our main contributions are as follows:

- We introduce AGILE, an agentic jigsaw interaction learning framework that formulates jigsaw solving as a stepwise interactive process, thereby driving incremental improvements in both visual perception and reasoning capabilities of VLMs.
- We present a scalable jigsaw-based data generation method that yields high-quality multimodal reinforcement learning datasets with controllable difficulty, providing an efficient solution to the current shortage of high-quality training data.
- Extensive experiments demonstrate that our approach substantially improves performance on jigsaw tasks of varying complexity while exhibiting strong generalization across diverse vision benchmarks, highlighting its effectiveness in enhancing both perception and reasoning in VLMs.

108 **2 RELATED WORK**

110 **Reinforcement Learning for Vision Language Models.** Reinforcement learning (RL) (Schulman
 111 et al., 2017; Rafailov et al., 2023; Shao et al., 2024) has been widely applied to improve reasoning in
 112 language models, as exemplified by the success of DeepSeek-R1 (Guo et al., 2025) in mathematical
 113 reasoning. Building on this progress, recent studies have extended RL to VLMs, with rule-based
 114 RL in multimodal domains emerging as a promising direction. For perception enhancement, R1-V
 115 (Chen et al., 2025b) applies RL to object counting, while Perception-R1 (Yu et al., 2025) leverages
 116 object matching and IoU as reward signals to improve grounding. DeepEyes (Zheng et al., 2025)
 117 shows how RL can encourage models to invoke visual tools, thereby expanding perceptual abilities.
 118 For reasoning, MMEureka (Meng et al., 2025) demonstrates the effectiveness of rule-based RL
 119 in mathematical tasks. From a data perspective, Vision-R1 (Huang et al., 2025) and R1-OneVision
 120 (Yang et al., 2025) convert visual information into textual representations to build multimodal chain-
 121 of-thought (CoT) datasets that support stronger reasoning. Despite these advances, the lack of scal-
 122 able, high-quality RL datasets remains a fundamental bottleneck, severely limiting further improve-
 123 ments in both perception and reasoning for VLMs.

124 **Enhancing Generalization via Proxy Tasks.** Rule-based reinforcement learning (RL) has shown
 125 promise but typically requires large amounts of high-quality, verifiable data. Logic-RL (Xie et al.,
 126 2025a) addressed this by introducing the “Knights and Knaves” (K&K) puzzle, which offers con-
 127 trollable difficulty and algorithmically generated ground truth, enabling models to acquire reasoning
 128 skills transferable to mathematical tasks. Enigmata (Chen et al., 2025a) extended this approach to a
 129 broader set of text-based puzzles (e.g., cryptographic, arithmetic, logical), demonstrating that solv-
 130 ing such tasks with RL improves general reasoning without external data. RPT (Dong et al., 2025)
 131 further reframed next-token prediction as a reasoning task, using verifiable rewards to enhance pre-
 132 dictive ability and strengthen the basis for reinforcement fine-tuning. Inspired by these proxy-task
 133 successes in LLMs, recent work has begun exploring similar ideas for VLMs. Code2Logic (Tong
 134 et al., 2025) and ViGaL (Xie et al., 2025b) leverage code-synthesized games to improve mathe-
 135 matical reasoning, while ViCrit (Wang et al., 2025d) enhances perceptual robustness by reinforcing
 136 models to detect hallucinated entities via modified captions. Jigsaw-R1 (Wang et al., 2025e) pro-
 137 posed a puzzle-based RL paradigm, but due to training limitations, its models still performed poorly
 138 even on 2×2 jigsaw puzzles, failing to fully exploit the proxy-task benefits. To address this gap, we
 139 propose a agentic jigsaw interaction learning framework, which casts jigsaw solving as an interac-
 140 tive process between the model and environment. This enables iterative refinement of perception and
 141 reasoning with explicit stepwise feedback, allowing the model to better capture visual relationships
 142 and develop stronger perceptual and reasoning skills.

143 **3 METHOD**

144 In this section, we provide a comprehensive description of our proposed agentic jigsaw interaction
 145 learning framework. We first introduce the jigsaw task and describe how the model interacts with the
 146 environment to accomplish it (Sec. 3.1), then present the construction of the training data (Sec. 3.2).
 147 Finally, we outline the training paradigm, including both the cold-start stage and the reinforcement
 148 learning stage (Sec. 3.3).

149 **3.1 JIGSAW TASK AND ENVIRONMENT INTERACTION**

150 **Jigsaw Task.** Given an input image, we partition it into an $m \times m$ grid of jigsaw pieces, where task
 151 difficulty can be flexibly adjusted by varying m . If the image height or width is not divisible by m ,
 152 the image is resized so that its dimensions are exact multiples of the grid. The grid is then randomly
 153 shuffled, and each piece is assigned an index in row-major order ranging from 1 (top-left) to m^2
 154 (bottom-right). The shuffled configuration is denoted as

$$I_{\text{Shuffle}} = \{I_1, I_2, \dots, I_{m^2}\}, \quad (1)$$

155 where each I_k corresponds to one jigsaw block. Since the shuffling process is explicitly recorded
 156 during data generation, we can recover the ground-truth jigsaw layout as

$$I_{\text{GT}} = \{I_{\pi(1)}, I_{\pi(2)}, \dots, I_{\pi(m^2)}\}, \quad (2)$$

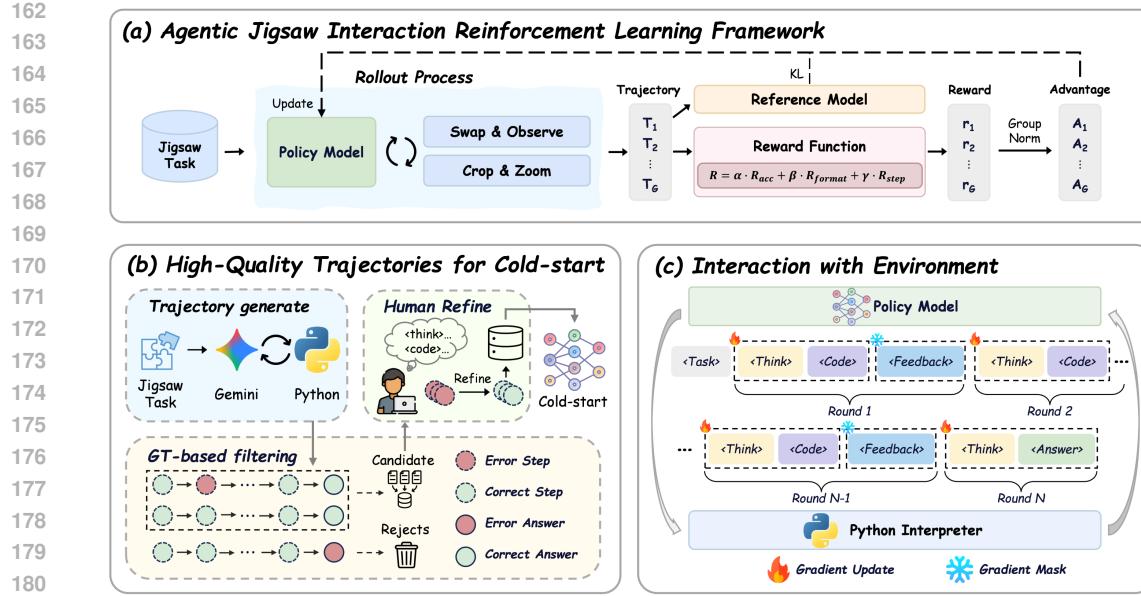


Figure 2: **Overview of the AGILE framework.** (a) depicts the interaction process between the model and the external environment, together with the implementation of the GRPO algorithm; (b) shows the collection of high-quality jigsaw trajectory data; and (c) illustrates the model–environment interaction during the jigsaw rollout process.

where π is a permutation over $\{1, 2, \dots, m^2\}$. During jigsaw solving, the model will maintain a current state

$$I_{State} = \{I_{\pi^*(1)}, I_{\pi^*(2)}, \dots, I_{\pi^*(m^2)}\}, \quad (3)$$

where π^* denotes the current arrangement of pieces. At each step, the model iteratively swaps two pieces, observes the resulting configuration, and aims to reconstruct the ground-truth layout I_{GT} .

Environment Interaction. The model interacts with the environment throughout the jigsaw-solving process in an iterative manner. As illustrated in Figure 1, we predefined a set of API for model–environment interactions in Python. The model expresses its actions by generating Python code, which is executed by the environment to produce the corresponding resulting image. The model then uses these results to reason further and decide on subsequent actions, repeating this process iteratively until the jigsaw is completed. Specifically, at each step, the model can perform the following actions:

- **Swap:** Given the current jigsaw state I_{State} , the model calls the Swap function to exchange the positions of any two jigsaw pieces.
- **Observe:** Given I_{State} , the model calls the Observe function to obtain the current jigsaw progress I_{Obs} , which is then used to determine the next action.
- **Crop & Zoom:** Given I_{Obs} , the model crops and zooms into a local region to inspect finer details and inform subsequent actions.

To better illustrate the overall procedure, the entire rollout process is presented in Algorithm 1.

3.2 DATA CONSTRUCTION

In Section 3.1, we have introduced the jigsaw task and described how the model interacts with the environment to complete it. Based on this task design, we observe that the model (e.g., Qwen-2.5VL-7B) exhibits limited foundational capabilities, manifested in poor instruction-following and incorrect Python code generation, which hinders proper interaction with the environment and introduces substantial training noise. Directly applying reinforcement learning (RL) thus results in low learning efficiency. To bridge this gap and more effectively train our model, we employ Gemini

216 **Algorithm 1** Interactive Jigsaw-Solving Procedure of VLMs with the Environment

217 **Input:** Query x , I_{Shuffle} , Policy model π_θ , External environment \mathcal{V} , Maximum iteration steps T .
 218 **Output:** Final interaction trajectory y .
 219 1: Initialize trajectory $y \leftarrow \emptyset$, step counter $t \leftarrow 0$.
 220 2: **while** $t < T$ **do**
 221 3: Sample response $y_t \sim \pi_\theta(\cdot | x, I_{\text{Shuffle}}, y)$.
 222 4: Append assistant response y_t to trajectory: $y \leftarrow y + y_t$.
 223 5: **if** <code> </code> detected in y_t **then**
 224 6: Parse the Python code (e.g., Swap, Observe, Crop, Zoom), execute it in the environment \mathcal{V} , and obtain
 225 feedback O_t .
 226 7: **else if** <answer> </answer> detected in y_t **then**
 227 8: **return** final trajectory y .
 228 9: **end if**
 229 10: Append environment feedback O_t as user input: $y \leftarrow y + O_t$.
 230 11: Update step counter $t \leftarrow t + 1$.
 231 12: **end while**
 232 13: **return** final trajectory y .

