

Chain-of-Procedure: Hierarchical Visual-Language Reasoning for Procedural QA

Guanhua Chen^{1*}, Yutong Yao^{1*}, Shenghe Sun¹, Ci-Jun Gao³, Shudong Liu¹,
Lidia S. Chao¹, Feng Wan^{2,3}, Derek F. Wong^{1†}

¹NLP²CT Lab, Department of Computer and Information Science, University of Macau

²Department of Electrical and Computer Engineering, University of Macau

³Centre for Cognitive and Brain Sciences, University of Macau

{nlp2ct.guanhua, nlp2ct.yutong, nlp2ct.shenghe, cijun.gao, nlp2ct.shudong}@gmail.com
{derekfw, lidiase, fwan}@um.edu.mo

Abstract

Recent advances in vision-language models (VLMs) have achieved impressive results on standard image-text tasks, yet their potential for visual procedure question answering (VP-QA) remains largely unexplored. VP-QA presents unique challenges where users query next-step actions by uploading images for intermediate states of complex procedures. To systematically evaluate VLMs on this practical task, we propose ProcedureVQA, a novel multimodal benchmark specifically designed for visual procedural reasoning. Through comprehensive analysis, we identify two critical limitations in current VLMs: inadequate cross-modal retrieval of structured procedures given visual states, and misalignment between image sequence granularity and textual step decomposition. To address these issues, we present Chain-of-Procedure (CoP), a hierarchical reasoning framework that first retrieves relevant instructions using visual cues, then performs step refinement through semantic decomposition, and finally generates the next step. Experiments across six VLMs demonstrate CoP’s effectiveness, achieving up to 13% absolute improvement over standard baselines.

1 Introduction

Recent Large Language Models (LLMs) such as GPT-4 (Hurst et al., 2024) and Claude (Anthropic, 2025) have demonstrated strong performance across various multimodal understanding tasks (Ma et al., 2025), including visual question answering (VQA) (Qiu et al., 2024; Nguyen et al., 2025) and retrieval-augmented generation (RAG) (Chen et al., 2025a; Yu et al., 2025). However, their capability in visual procedure question answering (VP-QA), a critical competency for applications like furniture assembly, device troubleshooting,

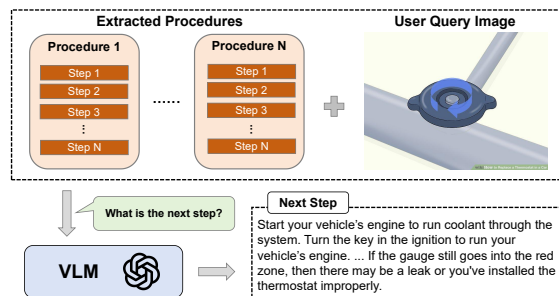


Figure 1: An example of the next-step prediction task.

and cooking assistance, remains largely underexplored. We formally define this task as a multimodal selection problem: given an image representing an intermediate state of a procedure for a complete task workflow, the model must reason about the progress and identify the correct next step (a major action in the procedure) from a set of candidate steps. As illustrated in Figure 1, when a user uploads an image of a partially disassembled radiator cap, the system must align this visual state with the specific step of the procedure to predict the immediate next action. This task requires fine-grained visual-textual alignment and hierarchical reasoning, presenting unique challenges that surpass those of conventional image-text tasks.

While current Vision-Language Models (VLMs) focus on static visual QA (Bazi et al., 2023; Guo et al., 2023) and text-guided image reasoning (Kelly et al., 2024; Zhou et al., 2024), procedural understanding requires a dynamic alignment between visual queries and structured plans. Existing benchmarks either focus on single-image recognition (Yang et al., 2021) or rely on pre-aligned procedural data (Lin et al., 2020), neglecting the core challenge of grounding evolving visual states to hierarchical procedure. Unlike text-guided document reasoning (Yang et al., 2018), it is significantly more challenging for VLMs to align visual states with discrete textual instructions, particularly

*Equal contribution

†Corresponding authors.

when the granularity of visual evidence does not perfectly match the text.

To fill these gaps, we construct **ProcedureVQA**, a novel benchmark for visual procedural step understanding. We reconstruct a released dataset proposed by [Chen et al. \(2024\)](#) into a multimodal benchmark spanning five common procedural domains (home repair, cooking, device setup, crafts, and health care). We evaluate six state-of-the-art VLMs on end-to-end next-step prediction and observe suboptimal accuracy (25.1-62.5%). Through error analysis, we identify a critical "granularity mismatch": high-level instructions often encompass multiple implicit atomic actions, making direct visual alignment difficult for current models.

Consequently, we propose **Chain-of-Procedure (CoP)**, a hierarchical reasoning framework designed to bridge this gap. Unlike prior retrieval-augmented approaches ([Panda et al., 2024](#)) that treat procedures as flat text, CoP operates in three distinct phases to align terminology with visual evidence. First, it employs VLMs to filter broadly relevant candidate procedures. Second, it dynamically decomposes the coarse procedure into fine-grained sub-steps to match the visual state granularity. Finally, it predicts the current sub-step action and infers the precise next step. By decoupling procedure retrieval, step decomposition, and visual-textual alignment, CoP significantly strengthens VLMs' procedural reasoning capabilities. We evaluate CoP on six VLMs using our ProcedureVQA benchmark, and experimental results demonstrate that CoP improves the performance of next-step prediction by 1.7-17.9% in BertScore and 2.2-13% in LLM-Score over baselines.

2 Related Work

2.1 Procedural Reasoning and Planning

Procedural reasoning involves understanding goal-oriented sequences, tracking progress, and anticipating future actions. Early text-based benchmarks like PIQA ([Bisk et al., 2020](#)) focused on physical commonsense reasoning, while ProPara ([Dalvi et al., 2018](#)) and OpenPI ([Tandon et al., 2020](#)) introduced entity state tracking within procedural texts. The field expanded with large-scale instructional video datasets such as COIN ([Tang et al., 2019](#)), HowTo100M ([Miech et al., 2019](#)), and Ego4D ([Grauman et al., 2022](#)), enabling tasks like weakly supervised procedure learning ([Zhukov et al., 2019](#)). RecipeQA ([Yagcioglu et al., 2018](#))

further explored multimodal comprehension over recipe procedures. Recent work has pivoted towards multimodal procedural planning: [Lu et al. \(2024\)](#) utilizes dual text-image prompting to generate coherent plans, while [Glória-Silva et al. \(2024\)](#) grounds instructional plans within visual environments. However, these approaches predominantly rely on continuous video temporal cues, sequential texts, or paired image-text contexts. In contrast, ProcedureVQA targets snapshot-based procedural reasoning, requiring models to infer implicit progress and predict next-step actions from a single static intermediate visual state.

