Confidence-guided Refinement Reasoning for Zero-shot Question Answering

Youwon Jang[1] **Woo Suk Choi**[1] **Minjoon Jung**[1]
Minsu Lee[2*] **Byoung-Tak Zhang**[1*]
[1]Seoul National University [2]Sungshin Women's University
{ywjang, wschoi, mjjung}@bi.snu.ac.kr, mslee@sungshin.ac.kr, btzhang@bi.snu.ac.kr

Abstract

We propose Confidence-guided Refinement Reasoning (C2R), a novel training-free framework applicable to question-answering (QA) tasks across text, image, and video domains. C2R strategically constructs and refines sub-questions and their answers (sub-QAs), deriving a better confidence score for the target answer. C2R first curates a subset of sub-QAs to explore diverse reasoning paths, then compares the confidence scores of the resulting answer candidates to select the most reliable final answer. Since C2R relies solely on confidence scores derived from the model itself, it can be seamlessly integrated with various existing QA models, demonstrating consistent performance improvements across diverse models and benchmarks. Furthermore, we provide essential yet underexplored insights into how leveraging sub-QAs affects model behavior, specifically analyzing the impact of both the quantity and quality of sub-QAs on achieving robust and reliable reasoning.

1 Introduction

Question-answering (QA) is a fundamental reasoning task that demands a deep understanding of given content and has been explored across various domains, including texts (Hendrycks et al., 2020; Wang et al., 2024b), images (Malinowski and Fritz, 2014; Zhu et al., 2016), and videos (Xiao et al., 2021; Engin and Avrithis, 2023). Traditional approaches treat QA tasks as a single-step reasoning problem, directly generating answers without intermediate analysis (Figure 1A). However, answers to complex questions may not be directly inferable and often necessitate multi-step reasoning. As illustrated in Figure 1 (B), multi-step reasoning methods (Uehara et al., 2022; Wang et al., 2022b; Khan et al., 2023; Yao et al., 2023; Besta et al., 2024) have been proposed that decompose the main question into several sub-question-answer pairs (sub-QAs) and subsequently utilize them to derive the final answer.

However, we argue that sub-QAs do not always enhance QA reasoning; their indiscriminate use can adversely affect and distract the model's inference procedure. Our investigation reveals that when sub-QAs are irrelevant to the main question or paired with inaccurate answers, they introduce noise into the reasoning process, ultimately degrading answer quality. Despite these risks, prior works (Uehara et al., 2022; Khan et al., 2023; You et al., 2023; Liao et al., 2024) often incorporate sub-QAs without sufficient verification or refinement, as their relevance to the main question is neither assessed nor guaranteed.

To this end, we propose Confidence-guided Refinement Reasoning (C2R), a framework comprising three core components: Generator, Refiner, and Answer Selector. The Generator decomposes a main question into multiple sub-questions and generates corresponding sub-answers. The Refiner then selectively curates a subset of these sub-QAs to explore diverse reasoning paths Each reasoning path yields an answer candidate accompanied by a confidence score, obtained directly from the model. Finally, the Answer Selector determines the most appropriate answer by comparing these confidence scores, including the score of the single-step answer that does not rely on any sub-QAs.

As C2R only relies on self-assessed confidence scores, it is inherently training-free and readily adaptable to various QA models. By extensive experiments across various QA tasks, C2R demonstrates consistent improvements on a range of state-of-the-art models (Yang et al., 2024; Team et al., 2025; Li et al., 2024; Hurst et al., 2024). To provide further insights, we examine the impact of visual information on C2R performance and determine the optimal number of sub-QAs for achieving robust performance. We also reveal the *confidence infla-*

---

[*] Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6933–6950
November 4-9, 2025 ©2025 Association for Computational Linguistics

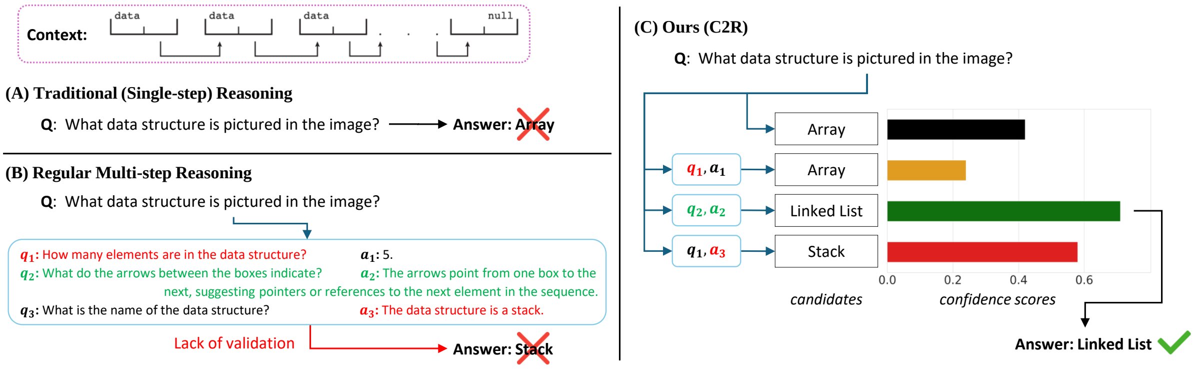

Context:
data
data
data
null
(A) Traditional (Single-step) Reasoning
Q: What data structure is pictured in the image?
Answer: Array
(B) Regular Multi-step Reasoning
Q: What data structure is pictured in the image?
q_1: How many elements are in the data structure?
a_1: 5.
q_2: What do the arrows between the boxes indicate?
a_2: The arrows point from one box to the next, suggesting pointers or references to the next element in the sequence.
q_3: What is the name of the data structure?
a_3: The data structure is a stack.
Lack of validation
Answer: Stack
(C) Ours (C2R)
Q: What data structure is pictured in the image?
q_1, a_1
q_2, a_2
q_1, a_3
Array
Array
Linked List
Stack
0.0
0.2
0.4
0.6
candidates
confidence scores
Answer: Linked List ✓

Figure 1: **Comparison of existing QA methods. (A):** Traditional single-step reasoning. **(B):** Multi-step reasoning via question decomposition, typically employed for complex problems, but often overlooking inaccuracies in generated sub-QAs (highlighted in red text). **(C):** C2R minimizes the risk of relying on *unverified* sub-QAs by selecting the most confident answer among multiple candidates based on their confidence scores.

tion problem, where utilizing sub-QAs increases the confidence scores for answer candidates even when they are incorrect, underscoring the need for careful usage of sub-QAs. To mitigate this, we set confidence thresholds to reliably select the final answer and analyze their impact on model behavior and performance.

To summarize, our contributions are threefold:

- We propose the Confidence-guided Refinement Reasoning (C2R) framework, which strategically utilizes sub-QAs and guides the model in selecting final answers based on self-assessed confidence scores.
- We demonstrate that C2R consistently achieves improvements across five models and five benchmarks in zero-shot settings.
- We provide insights into often-overlooked issues concerning the impact of utilizing irrelevant or incorrect sub-QAs within multi-step reasoning QA methodologies.

2 Related works

2.1 Question answering

Question-answering (QA) tasks span various modalities, including text, image, and video. Text-only QA evaluates a model's reasoning ability and factual knowledge across diverse domains. A range of benchmarks (Hendrycks et al., 2020; Wang et al., 2024b; Geva et al., 2021) evaluate a model's factual knowledge, complex reasoning capabilities, and ability to perform multi-hop inference. ImageQA (Malinowski and Fritz, 2014; Antol et al., 2015), also known as Visual Question Answering, involves providing an accurate natural language answer to a question based on a given image. Recently, to assess more profound knowledge and reasoning ability, MMMU (Yue et al., 2024) includes single- and multi-image questions in both multiple-choice and open-ended formats across diverse academic domains. VideoQA (Tapaswi et al., 2016; Xiao et al., 2021) extends the complexity by incorporating temporal dynamics. Datasets like EgoSchema (Mangalam et al., 2023) present long-form videoQA challenges, requiring models to comprehend and reason over extended video sequences. To address QA tasks across these diverse domains, powerful foundation models (Maaz et al., 2023; Li et al., 2024; Yang et al., 2024; Team et al., 2025) have been proposed, leveraging techniques like large-scale pretraining. In this work, we propose a general approach to improving QA performance across diverse domains by building upon unified models capable of handling text and vision modalities.