233 2.5 Pro¹ to collect expert trajectories for the cold-start phase, endowing Qwen-2.5VL-7B with ba-
 234 sis interactive jigsaw-solving capabilities. For this purpose, we design structured prompts to guide
 235 Gemini 2.5 Pro in interacting with the environment to complete the jigsaw tasks (The prompt tem-
 236 plates are provided in Appendix B.).

237 To further ensure data quality, we apply an additional filtering process: first, selecting samples
 238 where Gemini’s outputs match the ground truth, and then manually verifying each interaction step
 239 in the correct samples to ensure rationality and consistency. To ensure that our model can perform
 240 diverse jigsaw reasoning actions during reinforcement learning, we carefully balance the training
 241 data. Specifically, trajectories are balanced with respect to both the number of steps (4–8) and
 242 the types of actions involved (e.g., Swap, Observe, Crop, Zoom). This design ensures that the
 243 model is sufficiently exposed to the full action space and learns to handle various interactions with
 244 the environment effectively. Overall, we curate a dataset comprising 1.6K high-quality reasoning
 245 trajectories for the jigsaw task.

246
 247 3.3 TRAINING PARADIGM: COLD-START AND REINFORCEMENT LEARNING
 248

249 **Cold-start.** During the cold-start phase, we leverage the 1.6K high-quality jigsaw-solving trajec-
 250 tories constructed in Section 3.2 to equip the model with basic instruction-following and Python code
 251 generation capabilities, ensuring effective interaction with the environment.

252 **Reinforcement Learning.** Our framework adopts Group Relative Policy Optimization (GRPO),
 253 which utilizes the average reward of multiple sampled outputs as a baseline, instead of relying on a
 254 learned value function. The policy model is optimized by maximizing the following objective:
 255

$$256 \quad \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D}, \{y_i\}_{i=1}^G \sim \pi_{\text{old}}(\cdot | x; \mathcal{V})} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{\sum_{t=1}^{|y_i|} I(y_{i,t})} \sum_{t=1:I(y_{i,t})=1}^{|y_i|} \min \left(\frac{\pi_\theta(y_{i,t} | x, y_{i,<t}; \mathcal{V})}{\pi_{\text{old}}(y_{i,t} | x, y_{i,<t}; \mathcal{V})} \hat{A}_{i,t}, \right. \right. \\ 257 \quad \left. \left. \text{clip} \left(\frac{\pi_\theta(y_{i,t} | x, y_{i,<t}; \mathcal{V})}{\pi_{\text{old}}(y_{i,t} | x, y_{i,<t}; \mathcal{V})}, 1 - \epsilon, 1 + \epsilon \right) \hat{A}_{i,t} \right) \right] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \quad (4)$$

258 where the rollout module samples a group of trajectories $\{y_1, y_2, \dots, y_G\}$ from the old policy π_{old}
 259 for each input question x through interaction with the external environment \mathcal{V} . The advantage term
 260 $\hat{A}_{i,t}$ is computed based on the relative rewards of outputs within each group. Our reward system
 261 comprises three components: an accuracy reward, a format reward, and a step reward. The total
 262 reward is computed as the sum of these components.

263
 264 ¹The version is Gemini-2.5-Pro-Preview-05-06.
 265

270 • **Accuracy Reward:** We compare the model’s generated answer I_{Answer} with the ground truth
 271 I_{GT} . If all jigsaw image blocks are correctly placed, the accuracy reward is 1; otherwise, it is 0.
 272

273 • **Format Reward:** The model receives a reward of 1 if its output follows the required structured
 274 format, with the reasoning process, code, and final answer correctly enclosed within the `<think>`,
 275 `<code>`, and `<answer>` tags, respectively.

276 • **Step Reward:** We encourage the model to complete the jigsaw in as few steps as possible. For
 277 a 2×2 jigsaw, if each step swaps any two image blocks, at most three steps are theoretically
 278 required to place all blocks correctly. Excessive actions may lead the model to perform invalid
 279 steps, reducing the proportion of effective perception and reasoning. Furthermore, to prevent the
 280 model from prematurely hacking the step reward during early RL training, the step reward is applied
 281 only when the jigsaw is correctly completed. If the jigsaw is incorrect, the model is penalized by
 282 assigning the maximum step penalty. Formally, the step reward is defined as:
 283

$$R_{\text{step}} = \lambda \cdot \left(\mathbb{I}_{\{R_{\text{acc}}=1\}} \cdot step_{\text{num}} + \mathbb{I}_{\{R_{\text{acc}}=0\}} \cdot step_{\text{max}} \right), \quad (5)$$

286 where $step_{\text{num}}$ is the number of steps the model actually used to complete the jigsaw, $step_{\text{max}}$ is
 287 the maximum allowed steps, λ denotes the step penalty coefficient, which is set to -0.05 , and $\mathbb{I}_{\{\cdot\}}$
 288 is the indicator function. The final reward formulation is shown in Equation 6:

$$R = \alpha \cdot R_{\text{acc}} + \beta \cdot R_{\text{format}} + \gamma \cdot R_{\text{step}} \quad (6)$$

292 where the coefficients α , β , and γ weight the relative importance of accuracy, format, and step
 293 rewards, respectively. In our experimental setup, α , β , and γ are set to 0.8, 0.2, and 1.0, respectively.
 294

295 4 EXPERIMENT

297 In this section, we present a comprehensive experimental study. We first describe the implementation
 298 details, including datasets, training procedures, and inference settings (Sec. 4.1). We then evaluate
 299 the effectiveness of our agentic jigsaw interaction learning framework on both jigsaw-solving and
 300 general vision tasks (Sec. 4.2). Finally, we perform ablation studies to assess the impact of jigsaw
 301 data scale and to contrast the benefits of jigsaw data with those of general QA data (Sec. 4.3).
 302

303 4.1 IMPLEMENTATION DETAILS

305 **Datasets and Models.** Our training data consist of two components corresponding to the cold-start
 306 and reinforcement learning (RL) stages. In the cold-start stage, we employ 1.6K high-quality jigsaw
 307 puzzle trajectories collected in Sec. 3.2 to endow the model with basic interactive jigsaw-solving
 308 skills. In the RL stage, we construct a dataset of 15.6K images spanning diverse domains, including
 309 high-resolution visual search, OCR-based text recognition, real-world scenes, and structured dia-
 310 grams (see Appendix A for details). Each image is partitioned into 2×2 patches and randomly
 311 shuffled to ensure that all patches are initially misplaced. All experiments are conducted using
 312 Qwen2.5-VL-7B (Bai et al., 2025) as the base model.

313 **Training and Inference Setups.** We conduct supervised fine-tuning (SFT) on llama-factory (Zheng
 314 et al., 2024) and reinforcement learning (RL) training on verl (Sheng et al., 2024), both with full-
 315 parameter tuning. For inference and evaluation, we adopt VLMEvalKit (Duan et al., 2024) as the
 316 framework. **To ensure a fair comparison, all evaluations on general downstream benchmarks are**
 317 **conducted in a strict single-turn setting following VLMEvalKit, fully consistent with the evaluation**
 318 **protocol of all baseline models.** All experiments are performed on 8 NVIDIA A100 GPUs with
 319 80GB memory each. Detailed training hyperparameters are provided in the Appendix D.

320 4.2 MAIN RESULTS

322 **Significant Improvements on the Jigsaw Task.** To comprehensively evaluate performance on the
 323 jigsaw task, we curate a test set of 300 images spanning diverse scenarios, including high-resolution
 324 visual search, OCR-based text recognition, and real-world scenes. Each image is partitioned into

324
 325 **Table 1: Jigsaw Acc result.** LN indicates the difficulty level, where N denotes the initial number of
 326 correct pieces. A smaller N corresponds to a more scrambled jigsaw and higher difficulty. The best
 327 results are highlighted in **bold**, and the second-best results are underlined.

Model	2 × 2				3 × 3								
	L0	L1	L2	Avg.	L0	L1	L2	L3	L4	L5	L6	L7	Avg.
Random	4.5	3.7	4.2	4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
GPT-4o	38.7	37.7	47.0	41.1	1.0	1.3	2.7	3.0	3.0	7.0	7.7	13.7	4.9
Gemini-2.5-Pro	<u>43.3</u>	<u>46.3</u>	<u>49.7</u>	<u>46.4</u>	7.0	8.7	<u>9.7</u>	<u>10.0</u>	<u>11.0</u>	<u>15.0</u>	<u>23.0</u>	<u>32.0</u>	<u>14.6</u>
MiMo-VL-7B-RL	11.3	14.3	17.0	14.2	0.3	0.0	0.0	0.0	0.3	0.3	1.0	3.7	0.7
InternVL3-8B	3.7	3.7	5.0	4.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
InternVL3-78B	3.3	4.3	3.0	3.5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Qwen2.5VL-72B	22.7	24.3	35.3	27.4	0.3	1.3	2.3	2.7	2.0	3.0	6.7	12.7	3.9
Qwen2.5VL-7B	6.3	6.0	16.3	9.5	0.0	0.0	0.0	0.0	0.0	0.7	1.3	1.3	0.4
+Cold-Start	12.0	32.0	22.0	22.0	0.0	0.3	0.0	0.0	0.0	0.7	0.3	0.0	0.2
+RL	78.7	83.0	86.7	82.8	<u>5.0</u>	<u>7.0</u>	11.0	14.0	17.7	25.3	38.0	48.0	20.8

340
 341 **Table 2: Jigsaw Score result.** LN indicates the difficulty level, where N denotes the initial number
 342 of correct pieces. A smaller N corresponds to a more scrambled jigsaw and higher difficulty. The
 343 best results are highlighted in **bold**, and the second-best results are underlined.

