Towards Efficient and Robust VQA-NLE Data Generation with Large Vision-Language Models

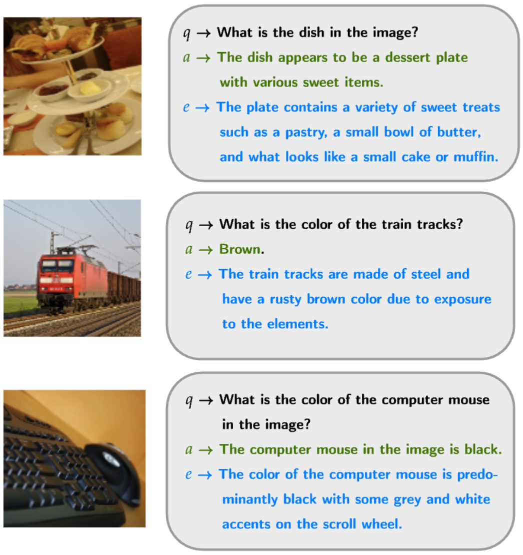
Patrick Amadeus Irawan[1], **Genta Indra Winata**[2*], **Samuel Cahyawijaya**[3], **Ayu Purwarianti**[1]
[1]Institut Teknologi Bandung [2]Capital One
[3]The Hong Kong University of Science and Technology
patrickai0309@gmail.com, genta.winata@capitalone.com,
scahyawijaya@connect.ust.hk, ayu@informatika.org

Abstract

Natural Language Explanation (NLE) aims to elucidate the decision-making process by providing detailed, human-friendly explanations in natural language. It helps demystify the decision-making processes of large vision-language models (LVLMs) through the use of language models. While existing methods for creating a Vision Question-Answering with Natural Language Explanation (VQA-NLE) datasets can provide explanations, they heavily rely on human annotations that are time-consuming and costly. In this study, we propose a novel approach that leverages LVLMs to efficiently generate high-quality synthetic VQA-NLE datasets. By evaluating our synthetic data, we showcase how advanced prompting techniques can lead to the production of high-quality VQA-NLE data. Our findings indicate that this proposed method achieves up to 20× faster than human annotation, with only a minimal decrease in qualitative metrics, achieving robust quality that is nearly equivalent to human-annotated data. Furthermore, we show that incorporating visual prompts significantly enhances the relevance of text generation. Our study paves the way for a more efficient and robust automated generation of multi-modal NLE data, offering a promising solution to the problem.

1 Introduction

Natural Language Explanation (NLE) is a valuable tool for elucidating a model's decision-making process, thereby enhancing transparency and fostering trust and accountability. This concept has been applied across various machine learning tasks (Hendricks et al., 2016; Ling et al., 2017; Kotonya and Toni, 2020; Aggarwal et al., 2021; Lu et al., 2022; Yang et al., 2015), including practical applications such as automated driving (Kim et al., 2018) and medical imaging (Kayser et al., 2022).



q → What is the dish in the image?
a → The dish appears to be a dessert plate with various sweet items.
e → The plate contains a variety of sweet treats such as a pastry, a small bowl of butter, and what looks like a small cake or muffin.

q → What is the color of the train tracks?
a → Brown.
e → The train tracks are made of steel and have a rusty brown color due to exposure to the elements.

q → What is the color of the computer mouse in the image?
a → The computer mouse in the image is black.
e → The color of the computer mouse is predominantly black with some grey and white accents on the scroll wheel.

Figure 1: Generated VQA data along with NLE of the predicted answers, offering better explainability over traditional VQA data. These are the three samples from our synthetic VQA-NLE dataset. We create a total of 66,682 unique instances of these triplets.

In the realm of vision-language tasks, explanation-rich datasets like VQA-X (Park et al., 2018), VQA-E (Li et al., 2018), VCR (Zellers et al., 2019a), e-SNLI-VE (Kayser et al., 2021), and GQA-REX (Chen and Zhao, 2022) have been instrumental in advancing vision-language NLE research. These datasets enable a deeper understanding and improved explainability of interactions within the vision-language modality, thereby enhancing the overall effectiveness of NLE in vision-language tasks, especially in Visual Question Answering (VQA).

Despite significant advancements on the topic, the scarcity of VQA-NLE data still prevails, potentially hindering further progress in the field. Existing VQA-NLE datasets (Do et al., 2021; Park et al., 2018; Zellers et al., 2019b) heavily rely on manual human annotations which is time-consuming

[*] The work was conducted outside Capital One.

Proceedings of the 31st International Conference on Computational Linguistics, pages 4323–4340
January 19–24, 2025. ©2025 Association for Computational Linguistics

and costly. This causes the data creation process inefficiency and difficult to scale underscoring the need for a more efficient method for generate VQA-NLE data (Lu et al., 2024; Li et al., 2018; Chen and Zhao, 2022).

In this work, we propose efficient and scalable methods for generating synthetic VQA-NLE data that eliminates the need for additional resource curation while maintaining quality comparable to human-generated data.[1] We introduce both single-step and multi-step approaches to produce high-quality data, utilizing visual prompts with bounding boxes to enhance focus and improve generation accuracy. With the advent of large vision language models (LVLMs) (Liu et al., 2024; Zhu et al., 2023; Bai et al., 2023), we leverage the generative capabilities of LVLMs to address current limitations for generating synthetic VQA-NLE data. Figure 1 showcases the samples of our generated VQA-NLE data. To quantitatively evaluate our method, we create an evaluation dataset and conducted a comparative analysis of various settings. Furthermore, we perform an efficiency analysis against crowdsourced data creation method to reinforce our primary objective of presenting a more efficient method for generating synthetic VQA-NLE datasets. Our contributions are three-fold:

- We propose methods to synthetically generate high-quality VQA-NLE data using LVLMs, which show a high correlation with human annotations.
- We demonstrate the impact of various synthetic VQA-NLE generation methods to identify best practices for constructing effective and efficient synthetic VQA-NLE data.
- We compare the effectiveness and efficiency of our data generation methods against human annotations for the same task. Our findings highlight the strong potential of LVLM-based synthetic VQA-NLE data generation as a viable alternative, producing high-quality data with up to 20× greater efficiency.

2 Methodology

In this section, we present a comprehensive description of our synthetic data generation methods, including detailed explanations of the three prompting pipelines employed. Formally, we define $\mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_N\}$, where $\mathcal{T}_i = (q_i, a_i, e_i)$ represent our synthetic data triplet, comprising a question q_i, an answer a_i, and an explanation e_i. We define $P_i = \{p_1, p_2, \dots\}$ as the set of prompts used to generate $\mathcal{T}_i$ with model M and input data $\mathcal{D}_i$. The process of generating each triplet $\mathcal{T}_i$ from a sample $\mathcal{D}_i$ can be expressed using inference denoted by the function f that relates these variables:

$$\mathcal{T}_i = f(M, \mathcal{D}_i, P_i). \tag{1}$$

Next, we employ three distinct prompting pipelines to generate $\mathcal{T}_i$, guided by the formulation in Equation 1. Each pipeline involves a two-step process: First, we produce contextually relevant q_i and a_i from the input data $\mathcal{D}_i$. Second, we generate an explanation e_i to justify the answer a to the corresponding question q_i. The distinctiveness of each pipeline is attributed to the specific prompting and pre-processing techniques used, as illustrated in Figure 2 and elaborated in the following explanations.