2.2 Multimodal Learning for Procedure Task

The application of VLMs to specific procedure domains, particularly cooking and maintenance, has garnered significant attention. [Kafle and Kanan \(2017\)](#) and [Yagcioglu et al. \(2018\)](#) laid the groundwork with datasets like RecipeQA. More recent efforts have developed specialized models for these tasks. CookingQA ([Khilji et al., 2021](#)) addresses QA based on ingredients, while FoodLMM ([Yin et al., 2023](#)) and LLaVA-Chef ([Mohbat and Zaki, 2024](#)) leverage large multimodal models to generate comprehensive recipes and cooking procedures. Furthermore, retrieval-based approaches have been explored to enhance recipe understanding ([Song et al., 2023](#)). Despite these advancements, a key limitation persists: the misalignment between the granularity of visual states and textual procedures. While general VLMs ([Liu et al., 2023](#); [Zhao et al., 2024](#)) show promise, they struggle with the structured retrieval required for complex procedures. Unlike previous works that focus on end-to-end generation or simple alignment, our CoP framework explicitly tackles the granularity mismatch through semantic step decomposition, enabling more accurate next-step predictions.

3 ProcedureVQA

We present a novel benchmark reconstructed from the WikiHow portion of the CoMM dataset released by [Chen et al. \(2024\)](#). Our framework targets the broad spectrum of real-world procedural tasks, ranging from technical vehicle maintenance to daily lifestyle skills.

3.1 Problem Formulation

To rigorously define the task and standardize terminology, we formalize VP-QA as follows. A procedure $\mathcal{P} = \{S_1, S_2, \dots, S_L\}$ is a complete task

Metric	Train	Test	Total
Total Samples	1,939	2,844	4,783
Avg. Length (tokens)	5.9	5.7	5.8
Step Length Std	7.1	4.8	5.9
Shortest Sample	3	3	3
Longest Sample	45	22	33.5
Domain Distribution			
Cars & Other Vehicles	476	509	985
Computers & Electronics	696	631	1,327
Hobbies & Crafts	551	533	1,084
Sports & Fitness	216	500	716
Work World	0	671	671

Table 1: The statistics of the ProcedureVQA dataset consist of benchmarks and a training set.

workflow composed of L sequential steps (major actions). Given a visual query v_t showing the intermediate state after completing step S_t , the model must: (1) retrieve the correct procedure \mathcal{P}^* from a candidate set $\mathcal{C}_p = \{\mathcal{P}_1, \dots, \mathcal{P}_N\}$ containing one correct procedure and $N - 1$ negative procedures, (2) localize v_t to the current step S_t within \mathcal{P}^* , and (3) predict the content of the next step S_{t+1} . This formulation tests cross-procedure retrieval, visual state grounding, and next-step reasoning under granularity mismatch between coarse-grained steps and fine-grained visual evidence.

3.2 Data Construction

Our reconstruction framework transforms raw procedural data into a multimodal reasoning benchmark through domain-specific processing.

Data Domains. We strategically selected five domains to ensure task diversity while maintaining domain coherence: (1) *Cars & Vehicles* covers maintenance tasks like oil checks; (2) *Computers & Electronics* includes hardware and software tutorials; (3) *Hobbies & Crafts* features visual guides on arts; (4) *Sports & Fitness* provides exercise guides; and (5) *Work World* covers professional skills. This selection balances everyday domains with technically complex ones. The example of each domain is shown in Appendix A.1.

Granularity Simulation via Step Fusion. A key challenge in procedural VQA is the inconsistency between visual and textual granularity. To simulate this real-world variation, we apply controlled step fusion to the raw textual instructions. With

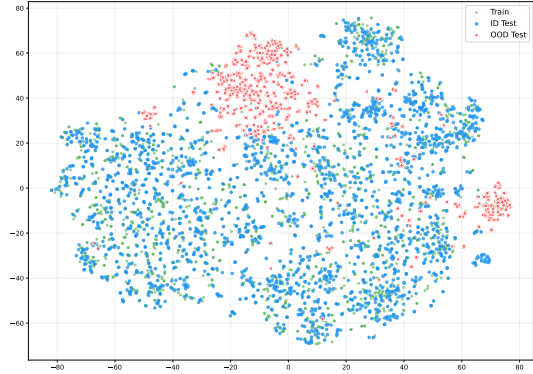


Figure 2: The t-SNE of the training data and in-domain (ID)/out-of-domain (OOD) test data.

a 50% probability, two consecutive atomic steps are merged into a single coarse-grained instruction. This approach, inspired by Gao et al. (2022) and Dalvi et al. (2018), creates a realistic mismatch where a single step may correspond to multiple visual states. This design is crucial for testing a model’s robustness to inconsistent step resolutions and motivates the need for step decomposition. We also conduct a sensitive analysis of this probability in Appendix A.2.

Negative Sampling and Data Structure. To create challenging distractors, we encode all images into 512-dimensional vectors via ResNet50 (He et al., 2016). For each input image v_t , we curate hard-negative candidates through intra-domain cosine similarity searches. We select visually proximate but procedurally irrelevant instructions as distractors to prevent the model from relying on simple visual shortcuts. Each data instance is finally constructed as a tuple:

$$D = (v_t, \mathcal{C}, y) \quad (1)$$

where v_t is the input image, $\mathcal{C} = \{c_{pos}, c_{neg}^{(1)}, c_{neg}^{(2)}\}$ is the set of three candidate instructions (one positive, two hard negatives), and y is the index of the ground truth. This structure ensures that the benchmark rigorously evaluates the VLM’s ability to perform precise image-to-text alignment within a procedural context.

3.3 Data Statistics

Our dataset contains 4,783 samples. To evaluate VLM fine-tuning, we split the data into training and benchmark sets: all “Work World” 671 samples are reserved as out-of-distribution (OOD) tests, while other domains use step-length-stratified splits for

Model	Zero-Shot	CoT
Qwen2.5-VL-7B	25.1	26.6
Qwen2.5-VL-72B	43.4	41.3
GPT-4o-mini	34.4	42.7
GPT-4o	62.5	51.8
Gemini-2.5-Flash	44.7	42.6
Claude-3-5-Sonnet	47.5	55.6

Table 2: The results (%) on the benchmarks of different VLMs. **Bold** indicate the best result for each setting.

balanced complexity. As Table 1 shows, training and in-domain (ID) benchmarks share similar lexical complexity, but training exhibits higher step variance, retaining long-tail sequences.

We measure semantic overlap using **all-roberta-large-v1** (Reimers and Gurevych, 2019). Benchmark-training pairs show a median cosine similarity of 0.64, balancing transfer potential and memorization resistance. The t-SNE visualizations in Figure 2 confirm structural alignment: training (green) and ID test data (blue) overlap significantly, while OOD samples (red) form distinct clusters. This design rigorously tests the learning and generalization capabilities of VLMs on VP-QA task.

4 Benchmarking and Analysis

This section evaluates the capabilities of powerful VLMs on VP-QA task using the **ProcedureVQA** benchmark. We first introduce the experimental setup and evaluation metrics, followed by a comprehensive analysis of benchmark results and five decomposed fundamental sub-tasks to enable systematic analysis of VLM behaviors.

4.1 Models and Evaluation Metrics

We evaluate two categories of VLMs: open-source (Qwen2.5-VL-7B and Qwen2.5-VL-72B) (Bai et al., 2023) and closed-source VLMs (GPT-4o-mini, GPT-4o, Gemini-2.5-Flash, Claude-3-5-Sonnet) (OpenAI, 2023; Reid et al., 2024; Priyan-shu et al., 2024). The implementation details of all models are explained in Appendix A.3. For evaluation, we employ **Accuracy** as the primary metric to measure whether the predicted next-step action matches the ground truth. This straightforward metric allows us to assess the model’s fundamental capability on this task. Two prompting strategies are compared: zero-shot prompting and chain-of-thought (CoT) prompting (Wei et al., 2022a).