Model	2 × 2				3 × 3								
	L0	L1	L2	Avg.	L0	L1	L2	L3	L4	L5	L6	L7	Avg.
Random	25.2	24.7	24.8	24.9	11.4	11.3	11.0	10.9	11.1	11.4	11.3	10.8	11.2
GPT-4o	54.8	59.1	<u>63.2</u>	<u>59.0</u>	25.9	27.3	34.1	36.3	42.3	48.0	54.4	64.0	41.5
Gemini-2.5-Pro	<u>55.3</u>	<u>60.1</u>	61.6	<u>59.0</u>	<u>31.9</u>	<u>31.8</u>	<u>35.7</u>	<u>39.4</u>	<u>43.1</u>	<u>48.7</u>	<u>61.8</u>	<u>68.6</u>	<u>45.1</u>
MiMo-VL-7B-RL	21.8	27.2	29.2	26.1	5.5	6.9	8.0	10.7	15.9	19.5	20.6	28.9	14.5
InternVL3-8B	24.6	24.3	25.3	24.7	7.5	10.1	15.7	16.8	20.3	25.1	28.2	31.7	19.4
InternVL3-78B	21.3	26.5	28.4	25.4	7.1	10.9	14.1	18.5	23.1	25.7	29.0	37.0	20.7
Qwen2.5VL-72B	40.7	46.1	56.1	47.6	21.7	25.6	30.8	33.4	34.7	40.0	46.3	55.2	36.0
Qwen2.5VL-7B	14.4	30.1	43.8	29.4	5.8	12.0	20.3	25.6	32.7	40.9	50.2	60.9	31.1
+Cold-Start	40.6	45.6	45.3	43.8	<u>6.5</u>	<u>6.9</u>	<u>7.8</u>	<u>10.2</u>	<u>10.4</u>	<u>13.2</u>	<u>15.7</u>	<u>17.0</u>	<u>11.0</u>
+RL	86.4	88.7	91.9	89.0	41.1	46.8	52.6	58.6	65.3	71.0	77.9	83.6	62.1

356 2×2 and 3×3 grids, and the performance is measured using two metrics: *Acc*, which equals
 357 1 only when all patches are correctly placed, and *Score*, defined as the ratio of correctly placed
 358 patches to the total number of patches. As shown in Table 1 and 2, the base model Qwen2.5-VL-
 359 7B performs poorly without training (achieving only 9.5% accuracy even under the simplest 2×2
 360 setting) while the proprietary Gemini2.5-Pro and the larger Qwen2.5-VL-72B models also struggle
 361 on this task. After supervised fine-tuning (SFT) with 1.6K cold-start trajectories and reinforcement
 362 learning (RL) with 15.6K images, Qwen2.5-VL-7B achieves substantial gains: on the 2×2 setting,
 363 *Acc* improves from 9.5% to 82.8% and *Score* from 29.4% to 89.0%; on the more challenging 3×3
 364 setting, it also generalizes well, with *Acc* increasing from 0.4% to 20.8% and *Score* from 31.1%
 365 to 62.1%, significantly surpassing Gemini2.5-Pro and Qwen2.5-VL-72B. These results demonstrate
 366 that modeling jigsaw solving as an interactive multi-turn dialogue enables the model to progressively
 367 enhance its visual perception and reasoning abilities, thereby achieving superior performance on the
 368 jigsaw task.

369 **Generalization to Downstream Visual Tasks.** The proposed agentic jigsaw interaction learning
 370 framework delivers substantial improvements on the jigsaw task. By leveraging explicit step-by-step
 371 feedback for iterative refinement, it enables the model to capture visual relations more effectively
 372 and to develop stronger reasoning capabilities.

373 To further assess whether jigsaw training enhances performance on general vision downstream tasks,
 374 we evaluate the model on 9 benchmarks: high-resolution image understanding (HRBench4K (Wang
 375 et al., 2025c), HRBench8K (Wang et al., 2025c), VStarBench (Wu & Xie, 2024)), real-world scene
 376 understanding (MME-RealWorld (Zhang et al., 2024), RealWorldQA (xAI, 2024)), fine-grained
 377 visual recognition (MMVP (Tong et al., 2024), BLINK (Fu et al., 2024)), visual reasoning (MMMU
 (Yue et al., 2024)), and hallucination evaluation (HallusionBench (Guan et al., 2024)). As shown in

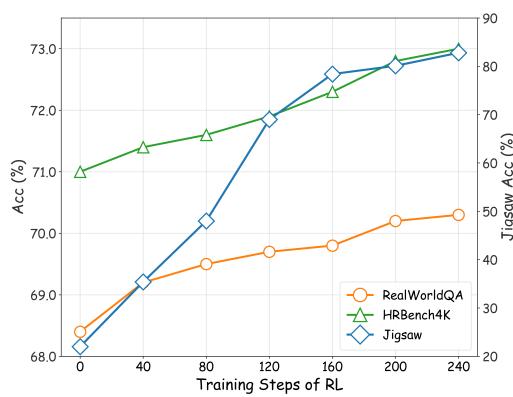
378
379 **Table 3: Main results.** Performance comparison of different models on the 9 benchmarks. Ab-
380 breviations: MME-RW (MME-RealWorld-Lite), RWQA (RealWorldQA), HRB4K (HRBench4K),
381 HRB8K (HRBench8K), HalBench (HallusionBench), MMMU (MMMU_VAL), Avg. denotes the
382 average performance across all 9 benchmarks. Δ represents the relative performance gain achieved
383 by RL compared to the base model Qwen2.5-VL-7B. The best results are highlighted in **bold**, and
384 the second-best results are underlined.

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
LLaVA-OV-7B	48.4	69.5	65.3	58.4	73.3	77.3	52.6	36.6	48.2	58.8
InternVL2.5-8B	48.2	69.4	68.0	63.3	71.7	75.7	54.9	49.9	53.6	61.6
InternVL2.5-78B	49.7	78.4	74.5	72.5	75.9	83.0	63.6	57.1	65.4	68.9
Qwen2.5-VL-72B	44.7	75.3	80.1	77.1	85.9	81.7	61.6	53.5	66.9	69.6
Qwen2.5-VL-7B	44.6	<u>68.5</u>	68.8	65.3	76.4	74.3	<u>56.4</u>	<u>50.1</u>	<u>54.8</u>	62.1
+ Cold-Start	<u>46.2</u>	68.4	<u>71.0</u>	68.4	<u>77.5</u>	<u>76.7</u>	55.7	49.8	54.0	<u>63.1</u>
+ RL	48.4	70.2	73.0	70.5	80.6	78.0	58.0	51.9	55.8	65.2
Δ (vs. Qwen2.5-VL-7B)	+3.8	+1.7	+4.2	+5.2	+4.2	+3.7	+1.6	+1.8	+1.0	+3.1

392
393 Table 3, the jigsaw-trained model demonstrates strong visual generalization, achieving notable gains
394 on HRBench4K (+4.2%), HRBench8K (+5.2%), and VStarBench (+4.2%). On average, it surpasses
395 the base model Qwen2.5-VL-7B by 3.1% across all 9 benchmarks, providing compelling evidence
396 that jigsaw-based training effectively enhances the model’s ability to capture visual relations and
397 strengthen reasoning skills, thereby improving its performance on general vision downstream tasks.

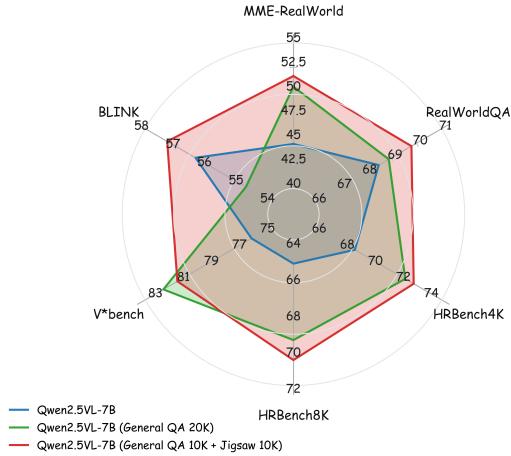
4.3 ANALYSIS

400 **Impact of Jigsaw Training Data Scale.** To investigate the effect of data scale, we systematically
401 vary the amount of training data and evaluate model performance on both jigsaw task and
402 general vision tasks. As shown in Figure 3, scaling up the data yields substantial performance gains:
403 jigsaw task accuracy increases from 22.0% to 82.8%, while HRBench4K and RealWorldQA see
404 improvements of 2.0% and 1.8%, respectively. These results highlight a clear data-to-performance
405 trend, demonstrating that larger training sets translate directly into stronger perceptual and reasoning
406 abilities. Crucially, because jigsaw data in our reinforcement learning stage are generated via
407 scalable programmatic synthesis, AGILE can easily leverage larger datasets. This scalability not
408 only provides richer learning signals for continual capability growth but also offers a practical and
409 sustainable solution to the scarcity of high-quality multimodal RL data.