2.1 Single-Step

We generate synthetic data using a straightforward prompting technique. Initially, we craft a single prompt template p_i to produce $\mathcal{T}_i$ in a single inference step. Next, we create a question prefix pool using stratified sampling based on the desired prefixes and their respective proportions, specified as hyper-parameters. We then select a sampled question prefix q_{p_i} and incorporate it into the prompt via a formatting process, denoted as $\Phi(p_i, q_{p_i})$. Hence, we can revise Equation 1 to represent this Single-Step instruction pre-processing as:

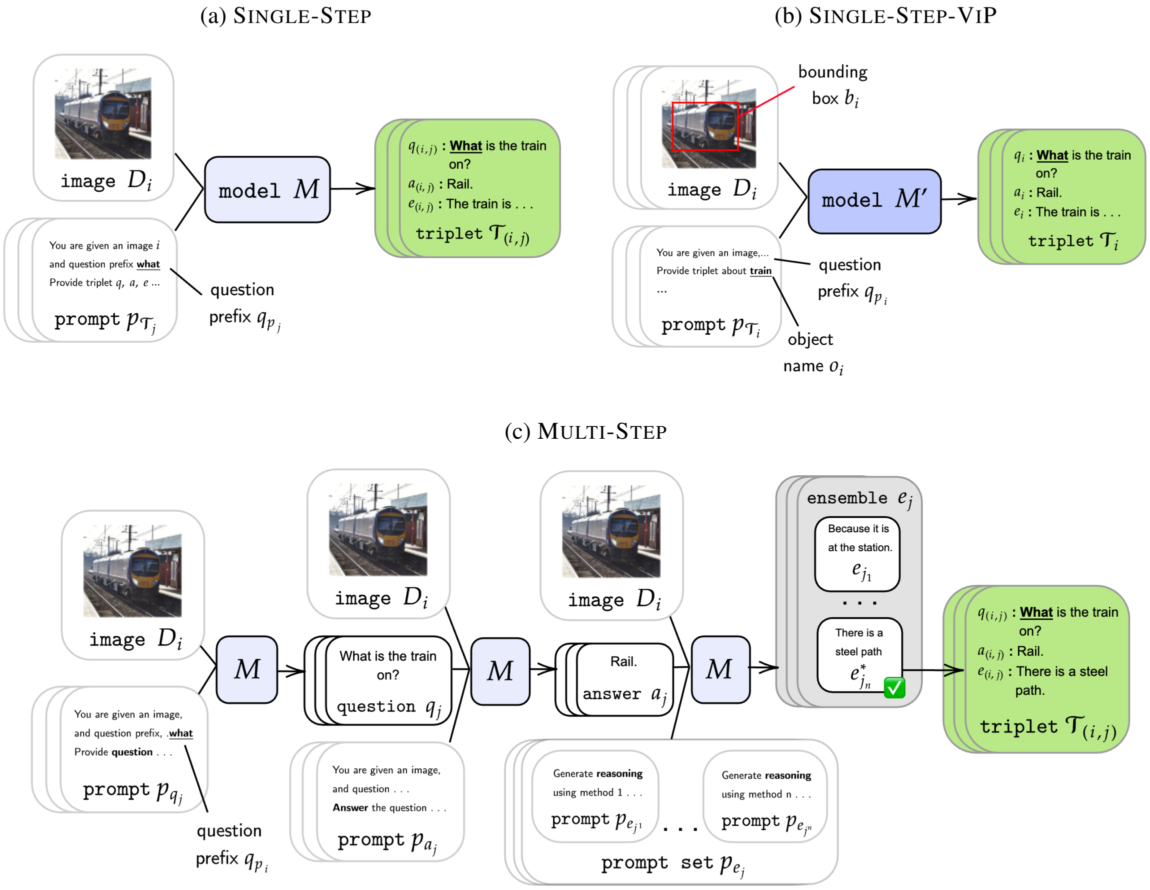
$$\mathcal{T}_i = f(M, \mathcal{D}_i, \Phi(p_i, q_{p_i})). \tag{2}$$

The details for Single-Step approach and the corresponding prompt template for this method are provided in Appendix 1.

2.2 Single-Step-ViP

Several studies have explored the idea of enhancing image interpretation by considering the objects they contain. Zhu et al. (2016) establish connections between object regions and textual responses, while Lovenia et al. (2023) introduce a novel approach by utilizing object lists to regenerate question templates. With the emergence of visual-prompt-aware models, such as Cai et al. (2024), we aim to employ a similar strategy to enhance the relevance

[1] The code is available at https://github.com/patrickamadeus/vqa-nle-llava. The dataset can be accessed at https://huggingface.co/datasets/patrickamadeus/vqa-nle-llava


(a) SINGLE-STEP

(b) SINGLE-STEP-VIP

(c) MULTI-STEP

image D_i
You are given an image i and question prefix what Provide triplet q, a, e ...
prompt $p_{\mathcal{T}_j}$
question prefix q_{p_j}
model M
$q_{(i,j)}$: What is the train on?
$a_{(i,j)}$: Rail.
$e_{(i,j)}$: The train is . . .
triplet $\mathcal{T}_{(i,j)}$
bounding box b_i
image D_i
You are given an image,... Provide triplet about train ...
prompt $p_{\mathcal{T}_i}$
question prefix q_{p_i}
object name o_i
model M'
q_i : What is the train on?
a_i : Rail.
e_i : The train is . . .
triplet $\mathcal{T}_i$
image D_i
You are given an image, and question prefix, what Provide question . . .
prompt p_{q_j}
question prefix q_{p_i}
M
What is the train on?
question q_j
You are given an image, and question . . . Answer the question . . .
prompt p_{a_j}
Rail.
answer a_j
Generate reasoning using method 1 . . .
prompt $p_{e_{j_1}}$
Generate reasoning using method n . . .
prompt $p_{e_{j_n}}$
prompt set p_{e_j}
ensemble e_j
Because it is at the station.
e_{j_1}
There is a steel path
$e^*_{j_n}$
$q_{(i,j)}$: What is the train on?
$a_{(i,j)}$: Rail.
$e_{(i,j)}$: There is a steel path.
triplet $\mathcal{T}_{(i,j)}$

Figure 2: An illustrative example of how we construct $\mathcal{T}(q, a, e)$ with three different approaches (SINGLE-STEP, SINGLE-STEP-VIP, and MULTI-STEP) using model M, prompt p, and image D. Each D_i is used to generate up to j triplets using different formatted prompt p_j with supporting instruction (e.g., question prefix).

and quality of $\mathcal{T}$ through the integration of regional subcontext extracted from image scene graphs.

At a high level, we follow the SINGLE-STEP approach for generating $\mathcal{T}$ with minor adaptation. This adjustments involve the following steps: First, we annotate the dataset D with bounding box b as our additional visual prompt (ViP), denoted as $\Theta(D, b)$. Second, we format p with not only question prefix q_p, but also object name o as additional instruction contexts to guide the inference towards the relevant object, denoted as $\Phi(p, q_p, o)$. Lastly, we employ an instruction-tuned model M' trained for visual prompts, instead of the base model M. Thus, we showcase this visual prompt fusion by modifying Eq. 2 as follows:

$$\mathcal{T} = f(M', \Theta(D, b), \Phi(p, q_p, o)). \qquad (3)$$

The detailed pseudocode for SINGLE-STEP-VIP approach and the corresponding prompts are available in Appendix 2.