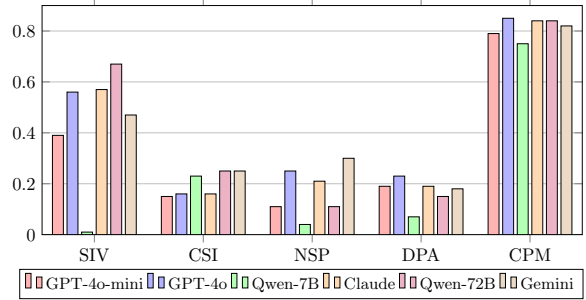


Figure 3: The experimental results (Accuracy) of our designed five different sub-tasks on six powerful VLMs.

4.2 Performance Across Different VLMs

Table 2 presents the benchmarking results across zero-shot and CoT prompting. Closed-source VLMs consistently outperform open-source models, with GPT-4o achieving the highest zero-shot accuracy of 62.5%. CoT prompting shows mixed effects on accuracy: while it improves performance for some models, Qwen2.5-VL-72B experiences a 2.1% accuracy decline, potentially due to over-elaboration in step generation. Model scale plays a critical role in performance, as the 72B variant of Qwen2.5-VL demonstrates an 18.3% accuracy advantage over its 7B counterpart. Despite these advancements, even the most powerful VLMs struggle to exceed 65% accuracy, highlighting persistent challenges in image-guided procedural reasoning.

4.3 Decomposing Task-Specific Abilities

To systematically investigate procedural reasoning capabilities in VLMs, we design five diagnostic sub-tasks with increasing complexity (detailed task formulations in Appendix A.4). Results are shown in Figure 3. We begin with Shuffled Procedure Verification (SIV), where models determine if procedure steps are correctly ordered—Qwen2.5-VL-72B achieves 67.2%, surpassing Claude-3.5-Sonnet. Current Step Identification (CSI) requires recognizing which step corresponds to a given visual state, testing temporal grounding. Next-Step Prediction (NSP) extends this by predicting the subsequent action, showing a 3% performance drop from CSI, indicating increased difficulty in forward temporal reasoning. Disordered Procedure Adaptation (DPA) simulates real-world scenarios by requiring step continuation from shuffled procedures. Finally, Cross-Procedure Matching (CPM) evaluates whether an image-procedure pair are semantically aligned, where GPT-4o achieves state-of-the-art performance (84.5%).

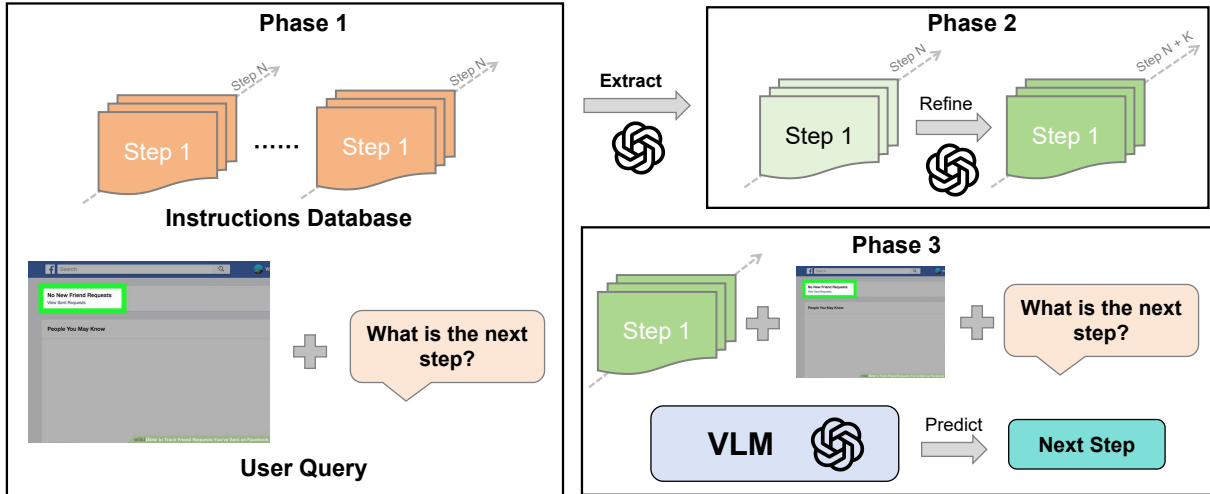


Figure 4: The overview of our proposed framework.

Three critical findings emerge. First, VLMs excel at structural validation tasks (SIV and CPM: 78.9% and 75.6% average accuracy) but struggle with procedural reasoning tasks (NSP and DPA), revealing that multimodal alignment capabilities exceed causal reasoning proficiency. Second, the 23.9% performance gap between CSI and NSP highlights VLMs’ fundamental difficulty in reasoning about future actions versus recognizing current states. Third, closed-source models show 15.2% smaller accuracy degradation on DPA compared to open-source alternatives, demonstrating superior robustness to procedural perturbations. These findings motivate CoP’s design to explicitly address the challenge of VP-QA task.

5 Method

Building upon the insights from Section 4.3, we present Chain-of-Procedure (CoP), a framework that redefines VP-QA through structured hierarchical reasoning. Figure 4 depicts CoP coordinates VLMs across three tightly coupled phases: context-aware procedure retrieval, step-wise procedural decomposition, and next-step prediction. This framework tackles key limitations of monolithic VLMs by explicitly decoupling three core aspects: cross-modal alignment, fine-grained procedural reasoning, and temporal state modeling.

5.1 Context-Aware Procedure Retrieval

The initial phase resolves the critical challenge of extracting the procedure relevant to the input image. Instead of relying on static embeddings, we operationalize this retrieval as an iterative cross-modal matching process. Motivated by Section

4.3, we design a verification loop where the VLM iteratively evaluates candidate procedures against the input image. Specifically, consider a database containing N procedures: [Procedure 1], [Procedure 2], ..., [Procedure N]. For each procedure, the VLM computes a multimodal alignment score using the format "Task Instruction + [Procedure] + Image Query", where the instruction prompts the VLM to provide a score addressing the core question: "Does this procedure accurately reflect the image’s content?" The highest-scoring procedure is dynamically selected and used in Phase 2 of our method, replacing static retrieval with an adaptive grounding process that leverages VLMs’ intrinsic cross-modal reasoning while ensuring computational efficiency.

5.2 Step-Wise Procedural Decomposition

A core innovation of CoP lies in its explicit modeling of procedural granularity, a dimension severely underexploited by conventional VLMs. We design a decomposition module that reconstructs atomic procedural steps from potentially over-grouped actions. The VLM first identifies composite steps through syntactic patterns and semantic meaning, then splits them into temporally ordered atomic actions while keeping unnecessary steps unchanged. Importantly, this process preserves the original procedure’s causal structure while recovering intermediate states omitted in aggregated steps.