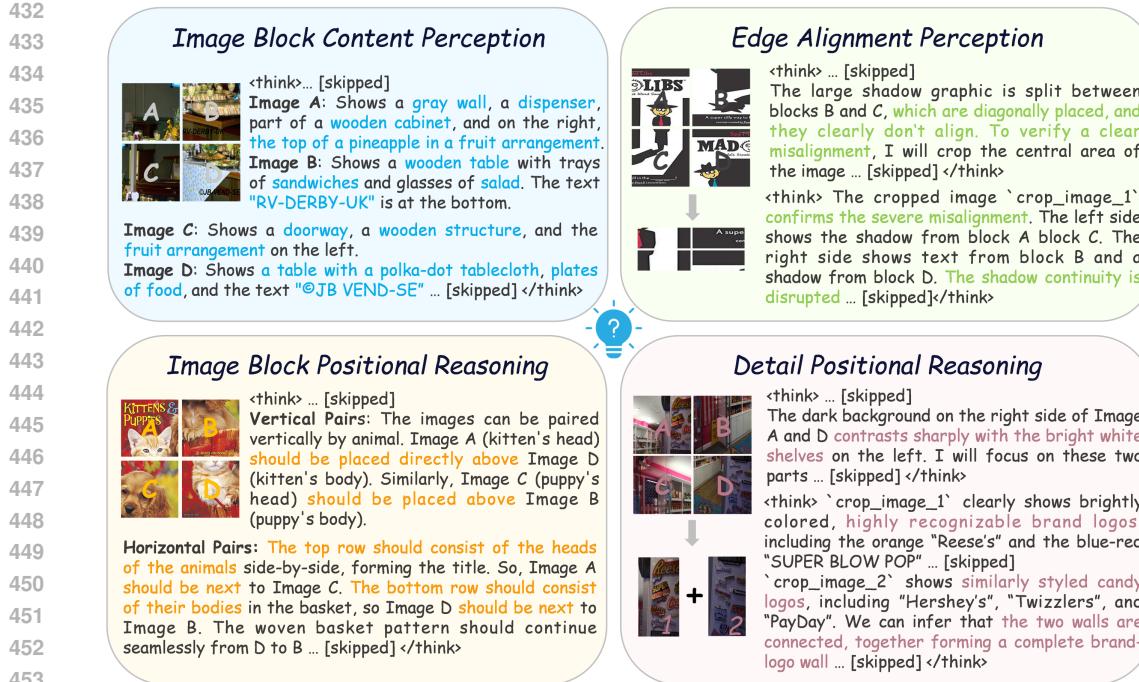


424 **Figure 3: Impact of Training Data Scale.** The left y-axis denotes the accuracies on
425 HRBench4K and RealWorldQA, while the right y-axis corresponds to the accuracy on the jigsaw
426 task.
427

431 **Performance Comparison: Jigsaw vs. General QA Data.** We compare the effectiveness of jigsaw
432 data against conventional General QA data for model training. As shown in Figure 4, models trained



424 **Figure 4: Comparison with General QA Data.** The total number of samples is consistently
425 maintained at 20K across both experimental se-
426 tups.
427

Figure 5: **Case Study.** Jigsaw-solving reasoning and behaviors exhibited by our model.

with jigsaw data consistently outperform those trained with QA data across multiple benchmarks. Notably, combining 10K jigsaw samples with 10K QA samples yields superior performance on general vision benchmarks compared to training with 20K QA samples alone. These results highlight the unique role of jigsaw data in enhancing fundamental visual generalization, demonstrating the efficiency and controllability of jigsaw-solving as a proxy for visual perception and reasoning. A key advantage of our approach lies in the programmatic synthesis of jigsaw data, which enables effortless construction of large-scale, high-quality datasets, a process that is otherwise costly and labor-intensive for QA data. Thus, jigsaw data not only serves as an effective alternative to General QA data but, in some cases, proves to be the superior choice. This finding underscores the potential of jigsaw tasks in alleviating the scarcity of multimodal RL data and opens a promising new direction for advancing multimodal model development.

Case Study. In Figure 5, we showcase several jigsaw-solving reasoning patterns and behaviors exhibited by our model. These cases illustrate the emergence of high-quality perceptual and reasoning strategies in the jigsaw task, including comprehensively interpreting the visual content of individual pieces to infer their spatial relations, employing cropping and zooming to examine and validate edge alignment, and reasoning about semantic consistency across pieces. Such behaviors demonstrate human-like reasoning, thereby effectively enhancing the model's perceptual and reasoning capabilities.

5 CONCLUSION

In this work, we propose a novel agentic jigsaw interaction learning framework that formulates jigsaw solving as an interactive process to enhance visual perception and reasoning in large Vision-Language Models (VLMs). By treating jigsaw solving as an iterative interaction, the model progressively refines its perceptual and reasoning capabilities through exploration and feedback. Our approach significantly improves performance on jigsaw tasks of varying complexity and demonstrates strong generalization to broader general visual tasks, including visual question answering and scene understanding. Furthermore, by analyzing the performance gains from increasing the scale of jigsaw data and comparing it with general QA data, we show that jigsaw serve as an effective proxy for alleviating the scarcity of high-quality RL data, highlighting the potential of task-driven proxy training for stimulating complex multimodal perception and reasoning in VLMs.

486

6 ETHICS STATEMENT

487
488 This work adheres to ethical research standards in data collection, model training, and evaluation.
489 All datasets used in this study are publicly available research datasets (e.g., COCO, TextVQA,
490 HRBench, RealWorldQA), which were collected and released under their respective licenses. No
491 private or personally identifiable information (PII) was used.
492493

7 REPRODUCIBILITY STATEMENT

494
495 We are committed to ensuring the reproducibility of our results. Comprehensive implementation
496 details, including training data and hyperparameter settings, are provided in Appendices A and D.
497 To further facilitate reproducibility, we will release the training code, datasets, and evaluation scripts
498 upon publication.
499500

REFERENCES

501
502 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang,
503 Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*,
504 2025.505
506 Fabio M Carlucci, Antonio D’Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Do-
507 main generalization by solving jigsaw puzzles. In *Proceedings of the IEEE/CVF conference on*
508 *computer vision and pattern recognition*, pp. 2229–2238, 2019.509
510 Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu,
511 Xuefeng Li, Jiaze Chen, et al. Enigmata: Scaling logical reasoning in large language models with
512 synthetic verifiable puzzles. *arXiv preprint arXiv:2505.19914*, 2025a.513
514 Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-v: Reinforcing super generalization
515 ability in vision-language models with less than \$3. <https://github.com/Deep-Agent/R1-V>, 2025b. Accessed: 2025-02-02.516
517 Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Daha-
518 Lin. Sharegpt4v: Improving large multi-modal models with better captions. In *European Confer-
519 ence on Computer Vision*, pp. 370–387. Springer, 2024a.520
521 Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi
522 Wang, Yu Qiao, Daha Lin, et al. Are we on the right way for evaluating large vision-language
523 models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024b.524
525 Lin Chen, Xilin Wei, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong
526 Duan, Zhenyu Tang, Li Yuan, et al. Sharegpt4video: Improving video understanding and genera-
527 tion with better captions. *Advances in Neural Information Processing Systems*, 37:19472–19495,
528 2024c.529
530 Yingyi Chen, Xi Shen, Yahui Liu, Qinghua Tao, and Johan AK Suykens. Jigsaw-vit: Learning
531 jigsaw puzzles in vision transformer. *Pattern Recognition Letters*, 166:53–60, 2023.532
533 Gheorghe Comanici, Eric Bieber, Mike Schaeckermann, Ice Pasupat, Noveen Sachdeva, Inderjit
534 Dhillon, Marcel Blstein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the
535 frontier with advanced reasoning, multimodality, long context, and next generation agentic capa-
536 bilities. *arXiv preprint arXiv:2507.06261*, 2025.537
538 Qingxiu Dong, Li Dong, Yao Tang, Tianzhu Ye, Yutao Sun, Zhifang Sui, and Furu Wei. Reinforce-
539 ment pre-training. *arXiv preprint arXiv:2506.08007*, 2025.540
541 Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong,
542 Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evalua-
543 ting large multi-modality models. In *Proceedings of the 32nd ACM international conference on*
544 *multimedia*, pp. 11198–11201, 2024.

540 Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A
 541 Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but
 542 not perceive. In *European Conference on Computer Vision*, pp. 148–166. Springer, 2024.

543

544 Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang
 545 Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for
 546 entangled language hallucination and visual illusion in large vision-language models. In *Pro-
 ceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–
 547 14385, 2024.

548

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

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

555

556 Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Os-
 557 trow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint
 arXiv:2410.21276*, 2024.

558

559 Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus:
 560 Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv
 preprint arXiv:2404.16790*, 2024a.

561

562 Xiaotong Li, Fan Zhang, Haiwen Diao, Yueze Wang, Xinlong Wang, and Lingyu Duan.
 563 Densefusion-1m: Merging vision experts for comprehensive multimodal perception. *Advances
 564 in Neural Information Processing Systems*, 37:18535–18556, 2024b.

565

566 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr
 567 Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European
 568 conference on computer vision*, pp. 740–755. Springer, 2014.

569

570 Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-
 571 Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of
 572 foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

573

574 Minesh Mathew, Viraj Bagal, Rubén Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar.
 575 Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer
 Vision*, pp. 1697–1706, 2022.

576

577 Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi,
 578 Wenhui Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with
 rule-based large-scale reinforcement learning. *CoRR*, 2025.

579

580 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 581 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances
 582 in neural information processing systems*, 36:53728–53741, 2023.

583

584 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
 585 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

586

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

590

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

594

595 Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh,
 596 and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF
 597 conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

594 Jingqi Tong, Jixin Tang, Hangcheng Li, Yurong Mou, Ming Zhang, Jun Zhao, Yanbo Wen, Fan Song,
 595 Jiahao Zhan, Yuyang Lu, et al. Code2logic: Game-code-driven data synthesis for enhancing vlms
 596 general reasoning. *arXiv preprint arXiv:2505.13886*, 2025.

597

598 Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide
 599 shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF*
 600 *Conference on Computer Vision and Pattern Recognition*, pp. 9568–9578, 2024.

601

602 Qiuchen Wang, Ruixue Ding, Zehui Chen, Weiqi Wu, Shihang Wang, Pengjun Xie, and Feng Zhao.
 603 Vidorag: Visual document retrieval-augmented generation via dynamic iterative reasoning agents.
 604 *arXiv preprint arXiv:2502.18017*, 2025a.

605

606 Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei
 607 Huang, and Feng Zhao. Vrag-rl: Empower vision-perception-based rag for visually rich in-
 608 formation understanding via iterative reasoning with reinforcement learning. *arXiv preprint*
 609 *arXiv:2505.22019*, 2025b.

610

611 Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng
 612 Tao. Divide, conquer and combine: A training-free framework for high-resolution image percep-
 613 tion in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial*
 614 *Intelligence*, volume 39, pp. 7907–7915, 2025c.

615

616 Xiyao Wang, Zhengyuan Yang, Chao Feng, Yongyuan Liang, Yuhang Zhou, Xiaoyu Liu, Ziyi Zang,
 617 Ming Li, Chung-Ching Lin, Kevin Lin, et al. Vicrit: A verifiable reinforcement learning proxy
 618 task for visual perception in vlms. *arXiv preprint arXiv:2506.10128*, 2025d.