2.3 MULTI-STEP

We generate q_i, a_i, and e_i sequentially rather the previous method's exhaustive generation of all $\mathcal{T}_i$ components in a single step. We aim to enhance the e component by adopting lightweight re-ranking self-consistency (Jain et al., 2024), a framework that enables the ensemble of open-ended generation by selecting the i-th generation with the highest average fractional generalized similarity score GSC_{Sim} among the other $K-1$ outputs, denoted as $GSC_{Sim}(i) = \frac{1}{K-1}\sum_{j=1,j\neq i}^{K} Sim(i,j)$. We change the similarity function Sim by utilizing encoder model (Song et al., 2020) instead of the unigram consistency score.

Initially, we design multiple prompt templates, including p_q for question generation, p_a for answer generation, and a set $P_e = \{p_{e_1}, p_{e_2}, ...\}$ consisting of multiple explanation generation instructions, whose outputs are combined using GSC_{Sim}. We then format each p with its required contexts to construct the subsequent output (e.g., p_q with q_p,

p_a with q, and so forth). Finally, we combine multiple explanations generated from P_e, denoted as $\Psi(e_1, e_2, ...)$, to determine the optimal e^* to accompany the previously generated q and a. The sequential generation process can be expressed as:

$$q = f(M, D, \Phi(p_q, q_p)). \tag{4}$$

$$a = f(M, D, \Phi(p_a, q)). \tag{5}$$

$$e^* = \Psi \begin{pmatrix} f(M, D, \Phi(p_{e_1}, q, a)), \\ f(M, D, \Phi(p_{e_2}, q, a)), \\ \dots \end{pmatrix}. \tag{6}$$

$$\mathcal{T} = (q, a, e^*) . \tag{7}$$

The details for MULTI-STEP and the corresponding prompts are available in Appendix 3.

3 Experimental Setup

In this section, we present the dataset, LVLM, and prompts used to artificially generate our data. These components are used in three experimental settings that are evaluated using the setup to analyze the strengths and limitations of each approach.

3.1 Dataset

We utilize the GQA dataset (Hudson and Manning, 2019) by sampling 10k images from the total collection of 110k along with their associated scene graphs to produce the evaluated data. We ensure that each of the selected images has a corresponding scene graph to facilitate the SINGLE-STEP-VIP setting. Additionally, we apply a filtering criterion to the scene graphs, considering only objects with bounding box areas above a certain area threshold. This filtering step aims to exclude very small and unclear objects, addressing two issues: (1) not all images have corresponding graph objects, and (2) some object graphs have excessively small areas, which could hinder their usefulness during inference.

3.2 Models

We employ three LLaVA-1.5 (Liu et al., 2024) variants: (a) LLaVA-1.5-7B, (b) LLaVA-1.5-13B, and (c) ViP-LLaVA-13B (Cai et al., 2024). The models integrate the pre-trained CLIP ViT-L/14 visual encoder with Vicuna via a simple projection matrix. LVLM (c) is a fine-tuned (b) with annotation-rich image input. The LLaVA with Vicuna backbone is selected due to its superior instruction-following capability compared to other variants, as highlighted in Shao et al. (2024).

In the SINGLE-STEP setting, we assess performance differences between (a) and (b) to probe the base model ability in following basic instruction to produce our data. We then proceed with (b) to run MULTI-STEP setting for more advanced prompting technique. Finally, we employ (c) in conjunction with the boundary box as an additional input in SINGLE-STEP-VIP setting.

3.3 Prompts

We structure our prompt templates by referring to the LLM-as-a-Judge approach, which has demonstrated the effectiveness reference-guided techniques in evaluation tasks (Zheng et al., 2023). Our contribution lies in adapting this template for generation purposes, with a specific focus on improving LVLM's rule adherence. We employ a straightforward prompting strategy in the SINGLE-STEP setting and make necessary adjustments for additional object names and annotation contexts in the SINGLE-STEP-VIP setting. Furthermore, we explore more advanced prompting methods in the MULTI-STEP setting, ensembling multiple reasoning paths such as CoT (Wei et al., 2023) and ReAct (Yao et al., 2023). The prompt templates are provided in Appendix C.

3.4 Experiment Settings

We define our methodology formulation into the following experiment settings as following:

- SINGLE-STEP-7B: We implement the SINGLE-STEP method, utilizing the LLaVA-1.5-7B model to assess the smallest model's capability in generating the dataset.
- SINGLE-STEP-13B: This setting is similar to the previous one, but employs the LLaVA-1.5-13B model to investigate the impact of model scale differences on synthetic data quality and similarity.
- SINGLE-STEP-VIP: We introduce visual-prompt-infused input data with the ViP-LLaVA-13B model as our instruction-tuned model.
- MULTI-STEP: We apply the sequential ensemble method using the LLaVA-1.5-13B model.
- HUMAN: We conduct human annotations with 10 annotators to generate 10 triplets each, enabling a comparison with our synthetic data pipeline.

Model	Triplet				Vocabulary Size			Avg. Sentence Length		
	# Valid Instance	# Unique	Valid %	Unique %	q	a	e	q	a	e
Single-Step-7B	19,309	15,328	94.2%	79.4%	5,196	11,495	12,645	8.7	8.0	28.1
Single-Step-13B	20,501	16,847	100.0%	82.2%	6,170	10,198	11,448	8.8	4.4	22.3
Single-Step-ViP	18,458	16,968	90.0%	91.9%	5,437	10,066	11,717	8.5	6.2	18.1
Multi-Step	20,501	17,539	100.0%	85.6%	8,163	8,345	14,340	10.5	2.3	42.2

Table 1: Experiment results across four different settings in a total of 66,682 valid unique instances over 82,004 expected amount. To determine the unique triplets, post-processing is conducted to filter out duplicate triplets, which has the same all three q, a, and e components. The vocabulary size and sentence length is derived by using a simple tokenization to separate whitespace on the cleaned sampled corpus without punctuation.

Our goal is to produce 20,501 synthetic triplets for each experiment. The specific hyper-parameters and engine details can be found in Appendix D.

3.5 Evaluation Settings

We evaluate 2,004 out of 82,004 synthetic triplets, comprised of 501 per setting. We then apply post-processing to assess both validity rate and time efficiency ($\bar{t}$) across all triplets. Finally, we use only the valid triplets from the 501 data points for similarity evaluation and sample 50 of them for quality evaluation.

Quality Evaluation Quantifying the quality of synthesized explanations for VQA is challenging, so we opt for a human evaluation approach as our primary evaluation source. Three annotators assess our synthetic data following the guidelines in Appendix E, and we also calculate the inter-annotator agreement Gwet-AC2 metric to ensure consistency. We employ a human evaluation scoring framework inspired by Zhang et al. (2023), making slight adjustments to align with our evaluation objectives. We emphasize assessment on *Accuracy*, *Logical*, *Clarity*, *Detail*, and *Relevancy* criteria, with each scored on a scale from 1 (worst) to 3 (best). Details on human annotation metrics are shown in Appendix A. Furthermore, we employ BERTScore, CLIPScore, and ROUGE metrics to support our subjective evaluation in assessing the quality of generated (q, a) and e against one another and the input image.

Similarity Evaluation We perform a comparative analysis by examining the text length distribution variations in q, a, and e between our synthetic data and human-generated data. To quantify these differences, we utilize Jensen-Shannon Divergence (JSD) and Pearson correlation as our evaluation metrics. JSD is employed to measure distribution shifts within a specific range, while Pearson correlation assesses the overall trend in text length across the two datasets.