2.2 Multi-step reasoning for QA

To tackle questions requiring complex reasoning, many multi-step approaches have recently been proposed. Modular approaches (Gupta and Kembhavi, 2023; Surís et al., 2023; Choudhury et al., 2023) decompose complex instructions into a sequence of sub-programs, executing them through sub-modules. Chain-of-Thought (Wei et al., 2022) and CoT-SC (Wang et al., 2022a) are prompting techniques that encourage models to generate intermediate reasoning steps before producing a final answer, significantly improving the capabilities of large language models (LLMs). However, these methods lack an automated mechanism for error

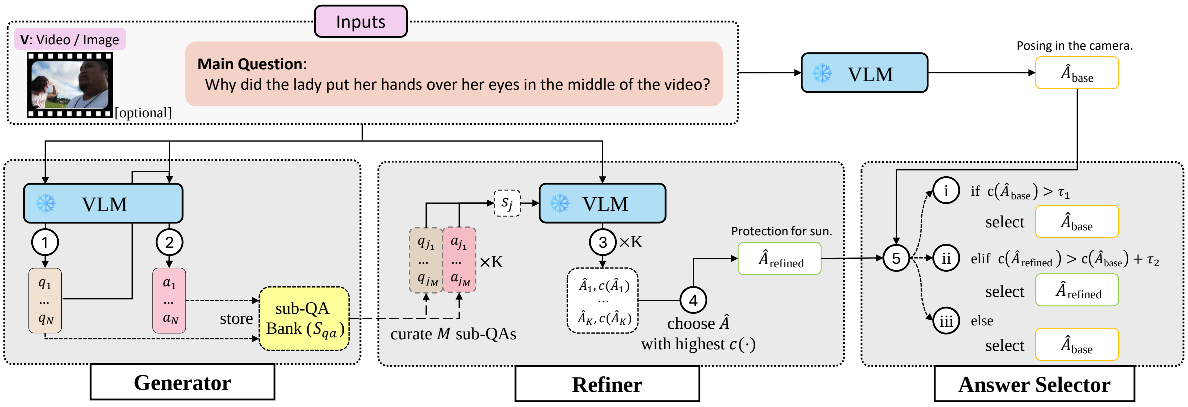

Inputs
V: Video / Image
[optional]
Main Question:
Why did the lady put her hands over her eyes in the middle of the video?
VLM
Posing in the camera.
$\hat{A}_{\text{base}}$
VLM
①
②
q_1 … q_N
a_1 … a_N
store
sub-QA Bank (S_{qa})
s_j
VLM
q_{j_1} … q_{j_M}
a_{j_1} … a_{j_M}
×K
③×K
$\hat{A}_1, c(\hat{A}_1)$ … $\hat{A}_K, c(\hat{A}_K)$
curate M sub-QAs
④ choose $\hat{A}$ with highest $c(\cdot)$
Protection for sun.
$\hat{A}_{\text{refined}}$
⑤
(i) if $c(\hat{A}_{\text{base}}) > \tau_1$
select $\hat{A}_{\text{base}}$
(ii) elif $c(\hat{A}_{\text{refined}}) > c(\hat{A}_{\text{base}}) + \tau_2$
select $\hat{A}_{\text{refined}}$
(iii) else
select $\hat{A}_{\text{base}}$
Generator
Refiner
Answer Selector

Figure 2: **An overview of Confidence-guided Refinement Reasoning (C2R)**. Given content V and main question Q, Generator constructs N sub-QAs. Refiner then curates K subsets, each consisting of M sub-QAs, to generate answer candidates with their corresponding confidence scores. Finally, Answer Selector chooses one as the final answer from $\hat{A}_{\text{base}}$ and $\hat{A}_{\text{refined}}$ using the confidence thresholds τ_1 and τ_2. Vision-Language Model (VLM) is freezed.

correction and are thus vulnerable to cascading failures caused by errors in early steps.

To overcome this limitation, methods like ToT (Yao et al., 2023), GoT (Besta et al., 2024), and DeAR (Xue et al., 2024) have been proposed, where intermediate steps are organized in tree or graph structures to explore optimal reasoning paths. They also include a verification step where the sub-answers are checked by prompting the LLM directly. However, verifying each intermediate step using an LLM can be suboptimal (Wang et al., 2023) and requires additional computation.

In this paper, we propose a confidence-based approach that minimizes the impact of inaccurate sub-QAs without directly evaluating each one, thereby improving both efficiency and robustness. We also analyze the effectiveness of directly assessing sub-QAs with an LLM in Appendix A.6.

3 C2R Framework

In this section, we begin by explaining how the model derives confidence scores from its answers (§3.1) and then outline the basic inference process in QA tasks (§3.2). Subsequently, we introduce the three core components of C2R: Generator, Refiner, and Answer Selector (§3.3–§3.5). Figure 2 illustrate our C2R framework. In the following explanation, V can be an image or a video.

3.1 Confidence of the answer

When the model f generates an answer $\hat{A}$ given context V and a main question Q, we measure its confidence as $c(\hat{A})$, where $c(\cdot)$ is the confidence scoring function. First, for the answer sequence $\hat{A} = \{y_1, \ldots, y_L\}$, we apply the softmax function to the logits and denote the probability of the selected token y_i as p_i. This yields a sequence of probabilities $[p_1, \ldots, p_L]$ corresponding to the generated tokens. Then, we compute the confidence score as $c(\hat{A}) := \min_{i=1}^{L} p_i$. We adopt the minimum probability across the answer tokens as a conservative strategy (Geng et al., 2023; Kim et al., 2024), under the assumption that the least confident token may indicate a potential failure point. Additional confidence metrics are discussed in Appendix A.7.

3.2 Base inference process

Given context V and a main question Q, the corresponding answer $\hat{A}_{\text{base}}$ in an open-ended QA task is obtained from the model f as:

$$\hat{A}_{\text{base}} = f(V, Q). \tag{1}$$

For a multiple-choice QA task, the model f selects the answer from a set of answer options $\{A_{\text{option}}\}$ as:

$$\hat{A}_{\text{base}} = f(V, Q, \{A_{\text{option}}\}). \tag{2}$$

We refer to $\hat{A}_{\text{base}}$ as the *base answer*. If the confidence of the base answer (i.e., $c(\hat{A}_{\text{base}})$) is sufficiently high, the subsequent steps are skipped. We explain this in §3.5.

3.3 Generator

Given V and Q, the Generator constructs subquestions and their corresponding answers. Formally, the model f generates N sub-QAs as:

$$q_i \sim f(V, Q), \; a_i = f(V, q_i), \quad 1 \leq i \leq N. \tag{3}$$

Algorithm 1 Confidence-guided Refinement Reasoning

Input: V (optional): video or image(s), Q: main question (related to V)
Parameter: N, M, K, τ_1, τ_2
Output: $\hat{A}$: answer to main question Q

```
$\hat{A}_{\text{base}}, c(\hat{A}_{\text{base}}) = f(V, Q)$
if $c(\hat{A}_{\text{base}}) \geq \tau_1$ then
    return $\hat{A}_{\text{base}}$
end if
$S_{qa} \leftarrow \{\}$
for $i \leftarrow 1$ to $N$ do
    $q_i = f(V, Q),\ a_i = f(V, q_i)$
    $S_{qa} = S_{qa} \cup \{(q_i, a_i)\}$
end for
for $j \leftarrow 1$ to $K$ do
    curate $s_j := (q_1^j, a_1^j), ..., (q_M^j, a_M^j)$ from $S_{qa}$
    $\hat{A}_j, c(\hat{A}_j) = f(V, Q | s_j)$
end for
idx $= \underset{j}{\operatorname{argmax}}\, c(\hat{A}_j)$
$\hat{A}_{\text{refined}} = \hat{A}_{\text{idx}}$
if $c(\hat{A}_{\text{refined}}) \geq c(\hat{A}_{\text{base}}) + \tau_2$ then
    return $\hat{A}_{\text{refined}}$
else
    return $\hat{A}_{\text{base}}$
end if
```

We refer to this set of generated sub-QAs as the sub-QA bank, denoted S_{qa}:

$$S_{qa} = \{(q_1, a_1), (q_2, a_2), \ldots, (q_N, a_N)\}. \quad (4)$$

3.4 Refiner

We introduce the Refiner to avoid irrelevant sub-QAs leading the model to an incorrect conclusion. First, the Refiner curates K sub-QA subsets from the sub-QA bank S_{qa} while satisfying the following conditions: 1) The indices of sub-QA pairs within each subset should be unique. 2) Any two curated subsets may share some sub-QA pairs, but they must not be identical in composition.

Formally, the j-th curated sub-QA subset is denoted as $\{(q_1^j, a_1^j), \ldots, (q_M^j, a_M^j)\}$, where $1 \leq j \leq K$. We then concatenate the selected sub-QAs to form the curated context s_j:

$$s_j = [(q_1^j, a_1^j) \mid\mid \ldots \mid\mid (q_M^j, a_M^j)], \quad (5)$$

where $\forall i, k \in \{1, \ldots, M\}$, $(q_i^j, a_i^j) \neq (q_k^j, a_k^j)$ for $i \neq k$, and $||$ denotes the concatenation operator. We then feed the s_j as additional input to the model to generate a candidate answer $\hat{A}_j$:

$$\hat{A}_j = f(V, Q \mid s_j) \quad (6)$$

As a result, we obtain K answer candidates $\{\hat{A}_j\}$, where $1 \leq j \leq K$.[1] Finally, the refined answer, $\hat{A}_{\text{refined}}$, is selected as the candidate from $\{\hat{A}_j\}_{j=1}^{K}$ that has the highest confidence score.