5.3 Next Step Prediction

In the final phase, our method establishes cross-modal alignments between the input image and the corresponding procedural step. As demonstrated

Model	BertScore	LLM-Score
ProcedureVQA		
Qwen2.5-72B	30.1	40.6
Qwen2.5-72B (Ours)	51.7	49.6
GPT-4o-mini	38.6	40.2
GPT-4o-mini (Ours)	48.3	48.7
GPT-4o	43.5	54.2
GPT-4o (Ours)	61.4	67.2
Gemini-2.5	46.4	43.6
Gemini-2.5 (Ours)	48.1	47.3
Claude-3-5	44.7	45.6
Claude-3-5 (Ours)	48.8	47.8
RecipeQA		
Qwen2.5-72B	23.9	35.0
Qwen2.5-72B (Ours)	26.5	32.3
GPT-4o-mini	20.7	31.9
GPT-4o-mini (Ours)	22.0	29.6
GPT-4o	38.7	48.6
GPT-4o (Ours)	46.6	49.7
Gemini-2.5	42.8	47.2
Gemini-2.5 (Ours)	37.8	44.6
Claude-3-5	16.8	26.3
Claude-3-5 (Ours)	19.3	33.6

Table 3: The main results (%) of our CoP framework on the ProcedureVQA and RecipeQA datasets. **Bold** numbers indicate the better result for each baseline.

in Section 4.3, VLMs exhibit superior capability in identifying the current state of an input. Thus, our approach first reasons about the input’s present state before inferring subsequent steps, rather than directly predicting the next step. Specifically, the VLM implicitly computes a multimodal alignment score between the image and each decomposed step. The model then outputs the identified current state, and implicitly predicts the next step.

Through these three phases, we establish a new paradigm for hierarchical procedural reasoning for VLMs. All the detailed instruction templates for VLMs are explained in Appendix A.7.

6 Experiments and Analysis

6.1 Setup

To validate the effectiveness of the proposed CoP framework, we conducted comprehensive experi-

Model	BertScore	LLM-Score
Baseline	59.7	67.4
End2End (Ours)	61.2	69.2
Pipeline (Ours)	63.6	72.1

Table 4: The results of fine-tuning Qwen2.5-VL-7B on ProcedureVQA dataset with three different setting. **Bold** numbers indicate the best result.

ments across six VLMs following Section 4. Moreover, we evaluate the performance of the fine-tuned Qwen2.5-VL-7B model. Besides our proposed ProcedureVQA dataset, we also evaluate our CoP on the RecipeQA dataset (Yagcioglu et al., 2018) to verify the generalizability of our CoP. For deep analysis, we employ two different evaluation metrics: BERTScore (Zhang et al., 2019) and LLM-Score, which are detailed in Appendix A.5.

6.2 Main Results

Table 3 show the main results of our CoP framework on two benchmarks. On the ProcedureVQA dataset, our method achieves consistent improvements across all models. Notably, Qwen2.5-72B with CoP obtains a 21.6% absolute improvement in BERTScore and 9.0% in LLM-Score. GPT-4o also shows substantial gains, reaching 61.4% BERTScore and 67.2% LLM-Score. These results demonstrate that our framework effectively enhances the alignment between visual understanding and procedural reasoning.

On the RecipeQA dataset, the improvements are more varied. GPT-4o and Claude-3-5 benefit clearly from our method, with GPT-4o achieving 7.9% improvement in BERTScore and Claude-3-5 gaining 7.3% in LLM-Score. However, Gemini-2.5 shows decreased performance on this dataset. We attribute this to the inherent differences between the two benchmarks. RecipeQA contains more diverse and less structured cooking instructions compared to ProcedureVQA. Since Gemini-2.5 already achieves strong baseline performance on RecipeQA, the additional constraints from CoP may limit its flexibility in generating varied responses. Similarly, the slight LLM-Score decreases for Qwen2.5-72B and GPT-4o-mini suggest that these LLMs may over-adapt to the structured output format, which is less suitable for the open-ended nature of recipe descriptions.

We also compare different fine-tuning strategies in Table 4. Specifically, we examine the perfor-

Model	BertScore	LLM-Score
ProcedureVQA		
GPT-4o	43.5	54.2
GPT-4o (Phase 1)	47.3	62.6
GPT-4o (Phase 1&2)	47.1	52.6
GPT-4o (Phase 1&3)	59.8	66.8
GPT-4o (Ours)	61.4	67.2
Recipe		
GPT-4o	38.7	48.6
GPT-4o (Phase 1)	38.8	47.7
GPT-4o (Phase 1&2)	38.5	47.1
GPT-4o (Phase 1&3)	45.5	48.7
GPT-4o (Ours)	46.6	49.7

Table 5: The ablation study of our method on two datasets with GPT-4o. **Bold** numbers indicate the better result for each baseline.

mance of our step-wise LoRA fine-tuning against fine-tuning with CoT (Wei et al., 2022b) style synthetic data that follows our pipeline. The results show that both fine-tuning methods outperform the direct fine-tuning baseline. While the CoT fine-tuning approach offers faster inference speed, it achieves lower performance compared to our step-wise fine-tuning method. Details of training data construction are provided in Appendix A.9.

6.3 Ablation Studies

To investigate the necessity of each component in our framework, we conduct an ablation study on the GPT-4o model with two datasets. The experimental results are shown in Table 5. Our analysis progressively integrates components from phase 1 through phase 3. The results show that combining all three phases achieves the best performance: 61.4 BertScore and 67.2 LLM-Score on ProcedureVQA, and 46.6 BertScore and 49.7 LLM-Score on RecipeQA. Interestingly, adding phase 2 to phase 1 causes a performance drop (from 62.6 to 52.6 LLM-Score on ProcedureVQA), suggesting that phase 2 may introduce noise during procedure decomposition. However, phase 3 effectively addresses this issue: phase 1&3 substantially outperforms phase 1 alone, and our complete framework further improves upon Phase 1&3. This demonstrates that while phase 2 alone may be suboptimal, the combination of all three phases can achieve optimal performance.

Model	BertScore	LLM-Score
Baseline	43.5	54.2
CLIP-Full	29.1	34.5
CLIP-P1	47.5	51.6
CLIP-P3	37.9	45.0
Ours	61.4	67.2

Table 6: The results of replacing LLM with CLIP Embedding model for different phases on GPT-4o. **Bold** means the best result.

6.4 Comparison with Retrieval Models

To further validate the necessity of our approach, we compare against CLIP¹ (Radford et al., 2021), a strong multimodal retrieval model that maps image and text into a shared embedding space for similarity computation. We evaluate CLIP in three configurations: (1) CLIP-P1: replacing only Phase 1 with CLIP-based retrieval while keeping LLM-based generation; (2) CLIP-P3: replacing only Phase 3 with CLIP-based matching; and (3) CLIP-Full: a fully retrieval-based pipeline where CLIP retrieves the correct procedure, matches the current step, and outputs the next step directly.

As shown in Table 6, all CLIP-based configurations underperform our method. While CLIP-P1 achieves relatively better results than other CLIP variants by leveraging LLM execution, it still shows a substantial gap compared to the baseline. These results demonstrate both the necessity of our proposed CoP and the inherent difficulty of this task. Even though the retrieval models can reasonably match the correct procedure, fine-grained alignment at the step level remains challenging.