Validity and Efficiency Evaluation We perform a supplementary analysis to evaluate the reliability of each experiment set in generating valid data. Valid data, denoted as $\mathcal{T}_{\text{valid}}$, is defined as a triplet $\mathcal{T}$ whose elements collectively follow a predefined regular expression, ensuring natural sentence structure. For Single-Step-ViP, valid triplets must also not include any hidden annotation contexts (e.g., "`...in the red bounding box.`"). The validity proportion is defined as Valid $\% = \frac{|\mathcal{T}_{\text{valid}}|}{|\mathcal{T}_{\text{valid}}|+|\mathcal{T}_{\text{invalid}}|} \times 100\%$. Additionally, we conduct a comparative analysis of the time efficiency of our approach against conventional methodologies. We measure the average time $\bar{t}$ required to generate each valid data point by recording the total time for dataset creation and dividing it by the number of valid triplets, $|\mathcal{T}_{\text{valid}}|$, denoted as $\bar{t} = \frac{t}{|\mathcal{T}_{\text{valid}}|}$.

4 Results and Analysis

In this section, we present the general data generation and evaluation results for the entire dataset, as shown in Table 1. We then highlight key findings through quality and similarity analyses, using 50 sampled triplets from each setting. All discussions refer to Table 2 for similarity evaluation and Table 3 for quality evaluation.

4.1 Experiment Results

Single-Step-7B Our first setting successfully generated 94.2% valid triplets over the expected amount. Furthermore, a decent similarity is observed, with a Pearson correlation of 0.70 and a Jensen-Shannon Divergence (JSD) of 0.35. This similarity is predominantly attributed to the high similarity in the q component, as illustrated in Figure 3. However, only 79.4% of these triplets are unique, indicating a moderate scalability level

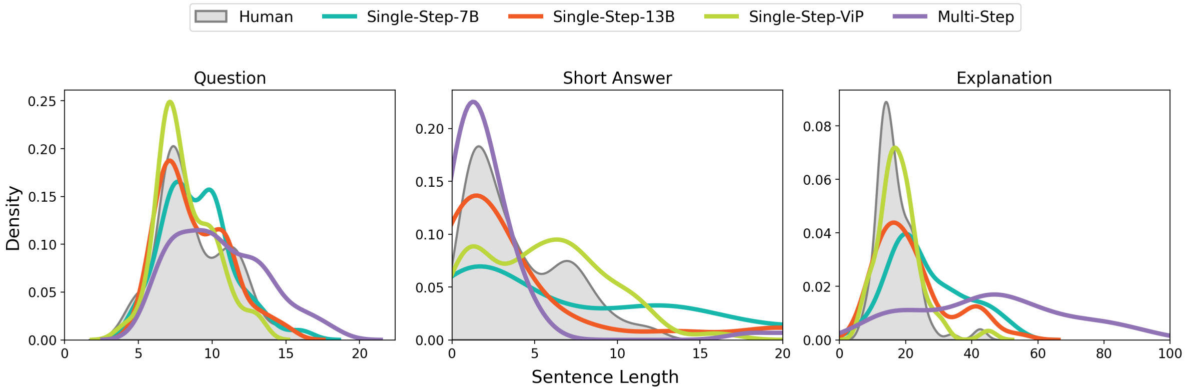


Figure 3: Comparison of density estimation for sentence length distribution across all experiment settings and human-generated $\mathcal{T}$. It provides a visual representation of the distribution differences, with more detailed numerical insights available in Table 2.

when this approach is used to generate unique synthetic data. In terms of quality, this approach achieves a score of 2.519, which is 12% short compared to human-generated data.

Single-Step-13B In contrast to the previous setting, the Single-Step setting with a larger model excels by generating 100% of the expected quantity. These triplets have an uniqueness rate of 82.5%. The similarity metrics also improve substantially, with a Pearson correlation of 0.80 and a JSD of 0.30. The q, a, and e components contribute evenly to these scores. In terms of quality, this approach scores 2.619, showing notable improvements over the smaller model in all criteria except for the detail aspect, reducing the gap to human quality to just 8%.

Multi-Step The multi-step setting generates 100% valid triplets with an improved unique rate of 85.9%. While the overall quality score slightly improves to 2.623, there is a notable drop in relevancy. This tendency for overly detailed explanations is also evident in the similarity metrics, resulting in the lowest performance among all settings, with a Pearson correlation of 0.58 and a JSD of 0.39.

Single-Step-ViP The visual-prompt infused Single-Step setting generates 90% valid triplets, 10% lower compared to Multi-Step. Despite this, a notable 92.1% unique rate is achieved indicating that visual-prompt helps to generate unique data instance. This setting also exhibits improvement in similarity, marking Pearson correlation of 0.84 and JSD of only 0.25. Furthermore, it increases quality score to 2.646, lacking only 7% compared to human. This is predominantly achieved by alleviating the lack of relevancy suffered by previous settings while maintaining decent score of other criteria and championing in clarity aspect over human.

Model	Pearson				JSD			
	q	a	e	**Avg**	q	a	e	**Avg**
Single-Step-7B	0.88	0.70	0.50	0.70	0.16	0.43	0.44	0.35
Single-Step-13B	0.93	0.75	0.75	0.81	0.18	0.41	0.32	0.30
Single-Step-ViP	0.96	0.74	0.83	0.84	0.17	0.30	0.28	0.25
Multi-Step	0.78	0.76	0.22	0.58	0.27	0.43	0.47	0.39

Table 2: Similarity metrics evaluation result for different experiments, including averages for Pearson correlation and Jensen-Shannon Divergence. The evaluation is conducted to the 501 sampled synthetic data from each experiment setting compared to human generated data.

4.2 Larger Model Improves Instruction Obedience and Generated Data Quality

Table 3 reveals that Single-Step-7B achieves the lowest scores in 4 out of 5 human evaluation criteria, resulting in the worst overall performance among all settings. By using a larger model variant, Single-Step-13B secures a 5% overall improvement, including a significant 12% boost in relevancy, without any instruction modification. Other improvements are also evident in the distribution similarity. Figure 3 illustrates a better data distribution spread in Single-Step-13B compared to Single-Step-7B, closely resembling the human-generated data distribution. This improvement is also evidenced by a 14% increase in Pearson correlation and a 13% reduction in JSD distance, as shown in Table 2. These findings highlight that the size of LLaVA-1.5 scales positively with the quality & similarity metrics improvement, leveraging the advantages of larger models: (1) better instruction following ability, resulting in reduced

Model	Human Evaluation						CLIPScore		BERTScore	ROUGE
	Accuracy	Logic	Clarity	Detail	Relevancy	**Avg. Score**	$q, a \leftrightarrow D$	$e \leftrightarrow D$	$q, a \leftrightarrow e$	$q, a \leftrightarrow e$
Human	2.82	2.79	2.86	2.93	2.88	2.86	23.33	23.13	0.745	0.45
Single-Step-7B	2.56	2.47	2.74	2.52	2.30	**2.519**	28.49	30.20	0.772	0.34
Single-Step-13B	2.71	2.51	2.86	2.44	2.57	**2.619**	28.38	30.66	0.779	0.40
Single-Step-ViP	2.57	2.53	2.91	2.59	2.63	**2.646**	24.42	25.62	0.758	0.37
Multi-Step	2.69	2.64	2.76	2.68	2.35	**2.623**	28.71	27.77	0.749	0.30

Table 3: Quality evaluation results are obtained through human and automated metrics, utilizing 100 surveyed data points from human sources and 50 sampled synthetic data points across all experiment settings. We follow the human evaluation criteria outlined in Section 3.5, where green, underlined, and red numbers denote the best, second-best, and worst results, respectively. In automated evaluation, $x \leftrightarrow y$ denotes the score calculation based on the interaction between x and y.

formatting errors and better text length control, and (2) generation of more logically relevant explanations, leading to high-quality data.