619

620 Zifu Wang, Junyi Zhu, Bo Tang, Zhiyu Li, Feiyu Xiong, Jiaqian Yu, and Matthew B Blaschko.
 621 Jigsaw-rl: A study of rule-based visual reinforcement learning with jigsaw puzzles. *arXiv*
 622 *preprint arXiv:2505.23590*, 2025e.

623

624 Penghao Wu and Saining Xie. V?: Guided visual search as a core mechanism in multimodal llms.
 625 In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.
 626 13084–13094, 2024.

627

628 xAI. Grok-1.5 Vision Preview. <https://x.ai/news/grok-1.5v>, 2024.

629

630 Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu,
 631 Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement
 632 learning. *arXiv preprint arXiv:2502.14768*, 2025a.

633

634 Yunfei Xie, Yinsong Ma, Shiyi Lan, Alan Yuille, Junfei Xiao, and Chen Wei. Play to generalize:
 635 Learning to reason through game play. *arXiv preprint arXiv:2506.08011*, 2025b.

636

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

640

641 En Yu, Kangheng Lin, Liang Zhao, Jisheng Yin, Yana Wei, Yuang Peng, Haoran Wei, Jianjian Sun,
 642 Chunrui Han, Zheng Ge, et al. Perception-rl: Pioneering perception policy with reinforcement
 643 learning. *arXiv preprint arXiv:2504.07954*, 2025.

644

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

649

650 Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng
 651 Li, Kun Wang, Qingsong Wen, Zhang Zhang, et al. Mme-realworld: Could your multimodal
 652 llm challenge high-resolution real-world scenarios that are difficult for humans? *arXiv preprint*
 653 *arXiv:2408.13257*, 2024.

654

655 Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyuan Luo, Zhangchi Feng, and
 656 Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. *arXiv*
 657 *preprint arXiv:2403.13372*, 2024.

648 Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and
649 Xing Yu. Deepeyes: Incentivizing” thinking with images” via reinforcement learning. *arXiv*
650 *preprint arXiv:2505.14362*, 2025.

651 Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen
652 Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for
653 open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

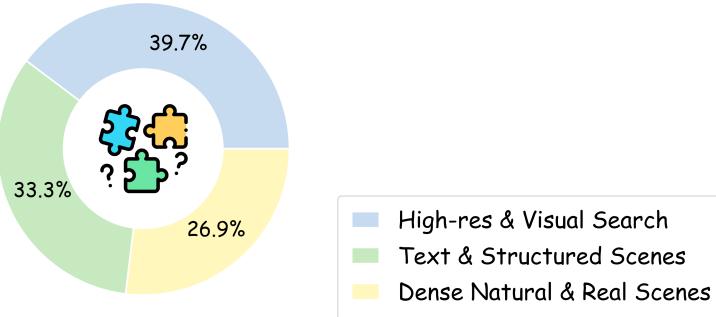
697

698

699

700

701

702 A DATA DISTRIBUTION
703704 As shown in Figure 6, our RL training corpus for jigsaw is derived from multiple sources, covering
705 a diverse range of visual scenarios:
706718 Figure 6: Distribution of jigsaw RL training data.
719720 • **High-resolution and visual search images (39.7%, 6.2K samples):** To enhance the model’s
721 fine-grained perception in high-resolution images, we collect data from the VStar (Wu & Xie, 2024)
722 dataset, high-resolution natural scenes in DeepEyes (Zheng et al., 2025), and selected samples from
723 HRBench (Wang et al., 2025c). This improves the model’s ability to capture subtle visual cues and
724 recognize small object attributes.725 • **Text recognition and structured scenes (33.3%, 5.2K samples):** To strengthen the model’s
726 capacity for text perception and recognition, we include text-rich images from diverse domains,
727 such as the TextVQA (Singh et al., 2019) training set, InfoVQA (Mathew et al., 2022) book covers
728 and posters, as well as structured visual reasoning tasks involving tables and subject-specific charts
729 (Yue et al., 2024; Li et al., 2024a; Chen et al., 2024b).730 • **Dense natural and real-world scenes (26.9%, 4.2K samples):** To enhance the model’s recogni-
731 tion and understanding in complex and real-world environments, we collected images from
732 COCO2017 (Lin et al., 2014) and RealWorldQA (xAI, 2024) datasets.
733734 B PROMPTS
735736 In this section, we present the prompt used to collect high-quality jigsaw trajectories, as illustrated
737 in Figure B. The same prompt is employed to prompt the Qwen2.5-VL-7B model during both the
738 SFT and RL stages.
739740 Prompt for Interactive Collection of High-Quality Jigsaw Trajectories
741742 **System Prompt:**743 You are a **skilled and experienced puzzle master**. I will divide an image evenly into 2×2
744 grid, get 4 image blocks, and then shuffle them. The image blocks will be labeled A, B, C,
745 and D. Initially, the image blocks are named and arranged in the following order:
746747 `["A", "B", "C", "D"]`
748749 That is, the initial layout of the image blocks is as follows:
750 A B
751 C D
752753 **Your Task:**754 Your goal is to **reconstruct the original image** by observing and analyzing the **visual**
755 **content** between the image blocks. Pay attention to the **details in each image block**

756

757

758

and establish **visual connections** between adjacent image blocks to achieve the goal of completing the puzzle correctly. For example:

759

- **Continuity of text information:** If the image contains text or logos, observe whether the text continues naturally between adjacent blocks (e.g., characters connect smoothly and the font direction is consistent). This is one of the key clues to determine whether the image blocks should be adjacent.

760

- **Structure and shape consistency:** Identify possible structural elements in the image (such as buildings, roads, object outlines, etc.), and infer which image blocks should be visually spliced together to form a complete and reasonable shape.

761

- **Continuation of edge visual details:** Observe the color, texture or pattern (such as sky, grass, lines or shadows) at the edge of the image block to determine whether it can be visually connected to the adjacent block naturally.

762

- **Direction and angle consistency:** Pay attention to the direction of image elements (such as human faces, object directions, text angles, etc.) to ensure the rationality of the overall visual direction of the image.

763

You can **swap any two image blocks each time** to achieve the correct layout.

764

Inference Process:

765

1. **State Representation:** At each step, you need to maintain a “state” list representing current arrangement of image blocks.

766

Example:

767

state = [“B”, “C”, “A”, “D”]

768

This corresponds to:

769

Top left (index 0): B

770

Top right (index 1): C

771

Bottom left (index 2): A

772

Bottom right (index 3): D

773

2. **Visual Analysis:** Carefully analyze the image patch content and visual details to determine the image patch location. To better complete the puzzle task, you can use additional image operations to enhance visual perception:

774

- **Crop the image area of observation.image** to more closely observe the visual correlation between different regions in multiple image patches. Used to find clues during the puzzle or verify whether the puzzle is complete (For example: it looks like the puzzle is not completely fixed, maybe because of the lower left and upper right corners of observation.image_2. I need to crop these two areas separately to get some visual information to further judge. First I will crop the lower left area `<code>...</code>` By observing the cropped lower left area, it shows that part of the text “OF H” may be related to the “appyness” text in the upper right corner. To further verify and manage, I will now crop the upper right image `<code>...</code>`. By observing the cropped image, I will swap image blocks B and C...)

775

- **Zoom area** to enlarge visual details (selectively observe the details of the cropped image).

776

These operations can help you make more informed decisions.

777

3. **Move Execution:** After making a decision, generate a Python code snippet to swap tiles. Use the ‘observation(state)’ function to get the updated layout image for the next step.

778

779

780

781

782

810

Available Image Operations:

You can use the following Python functions to assist in inference:

1. Crop a region from an image using normalized coordinates (from 0 to 1).

```
crop_box = [x1, y1, x2, y2]
crop_image_{id} = crop(image, crop_box)
```

2. Zoom in an image by a specified factor (e.g., $1.5 \times$ zoom means 150% zoom).

```
zoom_image_{id} = zoom(image, zoom_factor)
```

3. Swap tiles (e.g., top left and bottom left) and call observation function to see the jigsaw progress.

```
state[0], state[2] = state[2], state[0]
observation_image_{id} = observation(state)
```

Replace “id” with an integer to uniquely identify each operation result. The “image” parameter must point to an existing image (e.g., “observation_image_1”, “crop_image_1”, etc.).

Format:All Python code **must** be enclosed in `<code>...</code>` tags.All reasoning steps **must** be enclosed in `<think>...</think>` tags.The final answer **must** be enclosed in `<answer>...</answer>` tags.**Important:**1. For both swap and image manipulation, **always explain why you are doing this** before generating code.2. **Stop as soon as you generate a code snippet.** I will execute the code and return the generated image for the next step.

3. You can only perform one image operation or observation of the puzzle state per turn.

4. **Final answer:** When you are sure that the puzzle is fully reconstructed (all edges are aligned and the visual information is complete and continuous), return the final “state” list in the “`<answer>`” block.5. Please pay close attention to **the rationality of your cropping area** and ensure that the cropped area is **a reasonable criterion for completing the puzzle**.**User Prompt:**

The four images are respectively labeled A, B, C, and D. Please complete the jigsaw.

Image A: <image> **Image B:** <image> **Image C:** <image> **Image D:** <image>

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864 C CASE STUDY
865

866 In Figures 7 - 10, we present representative high-quality jigsaw trajectories collected to illustrate
867 how our model performs reasoning and interacts with the environment. These cases specifically
868 demonstrate: (1) how the model generates Python code to interact with the environment, such as
869 swapping any two jigsaw pieces, observing jigsaw progress, and cropping/zooming in on regions
870 for finer visual inspection; and (2) how the environment provides feedback that guides the model's
871 subsequent reasoning.