6.5 Human Evaluation

To rigorously evaluate the effectiveness of our framework, we conduct a human evaluation comparing our method against two baselines: GPT-4o and a fine-tuned Qwen2.5-VL-7B. We randomly selected 20 instances from the benchmark dataset for these two VLMs separately and tasked three computer science students with pairwise comparisons. Annotators independently judged outputs based on accuracy (faithfulness to ground truth) and conciseness (brevity without redundancy), categorizing results as “Better,” “Equivalent,” or “Worse” relative to each baseline.

As shown in Table 7, our framework demon-

¹<https://huggingface.co/openai/clip-vit-large-patch14>

Model	Win	Equal	Loss
Qwen2.5-VL-7B	65%	2%	33%
GPT-4o	48%	20%	31%

Table 7: The results of human evaluation on fine-tuned Qwen2.5-VL-7B and GPT-4o. **Bold** numbers indicate the better result for each VLM.

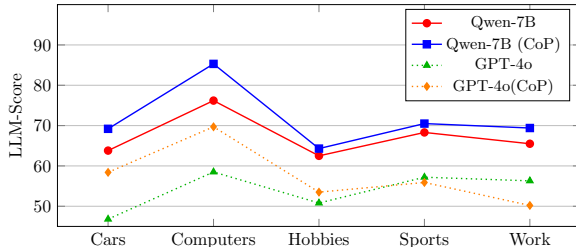


Figure 5: The LLM-score of fine-tuned Qwen2.5-VL-7B and GPT-4o on five domains separately.

strates statistically significant superiority across both metrics. Compared to the GPT-4o baseline, it achieves a 48% “Win” rate with 20% “Equal” rate. Against Qwen2.5-VL-7B baseline, it attains a 65% “Win” rate, which is large greater than the “Worse” rate. Inter-annotator agreement (Fleiss $k = 0.51$) confirms moderate consensus. These results validate that our CoP framework substantially improves both precision and clarity in next-step generation compared to existing approaches.

6.6 Performance Across Data Categories

To analyze the robustness and generalization capabilities of CoP, we conduct experiments to compare its performance against baselines across different domains and step distribution. Figure 5 illustrates the results on five distinct domains (as described in Section 3.3), where we employ LLM-Score as the evaluation metric due to its ability to jointly account for accuracy and conciseness. For the fine-tuned Qwen2.5-VL-7B, our framework consistently outperforms the baseline across all five domains. However, for GPT-4o, our approach underperforms the baseline in the “Sports and Fitness” and “Work World” domains. Considering that the fine-tuned Qwen2.5-VL-7B only marginally surpasses the baseline in these two domains, we think that “Sports and Fitness” and “Work World” procedures exhibit lower sensitivity to step order compared to the other three domains. Consequently, the second phase of our framework may introduce unintended noise or over-constraints, thereby nega-

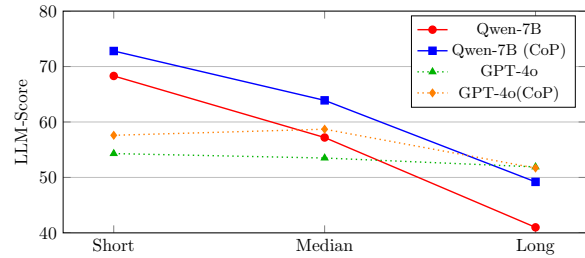


Figure 6: The average LLM-score of fine-tuned Qwen2.5-VL-7B and GPT-4o on different step lengths.

tively impacting final performance.

Figure 6 demonstrates the performance of our method versus baselines on procedures of varying step lengths. Our approach achieves superior results across all configurations. Empirically, procedures with more steps inherently demand higher comprehension complexity. These results suggest that powerful VLMs like GPT-4o can effectively handle difficult procedures in this task. In contrast, smaller-size VLMs such as Qwen2.5-VL-7B struggle to learn long procedural patterns. Nonetheless, our framework enhances their performance through systematic fine-tuning and inference following our proposed framework. Additionally, we compare the impact of different numbers of negative samples and sampling strategies in Appendix A.6.

6.7 Error Analysis

To systematically evaluate the effectiveness of our method, we conduct an error analysis comparing model capabilities by implementing our three-phase framework on GPT-4o-mini and GPT-4o across five domains. Phase 1 performance is measured through procedure retrieval accuracy, Phase 2 through BLEU score alignment between decomposed and original procedures, and Phase 3 via LLM-Score of next-step prediction. The normalization scores are shown in Figure 7.

As expected, GPT-4o consistently outperformed its smaller counterpart. Both models achieved over 90% accuracy in Phase 1 procedure retrieval, demonstrating robust image-text alignment capabilities. However, significant performance gaps emerged in Phase 2 decomposition quality despite equivalent retrieved procedures, revealing critical differences in procedural understanding. This capability gap directly impacts Phase 3 outcomes in procedure-sensitive domains like “Computers”, “Cars”, and “Hobby”, where GPT-4o’s superior decomposition quality translates to 9.7-14.1% higher

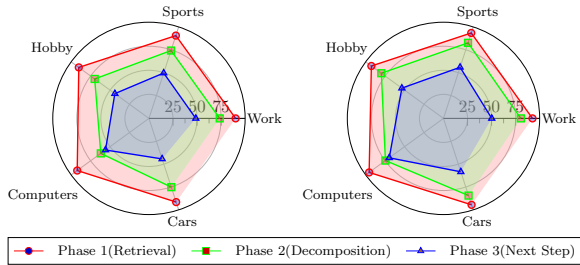


Figure 7: The results of three phases of our CoP on five domains on GPT-4o-mini (**Left**) and GPT-4o (**Right**).

LLM-Scores compared to GPT-4o-mini. Conversely, in knowledge-dependent domains, such as “Sports” and “Work”, Phase 3 performance shows weaker correlation with decomposition quality, with GPT-4o achieving only marginal improvements over GPT-4o-mini in “Work” task. This explains the baseline performance gap observed in Figure 5 for these domains, suggesting that our method’s effectiveness becomes constrained when procedural reasoning plays a secondary role to pre-existing world knowledge. The findings emphasize the dual dependence of visual procedural reasoning on both explicit procedure alignment and implicit procedural comprehension capabilities. More discussion will be shown in Appendix A.8.

7 Conclusion

In this work, we investigate the underexplored capability of VLMs in the visual procedural reasoning task. We propose a novel multimodal benchmark designed for evaluating procedural reasoning. Through systematic analysis, we propose a CoP framework that synergizes cross-modal procedure retrieval, context-aware step decomposition, and the next step prediction. Experimental results show that our approach significantly outperforms baselines, revealing critical insights about cross-modal alignment and procedural reasoning. In the future, we will explore more efficient workflow adaptation for domain-specific procedures.

Limitations

While our framework demonstrates significant improvements in visual procedural reasoning, we identify two key limitations. First, the multi-phase CoP framework costs around 2 times tokens compared to baselines, though our empirical results demonstrate that the substantial accuracy improvements justify this additional cost. The baseline costs average 2168 tokens per data, while our CoP

method costs average 4526 tokens per sample. Second, the system remains susceptible to error propagation due to its dependence on off-the-shelf VLMs for procedure parsing during the retrieval stage. Notably, the performance is particularly sensitive to the accuracy of relevant procedure extraction, which critically impacts the final results.