4.3 Irrelevant Logical Results in Multi-Step

The ensembling method effectively enhances the logic criterion, outperforming all other settings. This suggests that employing advanced prompting techniques such as CoT and ReAct can significantly boost LLaVA's reasoning abilities, resulting in more detailed and logically sound explanations. However, this improvement is accompanied by a trade-off, as the relevancy criterion score decreases by 9%. This finding indicates that while longer and more elaborate explanations may enhance logical coherence, they can compromise explanation precision. This observation is further substantiated by Figure 3, which illustrates a notable shift in the explanation distribution compared to other settings.

We attribute this issue to the chosen explanation sourced from either CoT or ReAct output prompts. As detailed in Appendix C, both techniques involve generating more detailed intermediary steps to reach the final conclusion. This can potentially lead to overly long outputs from the LLM, resulting in detailed but less precise explanations. These findings emphasize the importance of (1) implementing an appropriate token-limiting mechanism and (2) exploring sequential intermediary step generation to prevent overly large outputs and maintain explanation precision.

4.4 Overcoming Relevancy Issues with Visual Prompts

Enhancing the Single-Step method with visual prompts, particularly bounding boxes, results in a significant 12% increase in the relevancy score compared to the Multi-Step setting. Interestingly, all other criteria maintain impressive scores, including clarity, which surpasses human explanation quality. Furthermore, similarity evaluation reveals that Single-Step-ViP achieves the best similarity score, with a Pearson correlation of 0.84 and a minimized JSD of 0.25. This exceptional performance is notably attributed to how the explanation aspect closely follows the human-generated data distribution as depicted in Figure 3. These findings highlight that visual prompts serve as an effective guide for ViP-LLaVA, enabling it to produce logically relevant explanations while maintaining excellence in other aspects.

4.5 Automatic Quality Metrics Analysis

Recall that we employ BERTScore, CLIPScore, and ROUGE as our automated quality metrics in Table 3. While we do not explore into every detail, we highlight some interesting points that reinforce our previous analyses. The enhanced performance across all automated metrics indicates that Single-Step-13B surpasses Single-Step-7B, consistently delivering superior results in all $\mathcal{T}$ components and reinforcing the analysis from Section 4.2. Secondly, the lowest scores observed for BERTScore and ROUGE in the Multi-Step setting validate the presence of irrelevant context within e, as elaborated in Section 4.3. This observation highlights the diminished contextual relevance between q, a and e. Lastly, the worst result for CLIPScore in the Single-Step-ViP setting, which contrasts the human evaluation's best mark, can be attributed to the localization technique detailed in Section 4.4. It is important to note that CLIPScore is an image captioning metric, designed to assess how well a text contextually represents the entire image. In contrast, Single-Step-ViP captures subregions of the image to increase focus on specific contexts by utilizing bounding box. Given that both q, a, and e are more likely to represent subregions rather than the entire image, this discrepancy leads to the observed lower CLIPScore performance.

4.6 Time Efficiency

Table 4 presents the inference times required to generate 501 synthetic data points for each experiment. The results demonstrate that generating triplets $\mathcal{T}$ synthetically can achieve up to a 20× efficiency improvement compared to the human generation approach. All SINGLE-STEP-based experiments yield comparable results, while MULTI-STEP is approximately four times slower due to its sequential generation and multiple reasoning paths.

Model	$\lvert\mathcal{T}_{\textbf{valid}}\rvert$	t	$\bar{t}$
Human	501*	350m 5s*	42.1s (1×)
SINGLE-STEP-7B	476	16m 41s	**2.10s** (20.0×)
SINGLE-STEP-13B	501	19m 23s	2.32s (18.1×)
SINGLE-STEP-VIP	450	**15m 54s**	2.12s (19.9×)
MULTI-STEP	483	66m 51s	8.01s (5.3×)

Table 4: Comparison of synthetic data generation time for all experiments. The multiplier for $\bar{t}$ is computed relative to the human annotation time (*estimated time per 501 generated data), following formula in section 3.5. **Bold** and underlined times represent the most efficient and second-most efficient experiments, respectively.

5 Related Work

Explanation Generation in VQA Several studies have emphasized the generation of explanations, either manually or automatically. Manual approaches, such as VCR (Zellers et al., 2019b) and e-SNLI-VE (Do et al., 2021), employ human annotators to derive explanations from existing VQA datasets. In contrast, automatic methods like GQA-REX (Chen and Zhao, 2022) utilize functional programming, allowing automatic explanations generation which are grounded on the reasoning process and tightly couple keywords and regions of interest. Another automatic methods like VQA-E (Li et al., 2018) aligns and merges constituency parse trees from QA-caption pairs, while VQA-X (Park et al., 2018) employs separate answer and explanation models for generation. In this paper, we introduce a novel approach by proposing a unified model using single LVLM, eliminating multiple architectures need.

Neural Synthetic Data Generation In the realm of multi-modal learning, particularly in the vision-language domain, the potential of synthetic data generation has been extensively explored. Li et al. (2023) discuss the application of synthetic data across various tasks and modalities. In computer vision, GAN-based models (Karras et al., 2019) and diffusion-based approaches (Nichol et al., 2022) are utilized for image synthesis. Within natural language processing domain, several studies (Kumar et al., 2021; Chung et al., 2023; Schmidt et al., 2024) employ synthetic data to enhance in-context learning. In the joint vision-language training, Xiao et al. (2023) leverage diffusion models to generate image captions, and Lovenia et al. (2023) create intermediary data to support object hallucination analysis. The joint training aligns vision and language representations, resulting in more relevant generation (Winata et al., 2024). In this work, we primarily focus on utilizing LVLMs to evaluate their capability in generating high-quality VQA-NLE data.

6 Conclusion

In this paper, we propose efficient methods for VQA-NLE data generation that leverage LVLMs through single-step and multi-step pipelines. Our methods produce high-quality data, achieving up to 20× greater time efficiency compared to traditional human annotation, with only a slight reduction in quality that remains closely comparable to human-annotated data without further fine-tuning. We demonstrate that incorporating visual prompts significantly enhances the relevance of text generation. Additionally, we emphasize the scalability of our approach to showcase the robustness of our solution for automatically generating multi-modal NLE data on an even larger scale.

Limitations

We have identified several promising avenues for enhancing our research outcomes. However, for the scope of this study, we have chosen to limit our experiments to three distinct vision-language pretrained models. This focused approach allows us to conduct a more detailed and manageable analysis within the constraints of our current resources and time frame. By concentrating on these specific models, we aim to provide a thorough evaluation and establish a solid foundation for future research.

Ethical Considerations

Our research focuses on generating synthetic VQA-NLE. We are committed to conducting our evaluations with the highest standards of transparency and fairness. Additionally, we ensure that the generation process strictly excludes any sensitive or personal data.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065, Online. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Mu Cai, Haotian Liu, Dennis Park, Siva Karthik Mustikovela, Gregory P. Meyer, Yuning Chai, and Yong Jae Lee. 2024. Vip-llava: Making large multimodal models understand arbitrary visual prompts. *Preprint*, arXiv:2312.00784.

Shi Chen and Qi Zhao. 2022. Rex: Reasoning-aware and grounded explanation. *Preprint*, arXiv:2203.06107.

John Chung, Ece Kamar, and Saleema Amershi. 2023. Increasing diversity while maintaining accuracy: Text data generation with large language models and human interventions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics.