3.5 Answer Selector

Although $\hat{A}_{\text{refined}}$ is obtained using sub-QAs that are likely relevant to V and Q, it remains uncertain whether it is the correct answer. To improve the reliability of the inferred answer, we design the Answer Selector to guide the model toward more accurate conclusions. Our Answer Selector follows two principles:

Principle 1. For V and Q that are relatively easy to understand and answer, overcomplicating the task can distract the model and hinder it from identifying the correct answer (Shi et al., 2023). Therefore, if the base answer $\hat{A}_{\text{base}}$ exhibits a sufficiently high confidence score (i.e., $c(\hat{A}_{\text{base}}) \geq \tau_1$), we choose it as the final answer, and subsequent refinement steps are skipped. We use a confidence threshold τ_1 for this purpose.

Principle 2. If $c(\hat{A}_{\text{base}}) < \tau_1$, the Answer Selector considers both $\hat{A}_{\text{base}}$ and $\hat{A}_{\text{refined}}$. However, a naive comparison of $c(\hat{A}_{\text{base}})$ and $c(\hat{A}_{\text{refined}})$ yields only marginal improvement. We empirically observe that using sub-QAs tends to inflate confidence scores compared to $c(\hat{A}_{\text{base}})$, even when incorrect sub-QAs are provided, potentially leading to erroneous selections. To address this, we introduce an additional confidence threshold τ_2. Specifically, $\hat{A}_{\text{refined}}$ is selected if its confidence score $c(\hat{A}_{\text{refined}})$ satisfies $c(\hat{A}_{\text{refined}}) \geq c(\hat{A}_{\text{base}}) + \tau_2$; otherwise, $\hat{A}_{\text{base}}$ is chosen.

These principles guide the selection of the more reliable answer between $\hat{A}_{\text{base}}$ and $\hat{A}_{\text{refined}}$. The full procedure is listed in Algorithm 1, and we analyze the effect of confidence thresholds in §5.3.

4 Experiments

In this section, we provide experimental results to demonstrate the effectiveness of our C2R framework. In §4.1, we outline the experimental setup,

---

[1] We do not necessarily explore all K reasoning paths for every question, as early stopping is possible when sufficient confidence is achieved. This is further explained in Appendix A.5.

Backbone	Size	Reasoning Method	N	M	K	Text-only QA			ImageQA	VideoQA
						MMLU	**MMLU-Pro**	**StrategyQA**	**MMMU**	**EgoSchema**
Qwen2.5	7B	Baseline	-	-	-	67.1	38.6	62.6	50.2	68.0
		SingleSubQA	1	1	1	67.5	42.9	62.6	47.6	69.2
		EverySubQA	5	5	1	67.7	42.6	65.3	47.6	69.0
		LLMVerified(sub-A)	5	2	1	67.3	43.2	64.5	49.4	67.0
		C2R(Ours)	5	2	4	**69.2**	**44.3**	**65.7**	**51.9**	**71.5**
		△				**+2.1**	**+5.7**	**+3.1**	**+1.7**	**+3.5**
LLaVA-Onevision	7B	Baseline	-	-	-	66.2	35.7	64.5	45.1	43.0
		SingleSubQA	1	1	1	64.3	36.6	67.8	44.1	40.2
		EverySubQA	5	5	1	63.4	37.0	67.6	46.1	39.8
		LLMVerified(sub-A)	5	2	1	65.1	36.9	68.2	47.2	45.2
		C2R(Ours)	5	2	4	**67.3**	**38.4**	**68.3**	**47.4**	**46.0**
		△				**+1.1**	**+2.7**	**+3.8**	**+2.3**	**+3.0**
Gemma 3	4B	Baseline	-	-	-	58.9	27.3	45.9	40.1	50.5
		C2R(Ours)	5	2	4	**60.0**	**30.5**	**50.1**	**42.1**	**53.2**
		△				**+1.1**	**+3.2**	**+4.2**	**+2.0**	**+2.7**
Qwen2	2B	Baseline	-	-	-	50.0	23.6	**55.0**	41.4	56.8
		C2R(Ours)	5	2	4	**50.8**	**24.7**	**55.0**	**42.4**	**61.0**
		△				**+0.8**	**+1.1**	+0.0	**+1.0**	**+4.2**
GPT-4o	-	Baseline	-	-	-	85.0	-	-	56.1	75.0
		C2R(Ours)	5	2	4	**86.2**	-	-	**58.3**	**78.2**
		△				**+1.2**	-	-	**+2.2**	**+3.2**

Table 1: **Zero-shot evaluation results of C2R.** We evaluate baseline models and our framework across various domains and benchmarks for comparison. N, M, and K denote the number of generated sub-QAs, the number of curated sub-QAs per reasoning path, and the total number of reasoning paths, respectively. The difference (△) denotes the score improvement of C2R over the vanilla models. **Bold text** indicates the highest score and underlined text the second-highest. Colors show performance relative to baseline: blue (higher), red (lower).

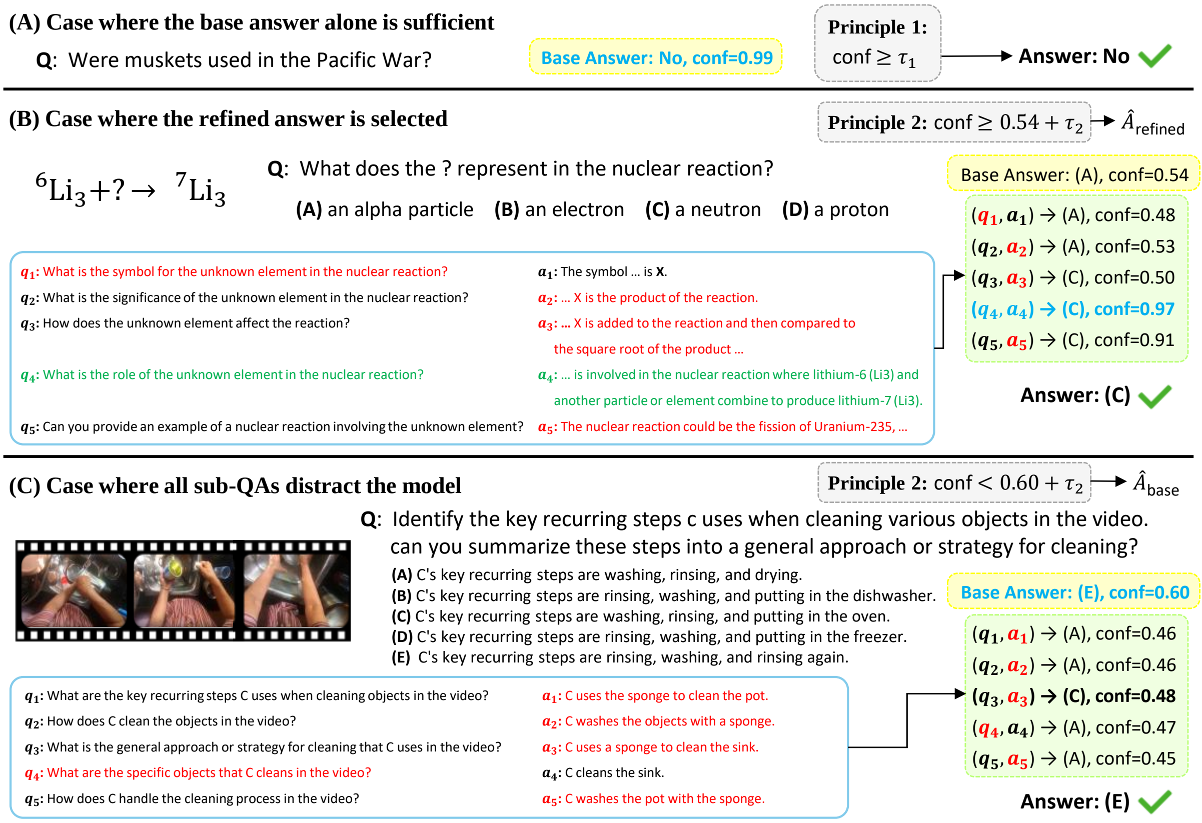
including datasets and baselines. Subsequently, we compare our framework with various multi-step reasoning methods across various QA models (§4.2), and visualize qualitative results (§4.3). The prompt designs used for sub-QA generation and model inference can be found in Appendix E.

4.1 Experimental setup

For all experiments, we fix the number of generated sub-QAs to $N = 5$ and consistently use greedy search for decoding. All experiments are conducted on a single A6000 GPU, except for closed-source models. For closed-source models, we evaluate one benchmark per domain due to cost constraints. Further details are provided in Appendix D.