Acknowledgments

This work was supported in part by the Science and Technology Development Fund of Macau SAR (Grant Nos. FDCT/0007/2024/AKP, EF2024-00185-FST), the UM and UMDf (Grant Nos. MYRG-GRG2024-00165-FST-UMDF, MYRG-GRG2025-00236-FST), the Tencent AI Lab Rhino-Bird Research Program (Grant No. EF2023-00151-FST), the Stanley Ho Medical Development Foundation (Grant No. SHMDF-AI/2026/001), and the National Natural Science Foundation of China (Grant No. 62266013).

References

- Anthropic. 2025. [Claude 3.7 sonnet system card](#).
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.
- Yakoub Bazi, Mohamad Mahmoud Al Rahhal, Laila Bashmal, and Mansour Zuair. 2023. Vision-language model for visual question answering in medical imagery. *Bioengineering*, 10(3):380.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34.
- Guanhua Chen, Yutong Yao, Lidia S. Chao, Xuebo Liu, and Derek F. Wong. 2025a. *SGIC: A self-guided iterative calibration framework for RAG*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28357–28370, Vienna, Austria. Association for Computational Linguistics.
- Guanhua Chen, Yutong Yao, Ci-Jun Gao, Lidia S Chao, Feng Wan, and Derek F Wong. 2025b. Not all lora parameters are essential: Insights on inference necessity. *arXiv preprint arXiv:2503.23360*.
- Wei Chen, Lin Li, Yongqi Yang, Bin Wen, Fan Yang, Tingting Gao, Yu Wu, and Long Chen. 2024. Comm: A coherent interleaved image-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2406.10462*.

- Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*.
- Bhavana Dalvi and 1 others. 2018. [Tracking state changes in procedural text: a challenge dataset and models for process paragraph comprehension](#). *arXiv preprint arXiv:1805.06975*.
- Shen Gao, Haotong Zhang, Xiuying Chen, Rui Yan, and Dongyan Zhao. 2022. [Summarizing procedural text: Data and approach](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Diogo Glória-Silva, David Semedo, and João Magalhães. 2024. [Show and guide: Instructional-plan grounded vision and language model](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 21371–21389. Association for Computational Linguistics.
- Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, and 66 others. 2022. [Ego4d: Around the world in 3, 000 hours of egocentric video](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 18973–18990. IEEE.
- Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. 2023. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10867–10877.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- OpenAI Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mkadry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alexander Kirillov, Alex Nichol, Alex Paino, and 397 others. 2024. [Gpt-4o system card](#).
- Kushal Kafle and Christopher Kanan. 2017. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163:3–20.
- Chris Kelly, Luhui Hu, Bang Yang, Yu Tian, Deshun Yang, Cindy Yang, Zaoshan Huang, Zihao Li, Jiayin Hu, and Yuexian Zou. 2024. [Visiongpt: Vision-language understanding agent using generalized multimodal framework](#). *arXiv preprint arXiv:2403.09027*.
- Abdullah Faiz Ur Rahman Khilji, Riyanka Manna, Sahinur Rahman Laskar, Partha Pakray, Dipankar Das, Sivaji Bandyopadhyay, and Alexander F. Gelbukh. 2021. [Cookingqa: Answering questions and recommending recipes based on ingredients](#). *Arabian Journal for Science and Engineering*, 46:3701–3712.
- Angela S Lin, Sudha Rao, Asli Celikyilmaz, Elnaz Nouri, Chris Brockett, Debadeepta Dey, and Bill Dolan. 2020. A recipe for creating multimodal aligned datasets for sequential tasks. *arXiv preprint arXiv:2005.09606*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.
- Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Wang, and William Yang Wang. 2024. [Multimodal procedural planning via dual text-image prompting](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, Findings of ACL, pages 10931–10954. Association for Computational Linguistics.
- Jingkun Ma, Runzhe Zhan, Yang Li, Di Sun, Hou Pong Chan, Lidia S. Chao, and Derek F. Wong. 2025. [Vi-saidmath: Benchmarking visual-aided mathematical reasoning](#). In *1st Workshop on VLM4RWD @ NeurIPS 2025*.
- Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
- Fnu Mohbat and Mohammed J Zaki. 2024. [Llava-chef: A multi-modal generative model for food recipes](#). *arXiv preprint arXiv:2408.16889*.
- Thuy Nguyen, Dang Nguyen, Hoang Nguyen, Thuan Luong, Long Hoang Dang, and Viet Dac Lai. 2025. [Owlviz: An open-world benchmark for visual question answering](#). *arXiv preprint arXiv:2503.07631*.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- OpenAI. 2025. [Introducing gpt-4.1 in the api](#).

- Pranoy Panda, Ankush Agarwal, Chaitanya Devaguptapu, Manohar Kaul, and Prathosh Ap. 2024. [HOLMES: Hyper-relational knowledge graphs for multi-hop question answering using LLMs](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13263–13282, Bangkok, Thailand. Association for Computational Linguistics.
- Aman Priyanshu, Yash Maurya, and Zuofei Hong. 2024. [AI governance and accountability: An analysis of anthropic’s claude](#). *CoRR*, abs/2407.01557.
- Jielin Qiu, Andrea Madotto, Zhaojiang Lin, Paul A Crook, Yifan Ethan Xu, Xin Luna Dong, Christos Faloutsos, Lei Li, Babak Damavandi, and Seungwhan Moon. 2024. Snapntell: Enhancing entity-centric visual question answering with retrieval augmented multimodal llm. *arXiv preprint arXiv:2403.04735*.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmlR.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew M. Dai, Katie Millican, Ethan Dyer, Mia Glaese, Thibault Sottiaux, Benjamin Lee, and 34 others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). *CoRR*, abs/2403.05530.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.
- Fangzhou Song, Bin Zhu, Yanbin Hao, and Shuo Wang. 2023. [Enhancing recipe retrieval with foundation models: A data augmentation perspective](#). In *European Conference on Computer Vision*.
- Niket Tandon, Keisuke Sakaguchi, Bhavana Dalvi, Dheeraj Rajagopal, Peter Clark, Michal Guerquin, Kyle Richardson, and Eduard Hovy. 2020. A dataset for tracking entities in open domain procedural text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6408–6417.
- Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. 2019. [COIN: A large-scale dataset for comprehensive instructional video analysis](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 1207–1216. Computer Vision Foundation / IEEE.
- Gemini Team. 2023. [Gemini: A family of highly capable multimodal models](#). *CoRR*, abs/2312.11805.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022a. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Semih Yagcioglu, Aykut Erdem, Erkut Erdem, and Nazli Ikizler-Cinbis. 2018. [RecipeQA: A challenge dataset for multimodal comprehension of cooking recipes](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1358–1368, Brussels, Belgium. Association for Computational Linguistics.
- Yue Yang, Artemis Panagopoulou, Qing Lyu, Li Zhang, Mark Yatskar, and Chris Callison-Burch. 2021. Visual goal-step inference using wikihow. *arXiv preprint arXiv:2104.05845*.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.
- Yuehao Yin, Huiyan Qi, Bin Zhu, Jingjing Chen, Yuhang Jiang, and Chong-Wah Ngo. 2023. [Foodllm: A versatile food assistant using large multi-modal model](#). *IEEE Transactions on Multimedia*, 27:6949–6961.
- Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2025. [Visrag: Vision-based retrieval-augmented generation on multi-modality documents](#). In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Xiangyu Zhao, Xiangtai Li, Haodong Duan, Haiyan Huang, Yining Li, Kai Chen, and Hua Yang. 2024. Mg-llava: Towards multi-granularity visual instruction tuning. *arXiv preprint arXiv:2406.17770*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. **Llamafactory: Unified efficient fine-tuning of 100+ language models**. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2024. **Navgpt-2: Unleashing navigational reasoning capability for large vision-language models**. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VII, Lecture Notes in Computer Science*, pages 260–278. Springer.