914 Figure 7: Representative high-quality jigsaw trajectory case 1 (part 1).
915
916
917

918
919
920
921
922
923
924
925
926
927



928 <think> **Current State Analysis:**
929 The current state is ["B", "D", "C", "A"].
930 * Top Left (B): Top-left part of the scene, seems correct.
931 * Top Right (D): Top-right part of the scene, the price sign "\$23.88" connects perfectly. This seems correct.
932 * Bottom Left (C): Bottom-right part of the purple display.
933 * Bottom Right (A): Bottom-left part of the purple display.
934 **Reasoning for the next move:**
935 1. The top row (B, D) looks correct. The price sign is complete, and the store background and ceiling align.
936 2. The bottom row (C, A) is incorrect. The purple display is split. Tile A shows the left side of the bottom of the purple display, and tile C shows the right side.
937 3. Tile A should be placed under tile B, as they both show the left side of the scene (the shelves with green/red boxes connect, and the purple display connects vertically).
938 4. Tile C should be placed under tile D, as they both show the right side of the scene (the white display with the woman's face is to the right of the purple display). Therefore, I need to swap C and A.
939 Before I perform the swap, I will crop a region to confirm the connection between tiles B and A. I'll crop the bottom edge of B and the top edge of A to see if the "Boost" boxes and the purple display align. I'll crop the area where the purple display stand meets the shelving unit. </think>

940 <code> # Crop region to verify alignment between B and A
941 crop_box = [0, 0.4, 0.5, 0.6]
942 crop_image_1 = crop(observation_image_2, crop_box)



941
942
943
944
945



946 <think>
947 **Current State Analysis:**
948 The current state is ["B", "D", "C", "A"].
949 The crop showed the bottom of B and the top of C. I wanted to check the seam between B and A. The crop shows the seam between B (top left) and C (bottom left). This seam is clearly incorrect. The "Nexium" display doesn't align, and the background shelving doesn't connect. This confirms that C is not in the correct position below B.

950 My initial analysis was correct: A should be below B, and C should be below D. The current layout has them swapped. I will now perform the swap between C and A. </think>

951 <code> # Current state of the tile
952 state = ["B", "D", "C", "A"]
953 # Swap tiles C (bottom left) and A (bottom right)
954 state[2], state[3] = state[3], state[2]
955 # Call observation function to see the result
956 observation_image_3 = observation(state)



955
956
957
958
959



960 <think>
961 **Final State Analysis:**
962 The current state is ["B", "D", "A", "C"].
963 Top Row (B, D): The price sign "\$23.88" is complete. The store's ceiling and background shelves align correctly.
964 Bottom Row (A, C): The purple "Nexium 24HR" display stand is now complete and correctly assembled. The text on the display flows naturally across the tiles. The white floor is continuous.
965 Vertical Alignment: The top of the purple display in B connects seamlessly with the bottom part in A. The background shelves behind the purple display in D connect with the floor and the adjacent white display in C. All edges and visual elements align perfectly. The image is now reconstructed. </think>

966 <answer> ["B", "D", "A", "C"] </answer>



967 Figure 8: Representative high-quality jigsaw trajectory case 1 (part 2).
968
969
970
971

972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025

The four images are respectively labeled A, B, C, and D.
Please complete the jigsaw.

User

Image A Image B Image C Image D

<think>
Based on the initial analysis of the four image blocks, I will begin by arranging the puzzle.
Image B shows the top-left corner of a postcard. It has the text "THIS SIDE OF CARD IS F..." and the left edge of a postmark. The top and left sides have a dark border, indicating it's the corner of the scanned item. This strongly suggests that B should be in the top-left position (index 0).
Image D shows a postage stamp and the right side of a postmark that seems to continue from the one in Image B ("EAPOLIS, MINN." and "MINN."). It also has the text "...OR ADDRESS", which completes the phrase from Image B. This suggests D should be in the top-right position (index 1).
Image C contains the beginning of a handwritten address: "Dr. R. E.", "Dept. of Pre...", "Washington Univ.". This should be below the postcard's header. This suggests C should be in the bottom-left position (index 2).
Image A continues the handwritten address from C: "...ntive Medicine", "School of Medicine", "...ouis 10, Missouri". This suggests A should be in the bottom-right position (index 3).
My target layout is 'B D' on top and 'C A' on the bottom, which corresponds to the state `["B", "D", "C", "A"]`.
I'll start by moving block B to the top-left corner. To do this, I will swap block A and block B.
Current state: `["A", "B", "C", "D"]`
Desired action: Swap A and B.
Resulting state: `["B", "A", "C", "D"]`
</think>

<code> # Initial state
state = ["A", "B", "C", "D"]
Swap image block A (top left) and B (top right)
state[0], state[1] = state[1], state[0]
Observe the new layout
observation_image_1 = observation(state)
</code>

The code is executed successfully. `observation_image_1` is

User

<think>
The current state is `["B", "A", "C", "D"]`.
From `observation_image_1`, I can see:
- Top-left (B) is correctly placed. It shows the top-left corner of the postcard.
- Top-right (A) is incorrect. The handwritten text does not connect with anything in block B. The correct block should be D, which has the stamp and completes the text "THIS SIDE OF CARD IS FOR ADDRESS".
- Bottom-left (C) seems correctly placed relative to the top row, as it's the start of the address.
- Bottom-right (D) is incorrect. It should be in the top right.
My next step is to move block D to the top-right position. I can achieve this by swapping block A and block D.
Current state: `["B", "A", "C", "D"]`
Desired action: Swap A (top-right) and D (bottom-right).
Resulting state: `["B", "D", "C", "A"]`
</think>

<code> # Current state
state = ["B", "A", "C", "D"]
Swap image block A (top right) and D (bottom right)
state[1], state[3] = state[3], state[1]
Observe the new layout
observation_image_2 = observation(state)
</code>



Figure 9: Representative high-quality jigsaw trajectory case 2 (part 1).

1026
 1027
 1028
 1029
 1030
 1031

User  The code is executed successfully. `observation_image_2` is 

1032 **<think>** The current state is `["B", "D", "C", "A"]`. Looking at `observation_image_2`:
 - The top row (B and D) looks correct. The text "THIS SIDE OF CARD IS FOR ADDRESS" is complete, and the postmark "MINNEAPOLIS MINN JAN 29 PM 1954" is also complete. The top dark border continues from B to D.
 - The bottom row (C and A) also looks correct.
 - Block C (bottom-left) has "Dr. R. E.", "Dept. of Pre...", "Washington Univ.", "Euclid Ave.", "St. L...".
 - Block A (bottom-right) continues this with "...Shank", "...ventive Medicine", "...School of Medicine", "... & Kingshighway", "...ouis 10, Missouri".
 - The handwritten text flows correctly from C to A.

1033 To be absolutely sure, I will crop the area where the handwritten text transitions between block C and block A. I will focus on the continuity of the words "Preventive Medicine" and "Washington Univ." with "School of Medicine", and "Euclid Ave." with "& Kingshighway".

1034 I will crop the region spanning the right edge of C and the left edge of A, focusing on the middle lines of text.
 Coordinates for C: `x_end` is 1.0, `y_start` is roughly 0.1 of C's height, `y_end` is roughly 0.6 of C's height.
 Coordinates for A: `x_start` is 0.0, `y_start` is roughly 0.1 of A's height, `y_end` is roughly 0.6 of A's height.

1035 In the combined image ('observation_image_2'), this translates to:
 `x1` (middle of C) = 0.4 (approx); `y1` (middle of C) = $0.5 + 0.5 * 0.1 = 0.55$ (approx); `x2` (middle of A) = 0.6 (approx); `y2` (middle of A) = $0.5 + 0.5 * 0.6 = 0.8$ (approx)

1036 Let's refine the crop box to be more specific to the join between C and A.
 The join is at `x=0.5`. I'll crop from `x=0.4` to `x=0.6`.
 The text of interest ("Dept. of Pre..." to "School of Med...") is in the vertical range from about `y=0.6` to `y=0.8` in `observation_image_2`. So, the crop box will be `[0.4, 0.6, 0.6, 0.8]`. **</think>**

1037

1038

1039

1040

1041

1042

1043

1044

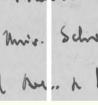
1045 **<code>** # Current state
 state = ["B", "D", "C", "A"]
 # Crop the region where handwritten text from C meets A 
 # for closer inspection of continuity.
 # x1=0.4, y1=0.6 (middle-ish of image, focusing on text lines)
 # x2=0.6, y2=0.8
 crop_box = [0.4, 0.6, 0.6, 0.8]
 crop_image_1 = crop(observation_image_2, crop_box) **</code>**

1046

1047

1048

1049

1050 **<think>** The code is executed successfully. `crop_image_1` is 

1051

1052

1053

1054

1055 **<think>** The cropped image `crop_image_1` shows the seam between the bottom-left and bottom-right blocks.
 - On the left side (from block C), we can see the end of "Washington Univ." and "Euclid Ave.".
 - On the right side (from block A), we can see the beginning of "School of Medicine" and "& Kingshighway".

1056 The lines of text do not seem to align perfectly. For example, "Univ." on the left is on the same line as "Schoo..." on the right. This is incorrect. The address should be on separate lines. "Dr. R. E. Shank"; "Dept. of Preventive Medicine"; "Washington Univ. School of Medicine"; "Euclid Ave. & Kingshighway"; "St. Louis 10, Missouri"

1057

1058

1059

1060 Looking back at `observation_image_2`, I can see:
 Line 1 in C: "Dr. R. E." should connect with "Shank" in A. Line 2 in C: "Dept. of Pre..." should connect with "ventive Medicine" in A. Line 3 in C: "Washington Univ." should connect with "School of Medicine" in A. Line 4 in C: "Euclid Ave." should connect with "& Kingshighway" in A. Line 5 in C: "St. L..." should connect with "ouis 10, Missouri" in A.

1061 Let's re-examine `observation_image_2` carefully.
 The word "Shank" starts in A, but its beginning is cut off. It seems to connect with "Dr. R. E." in C. "Dept. of Pre" in C seems to connect with "ventive Medicine" in A. "Washington Univ." in C seems to connect with "School of Medicine" in A. "Euclid Ave" in C seems to connect with "... & Kingshighway" in A. "St. L..." in C seems to connect with "ouis 10, Missouri" in A.

1062 The `crop_image_1` showed a misalignment. Let me re-evaluate my initial thought. "Washington Univ." from C and "School of Medicine" from A are parts of the same organization, but they might not be on the same line. Let's look at the vertical alignment. The text "Dr. R. E." from C looks horizontally aligned with "Shank" from A. "Dept. of Pre" from C looks horizontally aligned with "ventive Medicine" from A. And so on for the rest of the lines.