Virginie Do, Oana-Maria Camburu, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-snli-ve: Corrected visual-textual entailment with natural language explanations. *Preprint*, arXiv:2004.03744.

Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. 2016. Generating visual explanations. *Preprint*, arXiv:1603.08507.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506.

Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2024. Lightweight reranking for language model generations. *Preprint*, arXiv:2307.06857.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. *Preprint*, arXiv:1812.04948.

Maxime Kayser, Oana-Maria Camburu, Leonard Salewski, Cornelius Emde, Virginie Do, Zeynep Akata, and Thomas Lukasiewicz. 2021. e-vil: A dataset and benchmark for natural language explanations in vision-language tasks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1244–1254.

Maxime Kayser, Cornelius Emde, Oana-Maria Camburu, Guy Parsons, Bartlomiej Papiez, and Thomas Lukasiewicz. 2022. Explaining chest x-ray pathologies in natural language. *Preprint*, arXiv:2207.04343.

Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual explanations for self-driving vehicles. *Preprint*, arXiv:1807.11546.

Neema Kotonya and Francesca Toni. 2020. Explainable automated fact-checking: A survey. *Preprint*, arXiv:2011.03870.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2021. Data augmentation using pre-trained transformer models. *Preprint*, arXiv:2003.02245.

Qing Li, Qingyi Tao, Shafiq Joty, Jianfei Cai, and Jiebo Luo. 2018. Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions. *Preprint*, arXiv:1803.07464.

Zhuoyan Li, Hangxiao Zhu, Zhuoran Lu, and Ming Yin. 2023. Synthetic data generation with large language models for text classification: Potential and limitations. *Preprint*, arXiv:2310.07849.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. 2023. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. *Preprint*, arXiv:2310.05338.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Preprint*, arXiv:2209.09513.

Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. 2024. Machine learning for synthetic data generation: A review. *Preprint*, arXiv:2302.04062.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. 2022. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *Preprint*, arXiv:2112.10741.

Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal explanations: Justifying decisions and pointing to the evidence. *Preprint*, arXiv:1802.08129.

Maximilian Schmidt, Andrea Bartezzaghi, and Ngoc Thang Vu. 2024. Prompting-based synthetic data generation for few-shot question answering. *Preprint*, arXiv:2405.09335.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Preprint*, arXiv:2403.16999.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Preprint*, arXiv:2004.09297.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Genta Indra Winata, Hanyang Zhao, Anirban Das, Wenpin Tang, David D Yao, Shi-Xiong Zhang, and Sambit Sahu. 2024. Preference tuning with human feedback on language, speech, and vision tasks: A survey. *arXiv preprint arXiv:2409.11564*.

Changrong Xiao, Sean Xin Xu, and Kunpeng Zhang. 2023. Multimodal data augmentation for image captioning using diffusion models. In *Proceedings of the 1st Workshop on Large Generative Models Meet Multimodal Applications*, MM '23. ACM.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018, Lisbon, Portugal. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019a. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019b. From recognition to cognition: Visual commonsense reasoning. *Preprint*, arXiv:1811.10830.

Yue Zhang, Ming Zhang, Haipeng Yuan, Shichun Liu, Yongyao Shi, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Llmeval: A preliminary study on how to evaluate large language models. *Preprint*, arXiv:2312.07398.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *Preprint*, arXiv:2304.10592.

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. 2016. Visual7w: Grounded question answering in images. *Preprint*, arXiv:1511.03416.

A Annotation Metrics

Each of human annotation metric is rated on a scale of 1-3, with 1 being the lowest and 3 the highest quality:

1. *Accuracy*: Measures the precision of the q-a pair in the given context.
2. *Logical*: Evaluates the rationality of e in justifying a for q.
3. *Clarity*: Evaluates the clarity of e in explaining a to q.
4. *Detail*: Ensures e covers all necessary details for a in the context of q.
5. *Relevancy*: Ensures e covers only the necessary details for a in the context of q.

B Triplet Generation Pipeline Pseudocode

We outline the pseudocodes of our triplet generation pipeline of all methods. We denote the following variables:

- $\mathcal{T}(q, a, e)$: generated triplet
- $\bar{D}$: image dataset with size $|\bar{D}|$, each $\bar{D}_i$ is used to generate $|\mathcal{T}|$ triplet(s)
- M or $\bar{M}$: model (LVLM)
- P_{X}: prompt to craft X
- Φ: prompt formatting function
- Θ: image annotating function

- Ψ: ensembling function
- f: inference logic, as described in formula 1
- $\mathcal{Q}$: question prefix pool with length $|\bar{D}| \times |\mathcal{T}|$
- $\mathcal{O}$: object name pool with length $|\bar{D}| \times |\mathcal{T}|$
- $\mathcal{B}$: bounding box pool with length $|\bar{D}| \times |\mathcal{T}|$
- $\hat{x}$: set of x

Algorithm 1 SINGLE-STEP

```
Require: $\bar{D}, M, P_{\mathcal{T}}, \mathcal{Q}, f, \Phi$
  $\hat{\mathcal{T}} \leftarrow \{\}$
  for $i \leftarrow 1$ to $|\bar{D}|$ do
      for $j \leftarrow 1$ to $|\mathcal{T}|$ do
          $\mathcal{T} \leftarrow f(\bar{D}_i, M, \Phi(P_{\mathcal{T}}, \mathcal{Q}_{i+j}))$
          $\hat{\mathcal{T}} \leftarrow \hat{\mathcal{T}} \cup \mathcal{T}$
      end for
  end for
```

Algorithm 2 SINGLE-STEP-VIP

```
Require: $\bar{D}, \bar{M}, P_{\mathcal{T}}, \mathcal{Q}, \mathcal{O}, \mathcal{B}, f, \Phi, \Theta$
  $\hat{\mathcal{T}} \leftarrow \{\}$
  for $i \leftarrow 1$ to $|\bar{D}|$ do
      for $j \leftarrow 1$ to$|\mathcal{T}|$ do
          $\mathcal{T} \leftarrow f(\Theta(\bar{D}_i, \mathcal{B}_{i+j}), \bar{M},$
                  $\Phi(P_q, \mathcal{Q}_{i+j}, \mathcal{O}_{i+j}))$
          $\hat{\mathcal{T}} \leftarrow \hat{\mathcal{T}} \cup \mathcal{T}$
      end for
  end for
```

C Prompts

We utilize 4 distinct prompt templates for each setting. For SINGLE-STEP and SINGLE-STEP-VIP, we make slight modifications to the prompts to accommodate the use of object names and bounding box information, as illustrated in Tables 7 and 8. In the case of MULTI-STEP, we divide the prompt templates into QA and Explanation prompts, as shown in Tables 9 and 10.