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David F. Fouhey, Ivan Laptev, and Josef Sivic. 2019. **Cross-task weakly supervised learning from instructional videos**. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3532–3540.

A Appendix

A.1 Example of Different Domain Benchmark

To demonstrate the structure of our ProcedureVQA benchmark and the visual procedural reasoning task, Table 9 shows five examples covering the domains introduced in Section 3.2. Each instance in our benchmark consists of a step-by-step procedural sequence, where the system must identify the correct next step given a query image. The default question is "What is the next step?", reflecting real-world use cases: when a user uploads an image showing a particular state, the QA system must retrieve the relevant procedure and predict the subsequent action.

A.2 Sensitive Analysis of Fusion Probability

To justify our choice of the fusion probability parameter, we conducted a sensitivity analysis using GPT-4o, Gemini-2.5-Flash, and Claude-3.5-Sonnet on 1,000 sampled instances under three settings: Low-Fusion (25%), Balanced (50%), and High-Fusion (75%). As shown in Table 8, BERTScore remains relatively stable across the three settings.

Model	Setting	BERTScore
GPT-4o	Low	0.301
	Balanced	0.305
	High	0.312
Gemini-2.5-Flash	Low	0.453
	Balanced	0.463
	High	0.477
Claude-3.5-Sonnet	Low	0.345
	Balanced	0.349
	High	0.351

Table 8: Sensitivity analysis of fusion probability across three LLMs on 1,000 sampled instances.

These results validate the fusion probability as an effective difficulty control parameter, and our default setting of 50% strikes a reasonable balance between sufficient concept integration and manageable complexity.

A.3 Implementation

For closed-source VLMs and VLMs without fine-tuning, we directly use the models through their official APIs with temperature set to 0 to ensure experimental reproducibility. Regarding the fine-tuned Qwen2.5-VL-7B² (Wang et al., 2024) model described in Section 6, we fine-tune (PEFT) it with LoRA (Hu et al., 2022), a parameter-efficient fine-tuning method (Chen et al., 2025b) for LLMs, using the LLaMA-Factory³ (Zheng et al., 2024) framework on single H800 GPU. The training configuration employs a batch size of 4 with 4-step gradient accumulation, a learning rate of 1e-4, and a maximum sequence length of 8192 tokens. This setup ensures memory efficiency while maintaining stable optimization dynamics during the adaptation process. The two benchmarks constructed in this work will be released upon acceptance of this paper. The code will be available at <https://github.com/NLP2CT/Chain-of-Procedure>.

A.4 Detailed Sub-Task Formulations

We provide formal definitions for each diagnostic sub-task.

Shuffled Procedure Verification (SIV). Given a procedure $\mathcal{P} = \{S_1, \dots, S_L\}$, the model receives a permuted version $\mathcal{P}' = \{S_{\pi(1)}, \dots, S_{\pi(L)}\}$ where π is a random permutation. The task is binary

²<https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct>

³<https://github.com/hiyouga/LLaMA-Factory>

classification: predict whether $\mathcal{P}' = \mathcal{P}$. This tests sensitivity to procedural order without requiring visual grounding.

Current Step Identification (CSI). Given a visual query v_t and the complete procedure $\mathcal{P} = \{S_1, \dots, S_L\}$, the model identifies which step $S_i \in \mathcal{P}$ corresponds to the current visual state (multi-class classification over L steps). This evaluates temporal grounding between vision and text.

Next-Step Prediction (NSP). Given visual query v_t showing the state after completing S_t and the procedure \mathcal{P} , the model predicts the next step S_{t+1} . Unlike CSI, which requires recognition, NSP demands forward temporal reasoning about future actions.

Disordered Procedure Adaptation (DPA). The model receives visual query v_t and shuffled procedure $\mathcal{P}' = \{S_{\pi(1)}, \dots, S_{\pi(L)}\}$, then must predict the correct next step S_{t+1} from the original sequence \mathcal{P} . This simulates real-world scenarios where provided procedures may be incomplete or misordered.

Cross-Procedure Matching (CPM). Given a visual query v_t depicting procedure \mathcal{P}_i and a candidate procedure description \mathcal{P}_j , the model determines if $\mathcal{P}_i = \mathcal{P}_j$ (binary classification). This tests cross-modal semantic alignment without requiring fine-grained temporal reasoning.

Each task uses identical image-procedures pairs from our benchmark but varies the input format and prediction target to isolate specific reasoning capabilities. Note that these decomposed sub-tasks simplify the full VP-QA problem by removing one or more of the three core challenges: cross-procedure retrieval, visual state grounding, and next-step reasoning.

A.5 Evaluation Metrics

The LLM-Score employs Claude-3.7 (Anthropic, 2025), Gemini-2.5 (Team, 2023), and GPT-4.1 (OpenAI, 2025) as expert judges following Chia et al. (2024). These LLMs independently assess the semantic equivalence between model outputs and reference next-step actions through a scoring prompt. Final LLM-Scores represent normalized averages across all three judges, mitigating individual model biases. The instructions of all LLM judge experts are shown in Table 10.

A.6 The Impact of Negative Manuals

We analyze Phase 1 matching accuracy under different scenarios in Figure 8, varying the num-

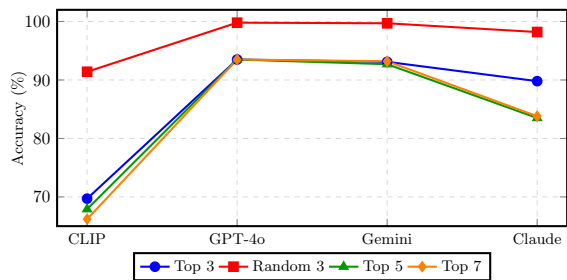


Figure 8: The results with different numbers of negative candidate procedures and sampling strategies.

ber of negative candidate procedures and sampling strategies across retrieval models and vLLMs (CLIP, GPT-4o, Gemini-2.5-Flash, and Claude-3.5-Sonnet). Top-k ($k=3,5,7$) refers to our construction in Section 3.2, where candidates include one correct procedure and $N-1$ visually similar negative procedures; Random-3 uses one correct procedure with two randomly sampled negatives. The results reveal two key findings: First, Random-3 achieves substantially higher accuracy than all Top-k settings, demonstrating the effectiveness of our hard negative sampling strategy. Second, increasing the number of negatives consistently degrades performance on CLIP and Claude-3.5-Sonnet, with slight drops even on GPT-4o and Gemini-2.5-Flash, highlighting the necessity and rationale of using large language models rather than pure matching models for Phase 1.