Datasets. We evaluate our framework on five diverse benchmarks for Text-only, Image, and Video QA, covering various domains for a comprehensive evaluation. For Text-only QA, we use the following three datasets: MMLU (Hendrycks et al., 2020), MMLU-Pro (Wang et al., 2024b), and StrategyQA (Geva et al., 2021), to cover different reasoning demands. MMLU consists of various academic and professional tasks requiring broad world knowledge, and MMLU-Pro extends the number of answer candidates from four to ten, making the model's decision more challenging. StrategyQA assesses a model's ability to perform implicit, multi-hop reasoning by requiring strategic inference over unstated facts. For ImageQA, we use MMMU (Yue et al., 2024), which evaluates deliberate reasoning across diverse image types with college-grade questions spanning six disciplines. Its breadth and depth make it a strong benchmark for assessing domain-specific visual understanding. For VideoQA, we employ EgoSchema (Mangalam et al., 2023), which focuses on long-form egocentric videos depicting naturalistic human activities. This dataset assesses comprehension of extended temporal contexts and complex behaviors, offering a rigorous testbed for evaluating advanced video reasoning capabilities.

Backbones. We apply our framework to several vision-language models capable of performing QA tasks: Qwen2.5-VL-7B (Yang et al., 2024), Gemma-3-4B (Team et al., 2025), and LLaVA-Onevision-7B (Li et al., 2024). To broaden the evaluation spectrum, we also include a relatively small open-source model, Qwen2-VL-2B (Wang et al., 2024a), and a state-of-the-art very large closed-source model, GPT-4o (Hurst et al., 2024). Please refer to Appendix C for further details.


(A) Case where the base answer alone is sufficient

Q: Were muskets used in the Pacific War?

Base Answer: No, conf=0.99

Principle 1: conf $\geq \tau_1$ → **Answer: No** ✓

(B) Case where the refined answer is selected

Principle 2: conf $\geq 0.54 + \tau_2$ → $\hat{A}_{\text{refined}}$

$^{6}\text{Li}_3 + ? \rightarrow\ ^{7}\text{Li}_3$

Q: What does the ? represent in the nuclear reaction?

(A) an alpha particle **(B)** an electron **(C)** a neutron **(D)** a proton

q_1: What is the symbol for the unknown element in the nuclear reaction? a_1: The symbol … is **X**.

q_2: What is the significance of the unknown element in the nuclear reaction? a_2: … X is the product of the reaction.

q_3: How does the unknown element affect the reaction? a_3: … X is added to the reaction and then compared to the square root of the product …

q_4: What is the role of the unknown element in the nuclear reaction? a_4: … is involved in the nuclear reaction where lithium-6 (Li3) and another particle or element combine to produce lithium-7 (Li3).

q_5: Can you provide an example of a nuclear reaction involving the unknown element? a_5: The nuclear reaction could be the fission of Uranium-235, …

Base Answer: (A), conf=0.54

(q_1, a_1) → (A), conf=0.48

(q_2, a_2) → (A), conf=0.53

(q_3, a_3) → (C), conf=0.50

(q_4, a_4) → (C), conf=0.97

(q_5, a_5) → (C), conf=0.91

Answer: (C) ✓

(C) Case where all sub-QAs distract the model

Principle 2: conf $< 0.60 + \tau_2$ → $\hat{A}_{\text{base}}$

Q: Identify the key recurring steps c uses when cleaning various objects in the video. can you summarize these steps into a general approach or strategy for cleaning?

(A) C's key recurring steps are washing, rinsing, and drying.
(B) C's key recurring steps are rinsing, washing, and putting in the dishwasher.
(C) C's key recurring steps are washing, rinsing, and putting in the oven.
(D) C's key recurring steps are rinsing, washing, and putting in the freezer.
(E) C's key recurring steps are rinsing, washing, and rinsing again.

q_1: What are the key recurring steps C uses when cleaning objects in the video? a_1: C uses the sponge to clean the pot.

q_2: How does C clean the objects in the video? a_2: C washes the objects with a sponge.

q_3: What is the general approach or strategy for cleaning that C uses in the video? a_3: C uses a sponge to clean the sink.

q_4: What are the specific objects that C cleans in the video? a_4: C cleans the sink.

q_5: How does C handle the cleaning process in the video? a_5: C washes the pot with the sponge.

Base Answer: (E), conf=0.60

(q_1, a_1) → (A), conf=0.46

(q_2, a_2) → (A), conf=0.46

(q_3, a_3) → (C), conf=0.48

(q_4, a_4) → (A), conf=0.47

(q_5, a_5) → (A), conf=0.45

Answer: (E) ✓

Figure 3: **Qualitative Results.** For simplicity, we visualize the case where $M = 1$. **(A)** shows a case where the base answer has a sufficiently high confidence, leading to its selection without generating refined answers. In **(B)**, the fourth sub-QA provides helpful information, enabling Refiner to arrive at the correct answer. With high confidence, Answer Selector then chooses the refined answer. In **(C)**, the refined answer is incorrect due to distraction from sub-QAs. Its low confidence leads to rejection by Answer Selector, which chooses the correct base answer.

Reasoning Methods. To further analyze the effectiveness of our framework, we introduce three additional reasoning methods. SINGLESUBQA is a baseline in which the answer is derived using only a single sub-QA, which is always accepted without further selection (Khan et al., 2023). EVERYSUBQA uses all N generated sub-QAs to answer the main question. LLMVERIFIED introduces a sub-QA verification step instead of exploring multiple reasoning paths (i.e., $K = 1$). It selects the M most helpful sub-QAs and generates a single refined answer using only these selected sub-QAs.

4.2 Zero-shot QA results

Although SINGLESUBQA and EVERYSUBQA often outperform the baseline, their improvements are unstable, leading to a substantial performance drop. For instance, SINGLESUBQA improves LLaVA-Onevision on Text-only QA datasets, but decreases its performance on both MMMU and EgoSchema. Furthermore, despite leveraging a larger number of sub-QAs, EVERYSUBQA frequently fails to yield any improvement and even underperforms than SINGLESUBQA on EgoSchema across different backbones. These results highlight that naively incorporating sub-QAs can negatively impact the model's performance. Moreover, simply increasing the number of intermediate reasoning steps is ineffective without proper verification.

We also compare C2R with LLMVERIFIED that sub-answers are evaluated by the underlying LLM (or VLM). LLMVERIFIED can show performance gains in some cases (e.g., LLaVA-Onevision's improvements on MMMU and EgoSchema), they can also lead to degraded performance on other benchmarks (e.g., MMLU). This indicates that such evaluation strategies may struggle to reliably identify truly helpful sub-QAs. In contrast, our approach increases the chances of accurate reasoning by aggregating signals from multiple reasoning paths—even when some paths include less reliable sub-answers. As a result, our method yields more consistent and robust performance improvements across diverse tasks. Related experiments on identifying the most helpful sub-questions or evaluating the overall quality of the sub-QAs set are provided in Appendix A.6.

Overall, our framework consistently improves performance over the baselines across different backbones and tasks, demonstrating its effectiveness. Additionally, we confirm that C2R performs well even with smaller backbone models: Gemma 3 and Qwen2, as well as with the closed-sourced model: GPT-4o, demonstrating its flexibility.

4.3 Qualitative results

Figure 3 illustrates the behavior of our C2R framework, across three distinct scenarios, showcasing its adaptability and robustness. In case (A), the base answer achieves a sufficiently high confidence score to be selected without invoking the Refiner. This demonstrates that C2R can efficiently recognize when refinement is unnecessary, thereby preserving computational resources while maintaining accuracy. Meanwhile, case (B) exemplifies a situation where the base answer's confidence is initially insufficient. Here, the fourth sub-QA helps the Refiner find the correct answer, which the Answer Selector chooses based on high confidence. This highlights C2R's ability to extract valuable signals from sub-QAs even when some are uninformative. However, there might be cases like (C) where the generated sub-QAs do not effectively support refinement. In this instance, the model falls back to the base answer, as it retains a higher confidence score than the refined options. This case illustrates the safety mechanism of the C2R, where unreliable refinement paths are effectively discarded. Overall, these cases highlight the adaptability of C2R, which can selectively refine when needed and fall back to the base answer when appropriate, ensuring both robustness and efficiency. For a detailed analysis of representative failure cases, please refer to Appendix F.

5 Analysis

In this section, we analyze our method on various axes. First, we investigate the optimal number of sub-QAs (M) used to generate each answer candidate. We then analyze the effectiveness of our framework on multimodal tasks by conducting a blind test that excludes access to visual information. Finally, we provide an analysis of the confidence score and its threshold. We also provide other aspects, such as computational cost, in Appendix A.

M	MMLU	MMMU	EgoSchema
-	67.1	50.2	68.0
1	+1.3	+0.9	+1.0
2	+2.1	**+1.7**	**+3.5**
3	**+2.2**	+1.1	+3.0
4	+2.1	+1.4	+1.5

Table 2: Performance gain with varying numbers of curated sub-QAs (M) per answer candidate. Results are based on Qwen2.5-VL with $N = 5$.

Model		MMMU		EgoSchema	
		Blind	**Std.**	**Blind**	**Std.**
Qwen2.5-VL-7b	Baseline	38.9	50.2	27.3	68.0
	C2R(Ours)	41.0	51.9	30.3	71.5
	Δ	**+2.1**	+1.7	+3.0	**+3.5**
LLaVA-Onevision-7b	Baseline	43.2	45.1	35.5	42.0
	C2R(Ours)	43.4	47.4	38.0	45.8
	Δ	+0.2	**+2.3**	+2.5	**+3.8**
Gemma-3-4b	Baseline	34.0	40.1	23.3	50.5
	C2R(Ours)	35.9	42.1	24.5	53.2
	Δ	+1.9	**+2.0**	+1.2	**+2.7**

Table 3: To validate the impact of visual inputs, we evaluate under a blind setting, where QA is performed without access to visual content. 'Std.' refers to the standard setting. C2R achieves greater improvements, confirming its effective use of visual information.