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

The continuity is correct. The crop may have been slightly misleading due to the cursive handwriting style and line breaks. The overall image `observation_image_2` looks correct. The top row is perfectly aligned, and the bottom row's text is a continuous, logical address. The dark border on the left of C and the right of A match the overall scan boundary. Therefore, the puzzle is solved. **</think>**

<answer> ["B", "D", "C", "A"] **</answer>**



Figure 10: Representative high-quality jigsaw trajectory case 2 (part 2).

D THE IMPLEMENTATION OF TRAINING

The detailed training hyperparameters are provided in Tables 4 and 5, and all experiments are conducted on 8 NVIDIA A100 GPUs, each equipped with 80 GB of memory.

1080
1081

Table 4: Key hyperparameters for SFT.

1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094

Name	Value
Finetuning type	Full
Freeze vision tower	True
Freeze multi-modal projector	True
Freeze language model	False
Cutoff len	16384
Image max pixels	401408
Epochs	2.0
Batch size	32
Gradient accumulation steps	4
Learning rate	1.0e-5
LR scheduler type	cosine
Warmup ratio	0.1

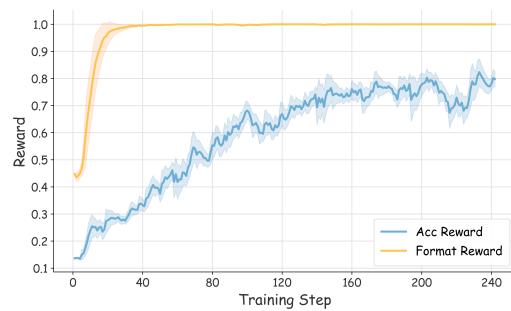
1095

1096
1097

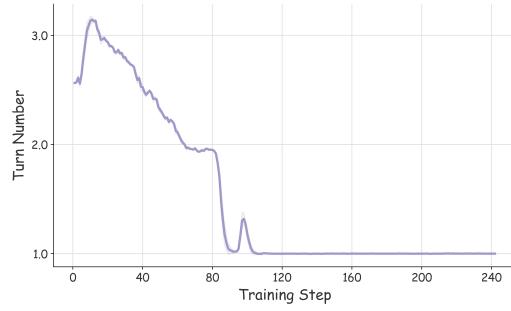
E ANALYSIS OF WANDB CURVES IN JIGSAW OPTIMIZATION

1098
1099
1100
1101

In this section, we present the training curves of our RL process and provide an analysis of the training dynamics. We monitor the evolution of rewards (including accuracy and format rewards), response length, number of interaction turns, and validation accuracy.

1102
1103
1104
1105
1106
1107
1108
1109
1110
1111

(a) Reward mean.

1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122

(c) Turn number mean.

1123
1124
1125
1126

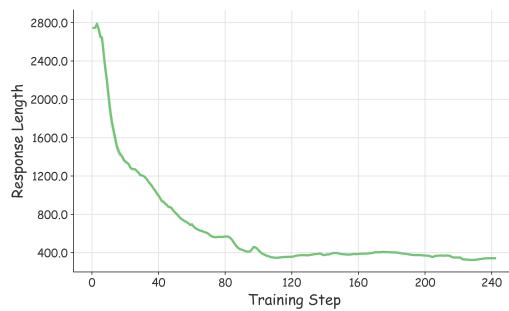
Figure 11: Visualization of Wandb curves in jigsaw RL optimization.

1127
1128
1129
1130
1131
1132
1133

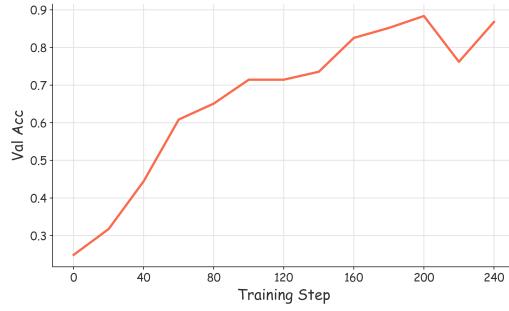
At the beginning of training, both validation accuracy and reward values are low, indicating that the model has very limited jigsaw-solving ability in the early stage. As training progresses, the number of interaction turns increases briefly, suggesting that the model explores more interactions (e.g., swapping tiles, cropping for observation) in order to achieve higher accuracy. With continued training, the reward values steadily increase, reflecting a stable improvement in jigsaw-solving capability through RL. After sufficient training, the model gradually reduces both the number of turns and response length, demonstrating enhanced perceptual and reasoning ability, such that it can solve jigsaw correctly with fewer interactions.

Table 5: Key hyperparameters for RL.

Name	Value
Max turns	5
Rollout num	8
Train batch size	64
Mini batch size	64
Micro batch size per GPU	2
Learning rate	2.0e-6
KL loss coefficient	0.0
Total epochs	1
Max prompt length	8192
Single response max tokens	2048
Max response length	20000
GPU memory utilization	0.7



(b) Response length mean.



(d) Validation accuracy.

1134 **F LIMITATIONS**
11351136
1137 Despite our best efforts, this work has several limitations. Multi-turn jigsaw interactions inevitably
1138 increase context length and introduce significant computational overhead. In the 3×3 setting,
1139 the context often exceeds the model’s maximum window size, restricting our RL training to the
1140 2×2 case. Future work could investigate more efficient interaction mechanisms with external
1141 environments or integrate memory modules to mitigate context length constraints, thereby enabling
1142 RL training on larger and more complex jigsaw tasks.
11431144 **G QUALITATIVE ANALYSIS ON ATTENTION PATTERNS**
11451146 To further verify that AGILE brings genuine improvements in perceptual and reasoning abilities
1147 rather than merely increasing benchmark scores, we conduct additional qualitative analyses on general
1148 downstream tasks. Specifically, we visualize the attention distributions of Qwen2.5-VL-7B
1149 before and after AGILE training. As shown in attention examples (Figure 12), the original model
1150 often exhibits scattered and unstable attention allocation, focusing on large irrelevant regions when
1151 processing complex high-resolution scenes or fine-grained recognition tasks. This dispersed attention
1152 leads to unreliable reasoning and misinterpretation of critical visual evidence.
11531154 In contrast, the AGILE-trained model shows significantly more concentrated attention on key visual
1155 elements, such as small objects, text regions, and structurally important areas. This improved focus
1156 enhances the model’s ability to extract fine-grained cues and supports more reliable multimodal rea-
1157 soning. These qualitative findings align with our quantitative gains across HRBench, VStarBench,
1158 and RealWorldQA, suggesting that AGILE enhances fundamental perceptual mechanisms underly-
1159 ing general visual understanding.
11601161 Beyond overall benchmark gains, we analyze the types of perceptual and reasoning skills most
1162 enhanced by AGILE:
11631164 • **Fine-grained recognition.** These tasks require high precision in identifying subtle visual cues,
1165 such as detecting small objects, recognizing text, or distinguishing fine-grained visual differences.
1166 AGILE enhances the model’s ability to focus on the most critical visual elements, improving its sen-
1167 sitivity to subtle but important features and thereby boosting fine-grained recognition performance.
11681169 • **Spatial relation reasoning.** Benchmarks such as VStarBench contain a substantial number of
1170 questions involving spatial positional reasoning. AGILE’s interaction-driven training helps the
1171 model better attend to spatial layouts and contextual relationships within a scene, substantially im-
1172 proving its ability to understand and reason about complex spatial arrangements. This capability
1173 transfers effectively to real-world spatial reasoning tasks.
11741175 • **Visual-textual integration.** AGILE also leads to notable improvements in tasks requiring the
1176 integration of visual and textual information. Tasks involving reading product details, interpreting
1177 signs, or extracting structured information from charts benefit from AGILE’s strengthened ability to
1178 jointly process textual cues grounded in visual context.
11791180 **H HANDLING INVALID CODE EXECUTIONS**
11811182 During reinforcement learning, the model may generate malformed or invalid code. Instead of
1183 discarding such rollouts, the environment executes the code and returns explicit error messages to
1184 the model. These erroneous interactions increase the total number of steps required to reach the
1185 correct solution and thus naturally lead to lower step rewards under our reward formulation (Eq. 5
1186 and Eq. 6). As a result, invalid code is implicitly penalized, encouraging the model to generate
1187 valid actions and interact more efficiently. This design stabilizes training without requiring complex
1188 filtering mechanisms.
1189