D Hyper-parameters and Experimental Settings

We utilize a single A100 40GB GPU for our data generation process. For all generation tasks, we use fp16 and employ the following hyper-parameters:

- `temperature: 1.0`
- `top_p: 1.0`

Algorithm 3 MULTI-STEP

```
Require: $\bar{D}, M, P_q, P_a, \hat{P_{r_*}}, \mathcal{Q}, f, \Phi, \Psi$
  $\hat{\mathcal{T}} \leftarrow \{\}$
  for $i \leftarrow 1$ to $|\bar{D}|$ do
      for $j \leftarrow 1$ to $|\mathcal{T}|$ do
          $q \leftarrow f(\bar{D}_i, M, \Phi(P_q, \mathcal{Q}_{i+j}))$
          $a \leftarrow f(\bar{D}_i, M, \Phi(P_a, q))$

          $\hat{r} \leftarrow \{\}$
          for $P_r$ in $\{P_{r_1}, P_{r_2}, \dots\}$ do
              $r \leftarrow f(\bar{D}_i, M, \Phi(P_r, q, a))$
              $\hat{r} \leftarrow \hat{r} \cup r$
          end for

          $\mathcal{T} \leftarrow (q, a, \Psi(\hat{r}))$
          $\hat{\mathcal{T}} \leftarrow \hat{\mathcal{T}} \cup \mathcal{T}$
      end for
  end for
```

- `top_k: 50`
- `do_sample: False`

When it comes to determining the maximum number of new tokens, we differentiate our approach based on the specific settings:

- For the SINGLE-STEP-* settings, we set the `max_new_tokens` to 1500, providing a comprehensive and detailed output.
- In the MULTI-STEP setting, we allocate 20 tokens for the q parameter and 25 tokens for the a parameter
- For the MULTI-STEP setting with the e parameter, we employ different token lengths depending on the reasoning approach: 70 tokens for Base, 70 tokens for CoT, and 300 tokens for ReAct.

For evaluation engine, we use `deberta-v2-xlarge-mnli` to calculate f1-BERTSCORE and `clip-vit-base-patch16` to calculate CLIPSCORE. We provide modifiable hyper-parameters used within the YAML configuration in Table 12 and 13. Our YAML configuration includes hyper-parameters such as LVLM handle, inference device settings, prompt choice, and run parameters. For SINGLE-STEP-VIP question prefixes, we pre-define the program with a fixed set of prefix choices and a proportion setting of `[2, 2, 2, 1, 1]`. This precaution is taken to prevent users from using prefixes such as "how" and "why," which could hinder the LVLM's ability to generate relevant output. These prefixes may require external knowledge that is not provided in the input prompt, thus potentially diminishing the model's performance.

E Evaluation Rules

We established a standardized evaluation rule, as presented in Table 11, which our annotators followed. These definitions are based on the human evaluation criteria outlined in Section 3.5. Additionally, we included an extra rule to assign a score of `-1` if the annotator identifies any generated data as syntactically or semantically invalid.

F Evaluation Results

F.1 Human Evaluation

We request three English-fluent annotators to assess the quality of our synthetic data. We ensure fair compensation based on the minimum standard hourly rate within their respective region. This evaluation is summarized in Table 5.

F.2 Similarity Evaluation

As mentioned in main section, we employ Jensen-Shannon Divergence and Pearson Correlation to compare each component with human data. Here, we provide the aggregated scores in Table 2 to facilitate a more detailed analysis in the main paper.

G Invalid Triplet Cases

We provide examples of invalid triplet cases that were identified through regular expression disobedience or human evaluation in Table 6. These invalid cases are considered solely for validity evaluation and are excluded from the quality and similarity analyses.

Experiment	Rater	Accuracy	Logic	Clarity	Detail	Relevancy	Avg Score	Gwet*
Base-Small	1	2.56	2.54	2.8	2.54	2.38		
	2	2.69	2.51	2.57	2.47	2.49		
	3	2.43	2.37	2.84	2.55	2.04		
	AVG	**2.56**	**2.473**	**2.737**	**2.52**	**2.303**	**2.519**	**0.64**
Base-Medium	1	2.72	2.56	2.86	2.56	2.46		
	2	2.78	2.4	2.88	2.28	2.88		
	3	2.64	2.56	2.84	2.48	2.38		
	AVG	**2.713**	**2.507**	**2.86**	**2.44**	**2.573**	**2.619**	**0.695**
Multi-Step	1	2.6	2.5	2.68	2.76	2.24		
	2	2.9	2.81	2.88	2.77	2.88		
	3	2.58	2.6	2.71	2.51	1.93		
	AVG	**2.693**	**2.637**	**2.757**	**2.68**	**2.35**	**2.623**	**0.669**
Base+Viz	1	2.57	2.51	2.92	2.57	2.65		
	2	2.55	2.49	2.92	2.55	2.65		
	3	2.58	2.58	2.9	2.65	2.6		
	AVG	**2.567**	**2.527**	**2.913**	**2.59**	**2.633**	**2.646**	**0.801**
human		**2.82**	**2.79**	**2.86**	**2.93**	**2.88**	**2.86**	

Table 5: Qualitative evaluation results breakdown from three annotators. The criteria for assessment adhere to the definitions outlined in Section 3.5, and each score is accompanied by Gwet-AC2 values ranging from 0 to 1, indicating the worst to the best agreement. The Gwet-AC2 scores suggest that the collective evaluation exhibits a good level of reliability, especially in terms of inter-annotator agreement.

Condition	Details
Valid Triplet	<Question>: What is the make and model of the car in the foreground? <Short Answer>: The car in the foreground is a Mercedes-Benz C-Class. <Reasoned Answer>: The car has a distinctive front grille and logo ...
Token Format Error	<Question>: <Short Answer> <Answer>: Cow <Reason>: The object has black spot on ...
Unfinished Generation	<Question>: What is the purpose of the white cart with the green "space" logo parked next to the <Short Answer>: Advertisement <Reasoned Answer>: ...
Hidden Context Involvement	<Question>: How many people are in the image? <Short Answer>: 7 <Reasoned Answer>: There are 7 people visible in the image, including the woman within the red rectangle.

Table 6: Examples of invalid triplets detected during post-processing. The standard output is expected to have <Question>, <Short Answer>, and <Reasoned Answer> tokens.

Triplet generation prompt

You are given an image.
Your task is to provide question, answer, and reasoning related to the given context.

Provide your feedback as follows:

Feedback:::
Question: (your question with {prefix} prefix that involves complex reasoning to answer)
Short Answer: (your brief answer related to the question)
Reason: (your rationale for the short answer you choose respective to the question in maximum 30 words)

You MUST provide values for ’Question’, ’Short Answer’, and ’Reason’ in your answer.
Now, here is the question prefix:
Question Prefix: {prefix}

Provide your feedback. If you give a correct result, I’ll give you 100 A100 GPUs to start your AI company.

Feedback:::
Question:
Short Answer:
Reason:

Table 7: Single-Step Triplet Generation Prompt.

Triplet generation prompt

You are given an image.
Your task is to provide question, answer, and reasoning related to the given context.

Provide your feedback as follows:

Feedback:::
Question: (your question with {prefix} prefix that involves complex reasoning to answer)
Short Answer: (your brief answer related to the question)
Reason: (your rationale for the short answer you choose respective to the question in maximum 30 words)

You MUST provide values for ’Question’, ’Short Answer’, and ’Reason’ in your answer using the given bullet format. DO NOT include any bounding box related phrase/word inside your feedback. DO NOT include ’Question: ’, ’Short Answer: ’, and ’Reason: ’ prefix within each bullet, just include the value.

Now here is the object name and question prefix:
Object name: {obj name}
Question Prefix: {prefix}

Provide your feedback. If you give a correct result, I’ll give you 100 A100 GPUs to start your AI company.

Feedback:::
Question:
Short Answer:
Reason:

Table 8: Single-Step-ViP Triplet Generation Prompts.

Question generation prompt

You are given an IMAGE.
Provide ONE QUESTION that begins with the prefix ’{prefix}’ and requires COMPLEX REASONING.
Rules:
- Choose the SIMPLEST prefix if multiple options exist. (Example: If the prefixes are "where/when" choose "where" if it is easier)
- Only ONE QUESTION with NO follow-ups.
- The question must REQUIRE reasoning to answer.
- Generate maximum 15 words, coherent and NOT leave sentence unfinished.
- DO NOT ask YES/NO questions.