5.1 The number of curated sub-QAs

Table 2 shows the importance of an appropriate number of sub-QA pairs (i.e., M) when curating. Although all cases show improvements, curating with too small ($M = 1$) or too many ($M = 4$) sub-QAs results in suboptimal performance gains. We choose $M = 2$ as it demonstrates the best performance. We provide additional analysis on varying the number of refined reasoning paths (i.e., K) in Appendix A.1.

5.2 Impact of visual input on QA reasoning

To investigate whether C2R genuinely references visual content during utilizing sub-QAs, we compare its performance against the baseline under two conditions: standard and blind, depending on the presence or absence of visual inputs. In Table 3, most cases show that the performance improvements are more pronounced in the standard settings than in the blind settings, confirming that our method effectively incorporates visual information. We also compare C2R with different prompting methods for Text-only QA: Chain-of-Thought (Wei et al., 2022) (CoT) and CoT-SC (Wang et al., 2022a), in Appendix A.4.

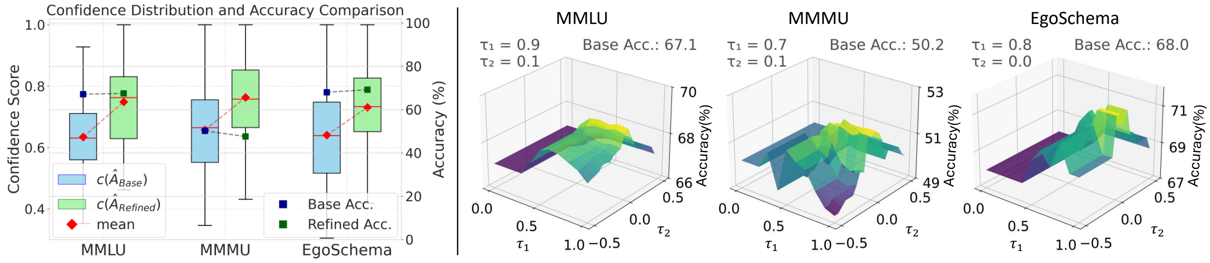


Figure 4: **Left:** Comparison of accuracy and average confidence scores for base and refined answers. While the accuracy difference is marginal, the average confidence score for refined answers increases significantly compared to base answers. **Right:** Accuracy results corresponding to different confidence threshold values for τ_1, τ_2. 'Base Acc.' denotes the accuracy of the baseline model (Qwen2.5-VL).

5.3 Confidence inflation problem

While confidence scores are crucial for selecting the final answer, we observe that using sub-QAs often increases confidence scores even for incorrect answer candidates, leading to inaccurate reasoning. We call this problem as *confidence inflation*. As illustrated in Figure 4 (Left), although the accuracy gap between base and refined answers is minimal (-0.3% on average), the mean confidence of refined answers is significantly inflated–by an average of 0.11. We hypothesize that the model tends to overestimate potentially incorrect information and reasons based on it. Consequently, adding sub-QAs to the input shifts the distribution of confidence scores, making it unreliable to rely solely on high confidence scores in refined answers. Due to this confidence inflation, the comparison between the base and refined answers requires careful handling, motivating the use of a confidence threshold τ_2 during the answer selection process. For a more detailed analysis of the criteria for selecting the best answer, see Appendix A.7.

5.4 Threshold Analysis

We report the results with the confidence thresholds τ_1 and τ_2 via a grid search on the validation sets (see Appendix D). To ensure these hyperparameters are not overfitted and to understand their impact, we conduct a sensitivity analysis on the test set. The performance landscape in Figure 4 (Right) offers insights into benchmark characteristics. A higher optimal τ_1 suggests that base answers are often insufficient, necessitating more complex reasoning. Conversely, a higher optimal τ_2 indicates that refined answers from sub-QAs are less reliable than base answers for that benchmark. For instance, on MMMU, a strategy that aggressively favors refined answers (e.g., high τ_1, low τ_2) leads to a performance drop compared to the baseline. This highlights the importance of our selective refinement mechanism, as indiscriminately incorporating all sub-QAs can degrade overall accuracy.

Model	MMLU	MMMU	EgoSchema
Baseline	67.1	62.6	68.0
C2R (Optimal τ)	69.2	65.7	71.5
C2R (Fixed τ)	69.1	65.6	71.2

Table 4: Performance comparison between optimally-tuned and fixed thresholds on test sets. The results show minimal degradation, highlighting the robustness of our method.

Furthermore, our analysis reveals that the framework's performance is stable across a range of threshold values. We found that using a fixed threshold pair ($\tau_1 = 0.7, \tau_2 = 0.1$) across all benchmarks results in performance nearly identical to that achieved with individually optimized thresholds. As shown in Table 4, the performance difference is marginal, demonstrating the generalizability of our approach and reducing the need for extensive per-dataset tuning. This analysis confirms the robustness of our chosen thresholds.

6 Conclusion

In this work, we address the challenge that sub-QAs do not always enhance QA reasoning. To tackle this, we propose Confidence-guided Refinement Reasoning (C2R), a model-agnostic framework that selectively curates sub-QAs into multiple subsets to construct diverse reasoning paths, rather than using all sub-QAs indiscriminately. By selecting the most reliable answer from multiple candidates based on confidence scores, our approach achieves higher accuracy than prior methods. We also find that sub-QAs—regardless of their actual relevance—tend to inflate the confidence of the resulting answers. To mitigate this, C2R employs appropriate confidence thresholds to guide the selection process, thereby reducing the risk of relying on incorrect sub-QAs. Extensive experiments on various benchmarks demonstrate that C2R consistently improves performance across models without requiring additional training.

Limitations

In C2R, the Refiner identifies more reliable answers by exploring diverse reasoning paths compared to single-step reasoning, and the Answer Selector ultimately chooses the most accurate answer. While C2R provides remarkable performance, it does have some limitations. First, although our framework effectively reduces the likelihood of utilizing suboptimal sub-QAs, it does not entirely prevent their generation. Second, while the depth of the reasoning process can be extended beyond two steps, this work does not explore such multi-level refinement. Third, since most benchmarks do not provide ground-truth sub-QAs, separate examples would need to be prepared for few-shot evaluation, which is beyond the scope of this paper. In future work, we plan to explore methods to further reduce the likelihood of generating low-quality sub-QAs.

Acknowledgements

This work was partly supported by the IITP (RS-2021-II212068-AIHub/10%, RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/15%, RS-2022-II220953-PICA/15%), NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/10%, RS-2024-00358416-AutoRL/5%), KEIT (RS-2024-00423940/10%), and Gwangju Metropolitan City (Artificial intelligence industrial convergence cluster development project/10%) grant funded by the Korean government.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 17682–17690.

Rohan Choudhury, Koichiro Niinuma, Kris M Kitani, and László A Jeni. 2023. Zero-shot video question answering with procedural programs. *arXiv preprint arXiv:2312.00937*.

Deniz Engin and Yannis Avrithis. 2023. Zero-shot and few-shot video question answering with multi-modal prompts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2804–2810.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A survey of confidence estimation and calibration in large language models. *arXiv preprint arXiv:2311.08298*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Preprint*, arXiv:2101.02235.

Tanmay Gupta and Aniruddha Kembhavi. 2023. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Zaid Khan, Vijay Kumar b g, Samuel Schulter, Manmohan Chandraker, and Yun Fu. 2023. Exploring question decomposition for zero-shot VQA. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kangsan Kim, Geon Park, Youngwan Lee, Woongyeong Yeo, and Sung Ju Hwang. 2024. Videoicl: Confidence-based iterative in-context learning for out-of-distribution video understanding. *arXiv preprint arXiv:2412.02186*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024. Llava-onevision: Easy visual task transfer. *Preprint*, arXiv:2408.03326.

Zhaohe Liao, Jiangtong Li, Li Niu, and Liqing Zhang. 2024. Align and aggregate: Compositional reasoning with video alignment and answer aggregation for video question-answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13395–13404.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. *Advances in neural information processing systems*, 27.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. *Advances in Neural Information Processing Systems*, 36:46212–46244.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Dídac Surís, Sachit Menon, and Carl Vondrick. 2023. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Kohei Uehara, Nan Duan, and Tatsuya Harada. 2022. Learning to ask informative sub-questions for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4681–4690.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

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. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022a. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024b. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhecan Wang, Haoxuan You, Yicheng He, Wenhao Li, Kai-Wei Chang, and Shih-Fu Chang. 2022b. Understanding me? multimodal evaluation for fine-grained visual commonsense. *arXiv preprint arXiv:2211.05895*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa: Next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9777–9786.