A.7 Instruction Template of VLMs

This section presents the complete collection of instructions used throughout our experimental analysis. As shown in Table 11, we provide the standardized prompt for our VLM baselines, where VLMs directly predict the next step. Table 12 outlines the operationalization of our five evaluation sub-tasks designed to assess VLM capabilities. Finally, Table 13 details the phase-specific instructions corresponding to different phases of our proposed CoP framework.

A.8 Discussion of Error Propagation

By comparing the two plots in Figure 7, we observe that for VLMs of different sizes, larger models consistently achieve better performance in phase 2, which subsequently leads to improved performance in phase 3. However, for the same VLM across different domains, the performance of phase 2 and phase 3 does not exhibit a positive correlation, indicating that the model possesses certain

robustness against error propagation throughout the pipeline. This robustness explains why fine-tuned models demonstrate substantial overall improvement, as capabilities are enhanced across all three phases. Nevertheless, examining the performance variations of the three phases across five domains reveals that our proposed method can better leverage the model’s inherent capabilities, ultimately achieving better performance on the final task compared to the baseline.

A.9 Synthesize Fine-tuned Data

We leverage Qwen3-VL-235B-A22B-Thinking (Team, 2025) for synthesizing procedure-aligned reasoning chains, following the data construction protocol outlined in Section 3. The synthesis process involves presenting the model with candidate procedures alongside query images, then explicitly specifying: (1) the appropriate procedure selection, (2) whether finer-grained decomposition is necessary, (3) the step corresponding to the query image, and (4) the subsequent action to execute. Based on these structured inputs, the model generates complete thinking trajectories. A subsequent LLM-based verification stage examines both step-wise answer correctness and adherence to the prescribed information sequence. Failed samples undergo up to three regeneration attempts before human annotators take over to craft the reasoning process manually, ensuring consistent quality across our fine-tuning and benchmark datasets.

Cars & Other Vehicles	Computers and Electronics	Hobbies and Crafts
<p>Procedure 1: Step 1: Pour coolant into the (Other Steps) Step 4: Top off your coolant ... (Other extracted procedures)</p>	<p>Procedure 1: Step 1: Understand shortcut (Other Steps) Step 10: Back up your files ... (Other extracted procedures)</p>	<p>Procedure 1: Step 1: Purchase frisket film (Other Steps) Step 6: Remove frisket from ... (Other extracted procedures)</p>
<p>User Query Image:</p> 	<p>User Query Image:</p> 	<p>User Query Image:</p> 
<p>Golden Next Step: Start your vehicle's engine ...</p>	<p>Golden Next Step: Type command prompt. A ...</p>	<p>Golden Next Step: Place unpeeled frisket over ...</p>
Sports and Fitness	Work World	
<p>Procedure 1: Step 1: Pick skis that are (Other Steps) Step 6: Protect yourself. ... (Other extracted procedures)</p>	<p>Procedure 1: Step 1: Let them lead in the (Other Steps) Step 3: Listen for sincerity. ... (Other extracted procedures)</p>	
<p>User Query Image:</p> 	<p>User Query Image:</p> 	
<p>Golden Next Step: Grab a set of poles. Ski poles ...</p>	<p>Golden Next Step: Don't brush off their guilt. ...</p>	

Table 9: The examples of five domain data in our benchmark.

Instruction of LLM-Score
<pre>## Correct Next Step: {LABEL}</pre>
<pre>## Predict Next Step: {PREDICT}</pre>
<pre>##Instructions: - The "Correct Next Step" is the ground truth of the next step. - The "Predict Next Step" is the model output of the next step. - Score the predicted next step (0-10) based on how concisely and accurately it covers all essential operations from the correct next step, without omissions or extra information. - The score should be between 0 to 10. - Please output the score directly without any introduction or explanation.</pre>

Table 10: The instruction for the LLM judge expert to score the experimental outputs.

Instruction of VLM baselines
<pre>## Instruction Manuals: {STEPS}</pre>
<pre>##Instructions: - Several instructions are provided, each containing steps in sequential order (with no shuffling). - Some steps may combine multiple actions into a single step. - Based on the given image, identify the corresponding instruction manual and determine the next step. - Please think step-by-step and make sure the last line of the output should be the following format: step_X: [content of the next step]</pre>

Table 11: The instruction for VLM baselines.

Instruction of Predict Current Step
<p><Image></Image></p> <p>## Instructions: {STEPS}</p> <ul style="list-style-type: none"> - Please identify which step the image corresponds to, based on the given instruction. - You should only output text in the form of 'step_X', without any other explanations or descriptions.
Instruction of Predict Next Step
<p><Image></Image></p> <p>## Instructions: {STEPS}</p> <ul style="list-style-type: none"> - Please identify the next step after the step shown in the given image, based on the given instruction. - You should only output text in the form of 'step_X', without any other explanations or descriptions.
Instruction of Randomly Shuffled Detection
<p>## Instructions: {STEPS}</p> <ul style="list-style-type: none"> - Please determine whether the operational procedure in the instructions has been shuffled. - You should only output 'True' or 'False', without any other introduction or explanation.
Instruction of Predict Next Step in Shuffled Instructions
<p><Image></Image></p> <p>## Instructions: {STEPS}</p> <ul style="list-style-type: none"> - Given the shuffled instructions, please identify the next step after the step shown in the given image. - You should only output text in the form of "step_X", without any other introduction or explanation.
Instruction of Cross-Procedure Matching
<p><Image></Image></p> <p>## Instructions: {STEPS}</p> <ul style="list-style-type: none"> - Please determine whether the content of the image corresponds to any step described in the given instruction. - You should only output 'True' if the image corresponds to the instruction, or 'False', without any other introduction or explanation.

Table 12: The instructions for the five evaluation sub-tasks.

Instruction of Context-Aware Procedure Retrieval

<Image></Image>

Task: Identify which instruction manual the image belongs to.

##Instructions:
{STEPS}

- Several instructions are provided (start with ###), each containing steps in sequential order (with no shuffling).
- Based on the given image, identify the corresponding instruction among the provided candidate instructions.
- Ensure that your response includes only the instruction ID.
- Output your result in the following format: [Instruction id]
- Do not include any explanations, reasoning, or additional information in your response.

Instruction of Step-Wise Procedural Decomposition

Task: Identify and split the combined steps in the instruction manual.

Instruction:
{STEPS}

- The instruction contains steps in strict sequential order (no shuffling or reordering).
- Some steps may combine two distinct actions into a single step.
- Split if the actions are clearly separate (unrelated tools, different targets, or independent operations).
- Do not split if the actions are part of a continuous process (same tool/object, sequential dependencies, or a single logical operation).
- Output the modified instruction with only the necessary splits applied, keeping all other steps and the step description unchanged.
- Output your result in the following format:
step_1: [content of step_1]
step_2: [content of step_2]
...
step_X: [content of step_X]
- Do not include any explanations, reasoning, or additional information in your response.

Instruction of Next Step Prediction

<Image></Image>

Task: Identify which step in the instruction the input image belongs to.

Instruction:
{STEPS}

- The instruction contains steps in sequential order (with no shuffling).
- Based on the given image, identify which step this image belongs to.
- Ensure that your response includes only the step of the image.
- Output your result in the following format:
step_X: [content of the current step]
- Do not include any explanations, reasoning, or additional information in your response.

Table 13: The instructions for each phase in our CoP framework.