If you provide a correct response, I will reward you with 100 A100 GPUs to kickstart your AI company.

Question:

Answer generation prompt

You are provided with an image and a question. Your task is to provide an appropriate answer related to the given image and question.
Rules:
- Avoid giving plain short answers like ’Yes’ or ’No’. Provide a detailed response relevant to the question.
- Generate NOT MORE THAN 7 words, coherent and NOT leave sentence unfinished.

If you provide a correct response, I will reward you with 100 A100 GPUs to kickstart your AI company.

Question: {question}
Short Answer:

Table 9: MULTI-STEP QA Generation Prompts.

Single-Step Explanation Prompt

You are provided with an IMAGE, a QUESTION, and a SHORT ANSWER.
Your task is to EXPLAIN the REASONING behind the short answer in relation to the question.

Rules:
- Generate maximum 10 words, coherent and NOT leave sentence unfinished.
- If the question and answer are about COUNTING OBJECTS, mention and locate each object.
- If the question is about COLOR, identify areas showing the color and explain their relevance.

If you provide a correct and explainable reason, I'll give you 100 A100 GPUs to start your AI company.

Question: {question}
Short Answer: {short_answer}
Reasoning:

CoT Explanation Prompt

You are provided with an IMAGE, a QUESTION, and a SHORT ANSWER.
Your task is to EXPLAIN the REASONING behind the short answer in relation to the question by detailing your thought process step-by-step in PARAGRAPH.

Rules:
- Your reasoning must be in MAXIMUM 30 WORDS.
- Break down the reasoning into clear, logical steps.
- DO NOT PROVIDE LIST, PROVIDE PARAGRAPH.

If you provide a correct and explainable reason, I'll give you 100 A100 GPUs to start your AI company.

Question: {question}
Short Answer: {short_answer}
Reasoning: Let's think step by step.

ReAct Explanation Prompt

You are provided with an IMAGE, a QUESTION, and a SHORT ANSWER.
Your task is to EXPLAIN the REASONING behind the short answer in using observation, thoughts, and action until you are sure you reach your final reason answer.

Rules:::
- Observation: Carefully examine the IMAGE to identify relevant details and elements related to the QUESTION.
- Thoughts: Analyze the observed details to understand their significance and how they relate to the QUESTION and SHORT ANSWER.
- Action: Based on your observations and thoughts, formulate reasoning that logically connects the elements to the SHORT ANSWER.
- Reason: The conclusion from Observation, Thought, and Action in NOT LESS THAN 10 words and NOT MORE THAN 30 words.

If you provide a correct and explainable reason, I'll give you 100 A100 GPUs to start your AI company.

Question: {question}
Short Answer: {short_answer}
Observation:
Thoughts:
Action:
Reason:

Table 10: Multi-Step Explanation Generation Prompts.

Accuracy	
1 (Disagree)	QUESTION & SHORT ANSWER are not at all aligned with the context in the image (e.g., asking about something not present, too assumptive, or not related).
2 (Neutral)	QUESTION is valid, but the ANSWER is less accurate and does not fully match the context of the image.
3 (Agree)	QUESTION is valid and SHORT ANSWER is accurate according to the context in the image, and appropriately addresses the question.
Logic	
1 (Disagree)	EXPLANATION provides an explanation that is incorrect or contains elements that are unreasonable or not aligned with the context in the image.
2 (Neutral)	EXPLANATION provides an explanation that is somewhat accurate or logical, but there are some misalignments or gaps with the context in the image.
3 (Agree)	EXPLANATION provides an explanation that is fully logical, clear, and entirely aligned with the context in the image, supporting the choice effectively.
Clarity	
1 (Disagree)	EXPLANATION provides an explanation that is not easy to understand, is convoluted, or poorly structured, making it difficult to follow.
2 (Neutral)	EXPLANATION provides an explanation that is somewhat understandable but contains complexity or unnecessary details that make it less clear.
3 (Agree)	EXPLANATION provides an explanation that is clear, straightforward, and easy to understand, presenting the information in a logical and concise manner.
Detail	
1 (Disagree)	EXPLANATION only repeats the short answer or lacks sufficient detail to explain the justification for choosing the short answer, making it incomplete.
2 (Neutral)	EXPLANATION contains some detail but does not cover the full explanation needed for justifying the choice of the short answer, leaving gaps in the reasoning.
3 (Agree)	EXPLANATION contains all necessary detail (or more) required to justify the choice of the short answer, providing a comprehensive and well-supported explanation.
Relevancy	
1 (Disagree)	EXPLANATION contains a lot of irrelevant context that does not pertain to justifying the short answer or the context in the image, leading to confusion.
2 (Neutral)	EXPLANATION contains some irrelevant context that does not fully pertain to justifying the short answer or the context in the image, but is mostly relevant.
3 (Agree)	EXPLANATION does not contain irrelevant context and is directly relevant to justifying the short answer based on the context in the image, staying on topic.

Table 11: Evaluation score guide for human evaluation based on Section 3.5.

Setting	Hyper-parameters
SINGLE-STEP-7B	test_name: SINGLE-STEP-7B seed: 42 dataset: name: GQA count: 167 use_scene_graph: 0 model: name: llava-hf/llava-1.5-7b-hf path: llava-hf/llava-1.5-7b-hf family: llava params: use_8_bit: 0 device: "cuda" low_cpu: 1 prompt: singlestep-optim run_params: num_per_inference: 3 use_img_ext: 1 q_prefix: ["what", "is/are (pick one that fits the most)", "which", "how many", "where"] q_prefix_prop: [3,2,1,1,1]
SINGLE-STEP-13B	test_name: SINGLE-STEP-13B seed: 42 dataset: name: GQA count: 167 use_scene_graph: 0 model: name: llava-hf/llava-1.5-13b-hf path: llava-hf/llava-1.5-13b-hf family: llava params: use_8_bit: 0 device: "cuda" low_cpu: 1 prompt: singlestep-optim run_params: num_per_inference: 3 use_img_ext: 1 q_prefix: ["what", "is/are (pick one that fits the most)", "which", "how many", "where"] q_prefix_prop: [3,2,1,1,1]

Table 12: Our experiment hyper-parameters in YAML format for SINGLE-STEP.

Setting	Hyper-parameters
SINGLE-STEP-VIP	test_name: SINGLE-STEP-VIP seed: 42 dataset: name: GQA count: 167 use_scene_graph: 1 model: name: llava-hf/vip-llava-13b-hf path: llava-hf/vip-llava-13b-hf family: vip_llava params: use_8_bit: 0 device: "cuda" low_cpu: 1 prompt: nonvis-optim run_params: num_per_inference: 3 use_img_ext: 1 q_prefix: <!hardcoded> q_prefix_prop: <!hardcoded>
MULTI-STEP	test_name: MULTI-STEP seed: 42 dataset: name: GQA count: 167 use_scene_graph: 0 model: name: llava-hf/llava-1.5-13b-hf path: llava-hf/llava-1.5-13b-hf family: llava params: use_8_bit: 0 device: "cuda" low_cpu: 1 prompt: self_consistency run_params: num_per_inference: 3 use_img_ext: 1 q_prefix: ["what", "is/are (pick one that fits the most)", "which", "how many", "where"] q_prefix_prop: [3,2,1,1,1]

Table 13: Our experiment hyper-parameters in YAML format for SINGLE-STEP-VIP and MULTI-STEP.