Shangzi Xue, Zhenya Huang, Jiayu Liu, Xin Lin, Yuting Ning, Binbin Jin, Xin Li, and Qi Liu. 2024. Decompose, analyze and rethink: Solving intricate problems with human-like reasoning cycle. *Advances in Neural Information Processing Systems*, 37:357–385.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Haoxuan You, Rui Sun, Zhecan Wang, Long Chen, Gengyu Wang, Hammad A. Ayyubi, Kai-Wei Chang, and Shih-Fu Chang. 2023. Idealgpt: Iteratively decomposing vision and language reasoning via large language models. *Preprint*, arXiv:2305.14985.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *Preprint*, arXiv:2311.16502.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4995–5004.

We present the following details that are not included in the main manuscript:

- **Additional Analyses**: We present additional analyses.
- **Benchmarks**: We provide detailed information on the benchmarks.
- **Details of Models**: We describe details of the models used in our experiments.
- **Details of Experiment Settings**: We present additional information on the experiment settings.
- **Prompt Designs**: We provide the prompts used for inference.

A Additional Analyses

A.1 Impact of the number of reasoning paths

We experimentally determine the optimal number of reasoning paths (i.e., K). Intuitively, having more reasoning paths leads to higher performance, and the results in Table 5 support this. Considering computational efficiency, we set the number of reasoning paths to $K = 4$ for our main experiments.

	K	**MMLU**	**MMMU**	**EgoSchema**
Baseline	-	67.1	50.2	68.0
C2R	1	+1.6	+1.1	+2.2
	2	+1.7	+1.2	+2.5
	4	+2.1	**+1.7**	**+3.5**
	8	**+2.2**	**+1.7**	+3.2

Table 5: Ablation of the number of reasoning paths. We denote the number of reasoning paths as K and find that $K = 4$ is best in our settings. All experiments are conducted using Qwen2.5.

A.2 Bootstrap significance test

To formally validate the robustness of our results, we conducted a bootstrap significance test to ascertain whether the performance gains of our C2R framework are statistically significant. We test against the null hypothesis (H_0) that there is no true performance difference between our framework and the baseline, using a significance level of $\alpha = 0.05$. We generate $B = 10k$ bootstrap samples by resampling from the test set with replacement. The p-value is then calculated by comparing the performance gap on the original test set, δ_{orig}, with the gaps observed on the bootstrap samples, δ_i. The formula is as follows:

$$p\text{-value} = \frac{1}{B} \sum_{i=1}^{B} \mathbb{I}(\delta_i \geq \delta_{\text{orig}})$$

where $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition is true and 0 otherwise.

Model	**MMLU**	**MMLU-Pro**	**StrategyQA**	**MMMU**	**EgoSchema**
Qwen2.5	0.0002	<0.0001	0.0037	0.0316	0.0013
Llava-Onevision	0.0076	<0.0001	<0.0001	0.0004	<0.0001
Gemma 3	0.0118	0.0006	<0.0001	0.0039	0.0131
Qwen2 2b	0.0390	0.0037	—	0.1179	0.0027
GPT-4o	0.0172	—	—	0.0314	0.0061

Table 6: The p-values from the bootstrap significance test ($B = 10k$) for the performance gains of C2R over the baseline. Values below 0.05 indicate statistical significance. A value of '<0.0001' indicates that no bootstrap sample showed a performance gap as large as the original.

Model	**Method**	**MMLU**	**StrategyQA**
Qwen2.5	Baseline	67.1	62.6
	C2R (Ours)	**69.2**	**65.7**
	5-shot ICL	69.5	66.1
LLaVA-Onevision	Baseline	66.2	64.5
	C2R (Ours)	**67.3**	**68.3**
	5-shot ICL	67.5	69.3

Table 7: Comparison of our zero-shot C2R framework with a 5-shot in-context learning (ICL) baseline on the MMLU and StrategyQA benchmarks. Our training-free method achieves performance competitive with the few-shot approach.

The resulting p-values for each model and benchmark are presented in Table 6. The results demonstrate that the p-values are well below the $\alpha = 0.05$ threshold in nearly all cases, with minor exceptions for Qwen2-2b. Therefore, we can confidently reject the null hypothesis (H_0) and conclude that the performance improvements achieved by our C2R framework are statistically significant.

A.3 Comparison with few-shot methods

To further contextualize the performance of our approach against methods that utilize labeled data, we conduct a new experiment comparing our zero-shot C2R framework with a competitive 5-shot in-context learning (ICL) baseline.

For the ICL setup, the models were provided with five demonstration examples sourced from a held-out split of the training data. The composition of these examples varied by dataset; for MMLU, the sub-QAs were generated by GPT-4o, whereas

	avg. # path[2]	MMLU	MMMU	EgoSchema
Baseline	1	67.1	50.2	68.0
CoT	1	65.3	49.8	62.3
CoT-SC	5	68.3	51.9	67.0
CoT-SC	40	**71.0**	**56.6**	69.0
C2R (Ours)	2.3	69.2	51.9	**71.5**
C2R (Ours)$*$	1.7	69.2	51.8	**71.5**

Table 8: Comparison with CoT-based methods (Qwen2.5-VL backbone). The total number of reasoning paths is averaged over all main questions and reported as 'avg. # path.' Bold text indicates the highest score, while underlined text represents the second-highest score.

for StrategyQA, we utilized the ground-truth sub-QAs to construct the demonstrations.

The results of this comparison are presented in Table 7. Our analysis shows that the proposed training-free C2R framework performs competitively against the 5-shot ICL method across both models and datasets. Notably, C2R achieves this level of performance without requiring any labeled examples for in-context demonstrations. This highlights the practicality and effectiveness of our approach, particularly in strict zero-shot scenarios where annotated data is unavailable or expensive to obtain.

A.4 Comparison with CoT methods

In this subsection, we compare C2R with CoT-based methods and analyze the computational cost of different approaches on MMLU. We compare our C2R framework with single-step reasoning (Baseline), Chain of Thought (CoT, (Wei et al., 2022)), and Self-Consistency with Chain of Thought (CoT-SC, (Wang et al., 2022a)) in Table 8. In the case of CoT, performance actually degraded across all benchmarks, and CoT-SC—which involves five reasoning passes—showed poor performance on EgoSchema, a dataset with long video contexts. This suggests that unconditionally incorporating unverified intermediate steps (such as sub-QAs) can not only underperform compared to simple single-step reasoning but also lead to inefficiencies.

MMLU	avg. # path[2]	Input / Generate tokens	Cost
Baseline	1	132 / 13	1
CoT	1	168 / 286	7.2
CoT-SC	5	830 / 1427	35.7
C2R (Ours)	2.3	708 / 84	5.7
C2R (Ours)$*$	1.7	442 / 52	3.5

Table 9: Cost analysis of different approaches on MMLU. We represent the cost as a value normalized by the cost of the baseline (single-step reasoning), which is set to 1. 'avg. # path' denotes average of the total number of reasoning paths.

A.5 Cost analysis

To analyze the cost of C2R, we report the average number of input and generated tokens in Table 9. The 'Cost' column is represented relative to the baseline, which is normalized to 1, and the cost of generated tokens is counted as four times higher, following the pricing of models like GPT-4o. Compared to CoT-based methods, our approach achieves competitive performance (Table 8) with lower cost. This is because C2R skips subsequent steps when the confidence of the single-step reasoning is high, avoiding complex computation for relatively simple questions. For MMLU, up to 68% of the main questions are answered using only single-step reasoning (this value varies depending on the benchmark). These results support that C2R avoids unnecessary computation and performs inference efficiently.

Furthermore, we observe that if the confidence of a refined answer is sufficiently high, it is not necessary to explore all K reasoning paths to maintain performance. For example, if the generation of refined answers is stopped when the confidence score exceeds 0.85, the overall cost can be reduced by 39% while preserving nearly the same level of performance. This result is reported as 'C2R (Ours)($*$)' in Tables 8 and 9.

A.6 Additional direct verification methods

We compare C2R with methods that use only sub-QAs verified through direct LLM prompting (LLMVERIFIED), and present the results in Table 10. We experiment with three settings: verifying only the quality of sub-questions (sub-Q), the correctness of sub-answers (sub-A), and both (sub-QA). In all cases, performance falls short of our proposed framework and, in some instances, even underperforms compared to the baseline. These results highlight that simply relying on LLM prompt-

---

[2] Due to Principle 1 of § 3.5, if the confidence of the base answer for a main question is sufficiently high, all subsequent steps are skipped, resulting in a total of 1 reasoning path. In the case of MMLU, this applies to approximately 68% of instances. When $K = 4$, the average number of reasoning paths becomes $0.68 \times 1 + 0.32 \times (1 + 4) \simeq 2.3$.