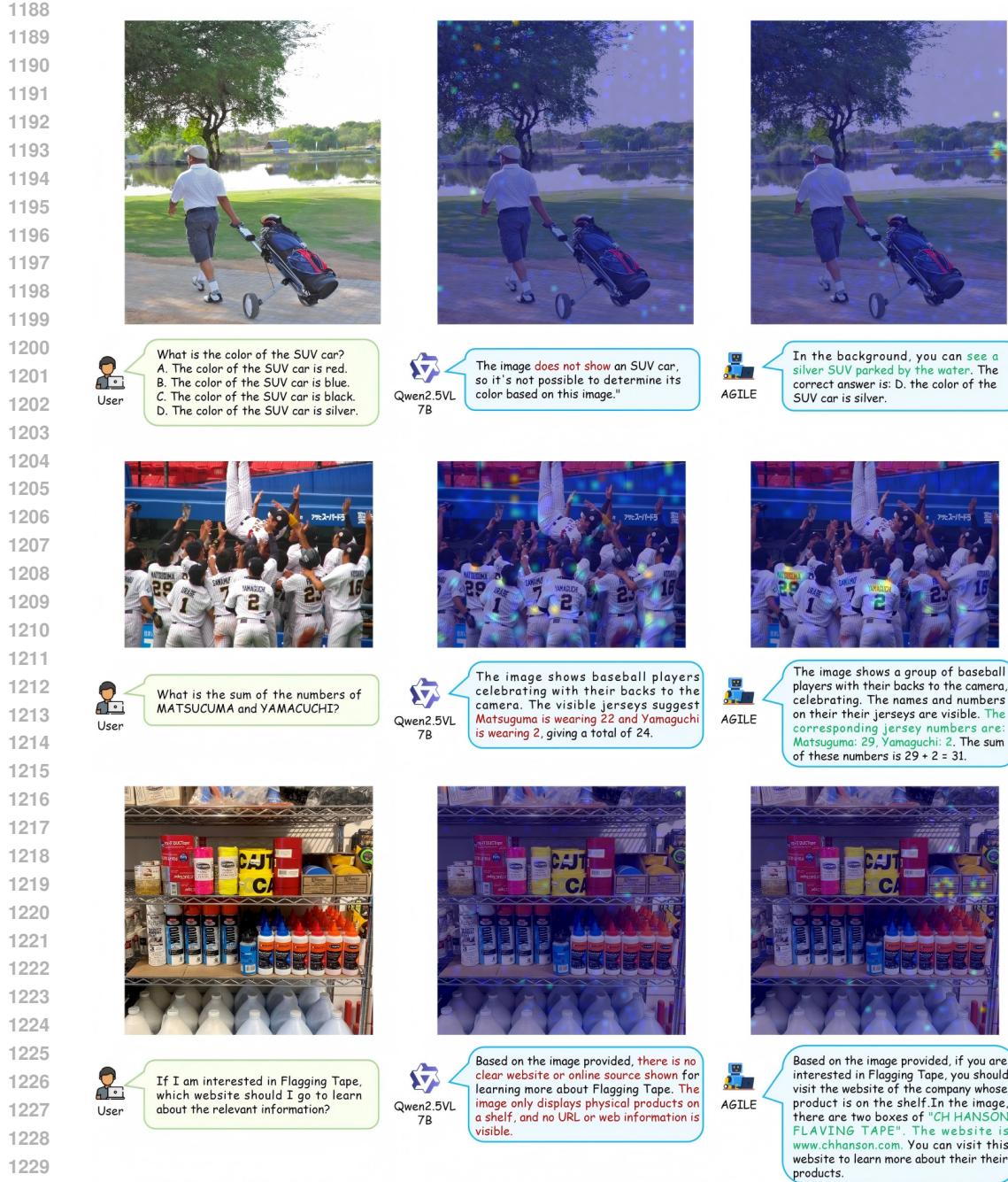


Figure 12: **Comparison of Attention Maps Before and After AGILE Training.** Warm colors indicate higher attention; cool colors indicate lower attention.

I ADDITIONAL EXPERIMENTS

I.1 SENSITIVITY TO REWARD COEFFICIENTS.

We further investigate the sensitivity of AGILE to the reward-weighting coefficients (α , β , and γ). As shown in Table 6, varying the values of α and β yields only minor fluctuations in performance, indicating that AGILE is robust to these parameters. In contrast, removing the step reward (γ) produces notable performance degradation, especially on VStarBench and MMVP. This finding ver-

ifies that the step-penalty plays a critical role in encouraging efficient and meaningful interactions, preventing long, unfocused rollouts during RL.

Table 6: **Ablation on Reward Coefficients.** Sensitivity analysis of the weighting coefficients α , β , and γ in the total reward.

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
Qwen2.5-VL-7B	44.6	68.5	68.8	65.3	76.4	74.3	56.4	50.1	54.8	62.1
$\alpha = 0.8$, $\beta = 0.2$, $\gamma = 1.0$	48.4	70.2	73.0	70.5	80.6	78.0	58.0	51.9	55.8	65.2
$\alpha = 0.8$, $\beta = 0.2$, $\gamma = 0.5$	49.2	69.7	72.8	69.8	78.5	79.0	57.5	50.8	55.8	64.8
$\alpha = 0.8$, $\beta = 0.2$, $\gamma = 0.0$	47.4	71.1	72.4	69.8	78.5	75.3	56.7	52.8	55.6	64.4
$\alpha = 0.9$, $\beta = 0.1$, $\gamma = 1.0$	48.9	70.8	73.3	69.6	79.1	78.0	57.5	51.4	55.1	64.9

I.2 RESULTS OF LARGER JIGSAW GRID SIZES AND CURRICULUM RL TRAINING

To further assess the scalability of AGILE beyond the 2×2 jigsaw setting, we conduct an additional reinforcement learning stage on 8K 3×3 jigsaw puzzles. This stage is applied after the initial 15.6K 2×2 RL training, forming curriculum-style optimization principle. As shown in Table 7, training on 3×3 jigsaw puzzles leads to further improvements across all 9 benchmarks. These results demonstrate that AGILE naturally generalizes to larger and more challenging jigsaw configurations when model capability allows, and further highlight that curriculum reinforcement learning on increasingly difficult grid sizes provides an effective path for continued scaling.

Table 7: Results of 3×3 Jigsaw RL Training.

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
Qwen2.5-VL-7B	44.6	68.5	68.8	65.3	76.4	74.3	56.4	50.1	54.8	62.1
+ 15.6K 2×2 RL	48.4	70.2	73.0	70.5	80.6	78.0	58.0	51.9	55.8	65.2
+ 15.6K 2×2 + 8K 3×3 RL	50.4	70.6	73.4	70.5	80.1	79.0	58.1	52.1	55.8	65.6

I.3 EXPERT TRAJECTORIES ABLATIONS

To disentangle the effect of expert cold-start trajectories and verify that AGILE’s improvements are not merely due to distillation, we conducted additional experiments without any expert trajectories. Specifically, we performed RL directly on the 15.6K 2×2 jigsaw dataset under two conditions: (1) the full action space, and (2) an ablated action space with `Crop`/`Zoom` removed. The results show that even in the absence of expert supervision, AGILE still delivers a +1.8% improvement over the Qwen2.5-VL-7B baseline, demonstrating that the performance gain arises from agentic visual interaction combined with verifiable RL, rather than imitation of Gemini 2.5 Pro trajectories.

Table 8: Performance of RL without Gemini 2.5 Pro expert trajectories and with ablated action spaces.

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
Qwen2.5-VL-7B	44.6	68.5	68.8	65.3	76.4	74.3	56.4	50.1	54.8	62.1
Cold-Start	46.2	68.4	71.0	68.4	77.5	76.7	55.7	49.8	54.0	63.1
Cold-Start + RL	48.4	70.2	73.0	70.5	80.6	78.0	58.0	51.9	55.8	65.2
Only RL (Full action space)	46.3	69.5	71.5	68.1	78.5	79.0	56.3	53.2	53.1	63.9
Only RL (No Crop/Zoom)	46.2	69.3	71.9	68.0	78.0	78.3	53.7	52.8	53.3	63.5

Furthermore, removing the `Crop`/`Zoom` operations leads to a clear decline in performance, underscoring the importance of AGILE’s interaction design in enabling effective fine-grained perceptual reasoning.

I.4 GENERAL QA DATASET DETAILS AND EXTENDED COMPARISON WITH JIGSAW TRAINING

We provide additional clarification regarding the composition of the general QA dataset. Our 20K QA corpus is constructed entirely from high-resolution, real-world visual scenes, consisting of 15K samples from DeepEyes (Zheng et al., 2025) and 5K samples from MMEureka (Meng et al., 2025).

1296 Table 9: Comparison between General QA and Jigsaw-based RL under equal training budgets (20K).
1297

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
Qwen2.5-VL-7B	44.6	68.5	68.8	65.3	76.4	74.3	56.4	50.1	54.8	62.1
20K General QA	50.5	68.9	72.4	69.5	82.7	78.3	54.8	51.3	54.2	64.7
10K General QA + 10K Jigsaw	51.6	69.8	73.0	70.6	81.7	79.0	57.3	51.2	55.7	65.5
20K Jigsaw-only	48.5	70.5	73.6	70.5	80.1	77.3	57.7	52.1	53.8	64.9

1302
1303
1304 These datasets span diverse perceptual and reasoning categories, including fine-grained attributes,
1305 spatial relationships, counting, and commonsense scene understanding.

1306 Accordingly, the comparison between jigsaw training and QA training is inherently fair: both training
1307 regimes operate on high-resolution, real-world, and complex perceptual inputs. To further
1308 strengthen this comparison, we additionally include the 20K pure jigsaw setting. As shown in Table
1309 9, jigsaw-based RL consistently outperforms high-resolution QA-based RL under the same training
1310 budget, demonstrating that AGILE’s performance gains do not stem from differences in input reso-
1311 lution, but instead from the structured, verifiable, and spatially grounded training signal introduced
1312 by the jigsaw task.

1313 I.5 IMPACT OF SCALING THE COLD-START (SFT) DATASET

1315 To understand whether training longer or scaling up the supervised cold-start phase affects perfor-
1316 mance, we expand the cold-start dataset to $1.5\times$ and $2\times$ its original size and repeated all experi-
1317 ments. As shown in Table 10, enlarging the SFT dataset leads to only marginal differences, with no
1318 signs of overfitting or meaningful performance degradation.

1319 Table 10: **Effect of scaling the cold-start (SFT) dataset size.** Increasing SFT data provides only
1320 marginal differences, indicating that most performance gains come from the RL stage.

Model	MME-RW	RWQA	HRB4K	HRB8K	VStar	MMVP	BLINK	HalBench	MMMU	Avg.
Qwen2.5-VL-7B	44.6	68.5	68.8	65.3	76.4	74.3	56.4	50.1	54.8	62.1
Cold-start (1.6K)	46.2	68.4	71.0	68.4	77.5	76.7	55.7	49.8	54.0	63.1
+RL	48.4	70.2	73.0	70.5	80.6	78.0	58.0	51.9	55.8	65.2
Cold-start (2.4K)	46.2	68.9	71.4	68.8	77.0	76.0	57.8	50.2	54.0	63.4
+RL	48.4	69.8	73.0	69.9	78.5	79.0	57.7	50.5	56.3	64.8
Cold-start (3.2K)	46.5	67.8	71.4	68.8	77.0	76.7	58.1	49.2	53.3	63.2
+RL	47.1	70.1	73.0	70.5	79.6	78.3	57.2	50.5	55.7	64.7

1321 In practice, the cold-start stage serves a very specific purpose: teaching the model the tool-use
1322 pattern (i.e., how to invoke the predefined Python APIs). Its goal is not to solve the jigsaw task fully.
1323 Consequently, the model’s performance is largely unaffected by moderate expansions of the SFT
1324 set. The primary source of generalization and performance improvement comes from the interactive
1325 RL stage, where the model learns to solve jigsaw puzzles through real environment feedback rather
1326 than through imitation.

1327 J USE OF LLMS

1328 Yes. We use LLMs solely to assist in language polishing and improving readability. All technical
1329 content, experiments, and analyses are conducted by the authors.