Backbone	Size	Reasoning Method	N	M	K	Text-only QA			ImageQA	VideoQA
						MMLU	MMLU-Pro	StrategyQA	MMMU	EgoSchema
Qwen2.5	7B	Baseline	-	-	-	67.1	38.6	62.6	50.2	68.0
		SINGLESUBQA	1	1	1	67.5	42.9	62.6	47.6	69.2
		EVERYSUBQA	5	5	1	67.7	42.6	65.3	47.6	69.0
		LLMVERIFIED (sub-Q)	5	2	1	67.5	42.7	64.0	48.2	69.2
		LLMVERIFIED (sub-A)	5	2	1	67.3	43.2	64.5	49.4	67.0
		LLMVERIFIED (sub-QA)	5	2	1	67.2	42.8	64.1	49.9	68.0
		C2R(Ours) (seq. prob.)	5	2	4	68.1	39.8	63.5	51.2	70.2
		C2R(Ours) (1/PPL)	5	2	4	67.8	43.4	63.6	50.7	**71.5**
		C2R(Ours) (min. token prob.)	5	2	4	**69.2**	**44.3**	**65.7**	**51.9**	**71.5**
		Δ				**+2.1**	**+5.7**	**+3.1**	**+1.7**	**+3.5**
LLaVA-Onevision	7B	Baseline	-	-	-	66.2	35.7	64.5	45.1	43.0
		SINGLESUBQA	1	1	1	64.3	36.6	67.8	44.1	40.2
		EVERYSUBQA	5	5	1	63.4	37.0	67.6	46.1	39.8
		LLMVERIFIED (sub-Q)	5	2	1	64.9	37.5	67.5	46.4	44.8
		LLMVERIFIED (sub-A)	5	2	1	65.1	36.9	68.2	47.2	45.2
		LLMVERIFIED (sub-QA)	5	2	1	64.6	37.8	68.1	47.0	45.8
		C2R(Ours) (seq. prob.)	5	2	4	67.1	38.0	66.7	47.3	43.5
		C2R(Ours) (1/PPL)	5	2	4	66.2	37.6	67.8	**47.4**	44.2
		C2R(Ours) (min. token prob.)	5	2	4	**67.3**	**38.4**	**68.3**	**47.4**	**46.0**
		Δ				**+1.1**	**+2.7**	**+3.8**	**+2.3**	**+3.0**

Table 10: **Zero-shot evaluation results of C2R.** We evaluate baseline models and our framework across various domains and benchmarks for comparison. N, M, and K denote the number of generated sub-QAs, the number of curated sub-QAs per reasoning path, and the total number of reasoning paths, respectively. The difference (Δ) denotes the score improvement of C2R over the vanilla models without sub-QAs. **Bold text** indicates the highest score and underlined text the second-highest. Colors show performance relative to baseline: blue (higher), red (lower).

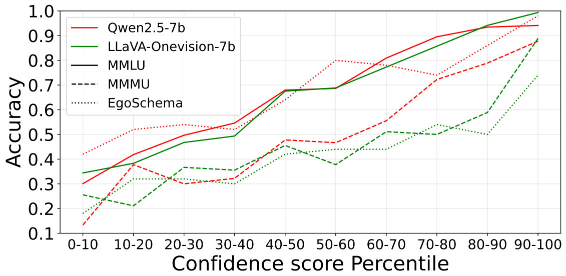


Figure 5: Correlation between confidence score and accuracy. We compute the base answer for each data instance, then sort these instances in ascending order of confidence and divide them into 10 bins. The results demonstrate a strong correlation between a response's confidence score and accuracy.

Method	MMLU	MMMU	EgoSchema
Baseline	67.1	50.2	68.0
C2R (Normalized)	68.9	51.4	71.2
C2R (Ours)	**69.2**	**51.9**	**71.5**

Table 11: Performance comparison on Qwen2.5 using normalized confidence scores versus our proposed threshold-based method. While normalization improves upon the baseline, our approach remains superior.

ing to validate intermediate steps is neither fully reliable nor consistently effective. The prompt used for verifying sub-QAs can be found in Appendix E.

A.7 Discussion of confidence score metrics

We compare three different metrics for calculating the confidence score of a generated answer: 1) minimum token probability (§3.1); 2) sequence probability (i.e., the generation probability of the entire sequence); and 3) the reciprocal of perplexity (i.e., 1/PPL). As shown in Table 10, using the minimum token probability consistently outperforms the other metrics in all cases. This supports the suitability of minimum token probability as a robust criterion for measuring confidence.

Also, for our framework to function effectively, the fundamental statement must hold that *if a model assigns a high confidence score to an answer, the answer is more likely to be correct.* Since we use the minimum token probability of an answer as its confidence score, it is necessary to examine the correlation between this probability and the actual accuracy. To validate this, we compute the base answers and their corresponding confidence scores (i.e., minimum token probability) for all data instances and sort them in ascending order based on confidence. We then divide the sorted instances into 10 bins and calculate the accuracy for each bin.

As shown in Figure 5, accuracy increases with confidence. Also, the Pearson correlation coefficient, averaged across all models and benchmarks, is as high as 0.95. This confirms that the minimum token probability of an answer is a reliable metric for selecting the best answer.

We also tested an alternative method for comparing scores from base and refined answers: normalization. In an ablation study, we normalized both scores (zero mean, unit variance) and selected the one with the higher value, foregoing the threshold τ_2. Table 11 shows that while normalization beats the baseline, our threshold-based method is superior. This validates our design, as it implies that simple normalization is insufficient to align the different confidence distributions of base and refined answers. Our threshold mechanism is therefore essential for handling these differences.

B Benchmarks

We evaluate models on five challenging QA benchmarks:

MMLU (Hendrycks et al., 2020) is a benchmark designed to measure a model's multitask accuracy across 57 diverse tasks like elementary mathematics, history, and computer science. It demands extensive world knowledge and problem-solving skills from models. The benchmark aims to analyze models broadly and identify their key shortcomings in academic and professional understanding. MMLU is MIT licensed.

MMLU-Pro (Wang et al., 2024b) is an enhanced and more challenging version of MMLU that integrates more reasoning-intensive questions and eliminates noisy ones. It spans 14 domains with over 12,000 questions and features ten answer choices to assess true understanding better. MMLU-Pro is designed to more effectively discriminate between advanced language models as their performance on the original MMLU has plateaued. MMLU-Pro is MIT licensed.

StrategyQA (Geva et al., 2021) is a QA dataset where the reasoning steps needed to answer questions are implicit and must be inferred. Its data collection process elicits creative questions and includes adversarial filtering. Each of its 2,780 examples provides the question, a reasoning decomposition, and supporting evidence paragraphs. StrategyQA is MIT licensed.

MMMU (Yue et al., 2024) is a benchmark with 11,500 multimodal questions from college exams and textbooks across six core disciplines like Science, Engineering, and Art & Design. It assesses expert-level multimodal understanding, requiring college-level knowledge and reasoning with diverse image types like charts and diagrams. MMMU challenges models on advanced perception and domain-specific reasoning. MMMU is Apache License 2.0 licensed.

EgoSchema (Mangalam et al., 2023) is a very long-form video question-answering dataset from Ego4D, designed to evaluate long video understanding. It includes over 5,000 human-curated multiple-choice questions on more than 250 hours of real-world video depicting natural human activities. EgoSchema specifically tests a model's ability to comprehend extended temporal structures and complex behaviors in video. EgoSchema is open-sourced under the Ego4D license.

C Details of Models

In this section, we describe three vision-language foundation models, each based on a different LLM, that are used in our experiments: Qwen2.5-VL (Yang et al., 2024), Gemma 3 (Team et al., 2025), and LLaVA-Onevision (Li et al., 2024). We use the checkpoint provided by huggingface.

Qwen2.5-VL is a vision-language model series based on Qwen2.5 LLM, showcasing significant advancements in visual understanding, object localization, and long-video comprehension. It introduces dynamic resolution processing and absolute time encoding, enabling efficient handling of images with varying sizes and videos spanning several hours. Also, Qwen2.5 is a versatile series of large language models significantly improved through expanded pre-training (18 trillion tokens) and advanced post-training techniques, including supervised finetuning and reinforcement learning. These enhancements lead to strong performance in reasoning, instruction following, long-text generation, and structured data analysis.

Gemma 3 is a multimodal extension of the Gemma model family, available in sizes from 1B to 27B parameters, with added vision capabilities and support for long contexts up to 128K tokens. Its architecture is optimized for memory efficiency by increasing the proportion of local attention layers and shortening local attention spans. A new post-training method and distillation enhance its performance in math, chat, multilingual understanding, and instruction following. All models are openly released to the community.

LLaVA-Onevision is a family of open large multimodal models designed to unify insights from data, model architecture, and visual representations. It is the first single model to advance performance across single-image, multi-image, and video scenarios simultaneously. The model also exhibits strong transfer learning across modalities, enabling emergent capabilities such as video understanding through image-to-video task transfer.

D Details of Experiment Settings

When evaluating on multiple-choice QA benchmarks, we include all answer choices in the input and have the model select the correct one. For VideoQA, we sample each video at 1fps; for videos over 32 seconds, we uniformly sample 32 frames from the entire video. We report performance on the validation set based on a single run; however, for datasets without a standard validation split (StrategyQA) or with very few validation examples, we evaluate on another split instead. For MMLU-Pro, since the validation set contains only 70 examples, we use the test set instead. As of May 2025, submissions to the StrategyQA leaderboard are impossible, so we report results on the training set. We tuned the confidence thresholds, τ_1 and τ_2, as hyperparameters on a validation set for each benchmark. The optimal values were identified through a grid search with a step size of 0.1. The search range for τ_1, which is compared against a confidence probability, was $[0, 1]$. For τ_2, representing the difference between two probabilities, the range was $[-1, 1]$. For datasets without an official validation split, we held out 10% of the training data for this purpose. For the closed-source models (i.e., GPT-4o), we enable the 'logprobs' option to obtain the probability of each generated token and use the 'gpt-4o-2024-08-06' version.

E Prompt Designs

We briefly describe the prompts used for generating sub-QAs, base answers, and refined answers:

- We use the prompt shown in Figure 6 when generating sub-questions.
- For obtaining sub-answers, we use the following simple prompt: "{sub-question} Answer in a maximum of one sentence."
- We design prompts to generate base answers for open-ended and multiple-choice QA tasks, as shown in Figure 7. For open-ended QA, we use "Question: {main question} Answer the question using a single word or phrase." For multiple-choice QA, the options are added after the main question: "A. {option A} B. {option B} … X. {option X}." Option X refers to the last option, with the total number of options varying depending on the question.
- For refined answers, we use the prompt shown in Figure 8.
- The prompt used to verify sub-QAs in the LLMVERIFIED method is presented in Figure 9.

F Failure Cases in sub-QA Generation

While our C2R framework demonstrates strong performance, its effectiveness is inherently dependent on the quality of the initial sub-QA generation. The framework's limitations become apparent when this process yields incomplete or flawed information, as we analyze in this section.

The primary failure modes revealed by our analysis are:

- **Irrelevant or semantically drifting sub-questions:** Sub-questions #2 and #4 are only tangentially related to the core reasoning path. They define "mascot" in a general sense or explore irrelevant aspects of a mascot's "diet," which does not help solve the specific question.
- **Incomplete sub-answers:** Sub-question #1 is relevant, but its answer crucially omits the key information needed for reasoning—the name of the high school Jindal attended.
- **Redundancy and semantic overlap:** Sub-questions #4 and #5 are semantically similar. While we note that exact duplication of sub-questions is rare (occurring in less than 0.1% of cases), this kind of semantic overlap can lead to inefficient reasoning.

In the example shown, only sub-QA #3 provides a directly useful fact. Due to the lack of critical connecting information from other sub-QAs, the Refiner failed to construct a correct reasoning path, ultimately leading to an incorrect output in most refined candidates.

```
### Instruction
Your task is to decompose a given question (or instruction) Q into sub-questions.
You need to generate {N} sub-questions that will help you answer the given Q.
[optional] Also, a single or multiple image/video(s) may be given.
Given Q, you need to generate sub-questions considering what to focus on in the image/video(s).
Please note that: You should output ONLY multiple sub-questions as shown in the following format.
### Format:
1. ...
2. ...
...
{N}. ...

### Input
The given question (or instruction) Q : '''{main question}'''

The decomposed sub-questions for Q is:
```

Figure 6: The prompt for generating sub-questions. Blue text is excluded from the prompt when performing Text-only QA. {N} denotes the number of sub-questions to be generated.

```
{main question}
Answer the question using a single word or phrase.
```

(A) open-ended

```
{main question}
A. {option A}
B. {option B}
…
X. {option X}
Answer with the option's letter from the given choices directly.
```

(B) multiple-choice

Figure 7: The prompt for generating base answer A_{base}. **(A)** corresponds to open-ended questions, while **(B)** represents multiple-choice questions. Option X refers to the last option, with the total number of options varying depending on the question.

```
You'll be given the <Main question>, the previous answer to <Main question>, and the sub-QA results as context.
Your task is to answer the next question or instruction correctly, referring to the model's previous answer and the sub-QA context.
<Main question>: {main question}
The previous answer to <Main question>: {A_base}
sub-QA context:
<sub-question 1>: {sub_question 1}          The answer to <sub-question 1}>: {sub_answer 1}
…
<sub-question M>: {sub_question M}          The answer to <sub-question M}>: {sub_answer M}

Please answer the following question or instruction based on the context.
<Main question>:
{prompt_base}
```

Figure 8: The prompt for generating refined answer $\hat{A}_{\text{refined}}$. The placeholder '{prompt_base}' (referring to the prompt used to generate the base answer) is detailed in Figure 7.

You are given the following: main_q: a main question, an image(s) or video of the main question,
and a list of sub-questions (sub_q_list) that are intended to be decomposed from the main question.
Your task is to evaluate how well each sub-question is decomposed based on the following criteria:
Relevance: Is the sub-question clearly related to the main question?
Usefulness: Does the sub-question provide necessary information for solving the main question?
Form: Does the sub-question maintain the format of a proper question?

After evaluating, output only the indexes of the two best sub-questions in the last line, based on the criteria above, in the format:
best_index_1, best_index_2
Note: In the last line, do not include any explanation or punctuation. Use zero-based indexing.

Output Format:
First, explain your evaluation process and reasoning: which sub-questions are most relevant and useful.
Then, in the final line, output the indexes of the two best sub-questions, using zero-based indexing (0 - {N}), in the format:
best_index_1, best_index_2
(Do not include any explanation or punctuation on the final line.)

Input:
main_q: {main question}
sub_q_list: {sub_question_list}

Output:

(A) sub-questions verification prompt

You are given the following: a list of sub-questions (sub_q_list) and a list of sub-answers (sub_a_list) where each answer corresponds to the sub-question at the same index. Your task is to evaluate how well each sub-answer addresses its corresponding sub-question with the provided image(s) or video, base on the following criteria:
Correctness: Does the sub-answer correctly and clearly respond to the sub-question?
Consistency: Is the sub-answer consistent with the information available?
Completeness: Is the sub-answer complete and specific, rather than vague or partial?

Output Format:
First, explain your evaluation process and reasoning: which sub-answers are most accurate and why.
Then, in the final line, output the indexes of the two most accurate sub-answers, using zero-based indexing (0 - {N}), in the format:
best_index_1, best_index_2
(Do not include any explanation or punctuation on the final line.)

Input:
sub_q_list: {sub_question_list}
sub_a_list: {sub_answer_list}

Output:

(B) sub-answers verification prompt

You are given the following: main_q: a main question, an image(s) or video of the main question,
sub_q_list: A list of sub-questions intended to help answer the main question,
sub_a_list: A list of answers, each corresponding to a sub-question.

Your task is to evaluate each sub-question and its corresponding sub-answer pair based on the following criteria:
Relevance to the main question: Does the sub-question meaningfully relate to the main question and help in answering it?
Usefulness: Does the sub-question extract information that is genuinely useful for solving the main question?
Question quality: Is the sub-question well-formed and grammatically a valid question?
Answer correctness: Is the sub-answer correct, complete, and specific based on the given vision?
Answer alignment: Does the sub-answer appropriately address the intent of the corresponding sub-question?

Output Format:
First, explain your reasoning: discuss which sub-question + sub-answer pairs are most effective and why.
Then, on the final line, output the indexes of the two best sub_q/sub_a pairs, using zero-based indexing (0 - {N}), in the format:
best_index_1, best_index_2
(Use zero-based indexing. Do not include any explanation or punctuation on the final line.)

Input:
main_q: {main question}
sub_q_list: {sub_question_list}
sub_a_list: {sub_answer_list}

Output:

(C) sub-QAs verification prompt

Figure 9: The prompt for verifying sub-questions and sub-answers directly (LLMVERIFIED). These sub-QA verification prompts may not be optimal.

Table 12: A representative failure case from the StrategyQA dataset. The model fails because the generated sub-questions do not furnish the critical information needed for the required multi-hop reasoning.

Main Question	Would Bobby Jindal's high school mascot eat kibble?
Ground Truth	Yes. (*Reasoning: Bobby Jindal attended Baton Rouge Magnet High, whose mascot is a bulldog. Bulldogs are dogs, and kibble is dog food.*)
Base Answer	Yes.
Generated Sub-Question and Sub-Answer Pairs	
Sub-Q #1	Who is Bobby Jindal?
Sub-A #1	*Bobby Jindal was an American politician who served as the 47th Governor of Louisiana from 2008 to 2015 and was a Republican candidate for the 2016 presidential election.*
Sub-Q #2	What is a high school mascot?
Sub-A #2	*A high school mascot is a symbol or character representing a school, often used in sports teams and school spirit events.*
Sub-Q #3	What is kibble?
Sub-A #3	*Kibble is a dry, processed food for pets, typically made from a mixture of grains, proteins, and other ingredients, designed to be easily digestible and convenient for feeding.*
Sub-Q #4	What is the diet of a high school mascot?
Sub-A #4	*A high school mascot typically does not have a specific diet as they are not living organisms and do not consume food.*
Sub-Q #5	How do high school mascots typically eat?
Sub-A #5	*High school mascots typically eat like regular students, with meals and snacks similar to those consumed by their peers.*