World to Code: Multi-modal Data Generation via Self-Instructed Compositional Captioning and Filtering

Anonymous ACL submission

Abstract

Recent advances in Vision-Language Models (VLMs) and the scarcity of high-quality multi-modal alignment data have inspired numerous researches on synthetic VLM data generation. Challenging the conventional norm in VLM data construction, which uses a mixture of specialists in caption and OCR, or stronger VLM APIs and expensive human annotation, we propose to leverage the VLM itself for extracting cross-modal information of each via different prompts and filter the generated outputs again by itself via a consistency filtering strategy. In this paper, we present World to Code (*W2C*), a meticulously curated multi-modal data construction pipeline that organizes the final generation output into a Python code format. Experiments have demonstrated the high quality of *W2C* by improving various existing visual question answering and visual grounding benchmarks across different VLMs. Further analysis also demonstrates that the new code parsing ability of VLMs presents better cross-modal equivalence than the commonly used detail caption ability. Our code and data will be made public.

1 Introduction

Fueled by the rapid development of Vision-Language Models (VLMs) (Zhu et al., 2023; Liu et al., 2024b; Team et al., 2023; Liu et al., 2024a; Dong et al., 2024b) and Diffusion Models (DMs) (Betker et al., 2023), collecting detailed and concrete high-quality captions for each image becomes more and more urging. However, expensive and tedious human labeling for high-quality image-text pairs further incurs the necessity of a cheap and reliable data construction pipeline without human intervention. Related works on image-text data curation can be divided into two main streams. Distillation-based methods leverage closed-source commercial products (e.g., GPT-4V (Achiam et al., 2023)) with the state-of-the-art performance for image caption (Chen et al., 2023a; Li et al., 2023e; Chen et al., 2024a). Another line of work curates an image caption pipeline with existing VLMs to filter high-quality image-text for the training of better VLMs. These methods usually combine open-source LLMs (Touvron et al., 2023a,b; Chiang et al., 2023) and different visual specialists (Li et al., 2023a; Huang et al., 2023b; Zong et al., 2023; Zhang et al., 2024a; Fang et al., 2023; Minderer et al., 2022; Ren et al., 2024; Zhang et al., 2023b) to endow existing VLMs with new abilities, e.g., pixel grounding in GLaMM (Rasheed et al., 2023). However, the dependency on a mixture of specialists and human feedback in filtering noisy generations (Wang et al., 2023b) makes it difficult to scale the generated data and automate the process. Recent progress shows that generated results of LLMs (Wang et al., 2022; Li et al., 2023c) and VLMs (Zhang et al., 2024b) for prompts with similar meanings should be alike and we can help filter out noisy generated texts and captions by consistency checking. In light of the above evidence, we present a self-instructed data construction pipeline, coined *W2C* , to filter generated image captions via existing VLMs through multiple instructed prompt consistency. The overall pipeline reduces requested specialists and frees off expensive human feedback as shown in Figure 1. In addition, we leverage the idea from human-machine interaction and organize the model-generated responses into a Python code format, following Eureka (Ma et al., 2023) and Text2Reward (Xie et al., 2023a).

Experiments have shown that our proposed *W2C* can improve VLMs on various visual question-answering benchmarks. To be specific, *W2C* performs the best in 7 out of 9 VQA benchmarks on LLaVA-NeXT-7B, and 6 out of 9 VQA benchmarks on LLaVA-NeXT-13B. Furthermore, *W2C* also improves few-shot evaluations on two widely used VQA benchmarks including GQA and MME. Especially, on the 2-shot evaluation of GQA, the

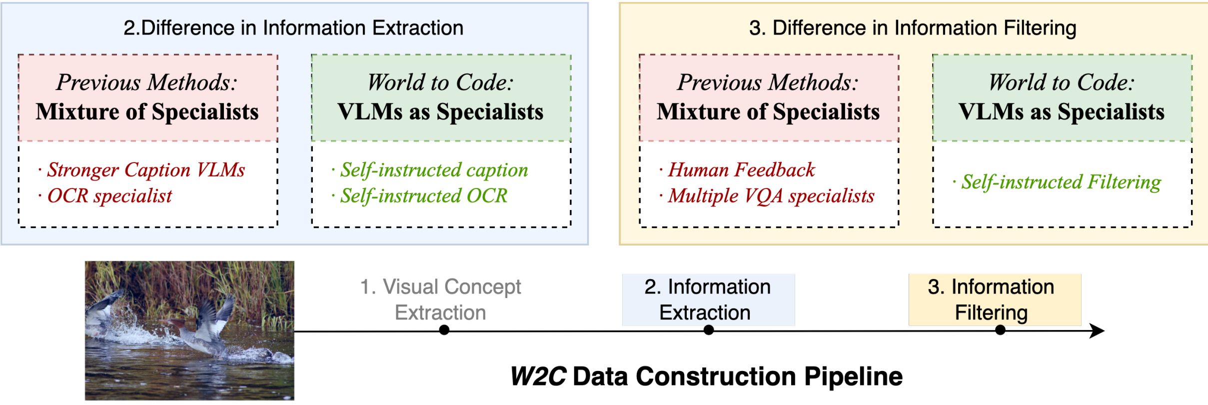

Figure 1: Overview of *W2C* and comparison of existing data construction pipelines. *W2C* differs from existing works by reducing the need for a mixture of specialists and expensive human annotations via self-instruct.

method achieves over 5 accuracy gains across different VLMs.

Our contribution is summarized in threefold:

- We present the data pipeline of *W2C* , which proposes to generate and filter data all by existing VLMs themselves via self-instruct, significantly reducing the need for a mixture of specialists or expensive human annotations in conventional pipelines.
- The generated data of *W2C* presents comparable better performance on classical VQA benchmarks and consistently better performance on visual grounding benchmarks than ShareGPT4V.
- Further analysis presents that the new code parsing ability displays better cross-modality equivalence than the commonly used detail caption ability in presenting the details of an image.

2 Related Work

Vision Language Models With the emergence of LLMs (OpenAI, 2023; Achiam et al., 2023; Touvron et al., 2023a; Team et al., 2023; Jiang et al., 2024), VLMs (Zhu et al., 2023; Zhang et al., 2023a; Team et al., 2023) have demonstrated exceptional capabilities in visual recognition and understanding, achieving remarkable results on various VLM benchmarks (Singh et al., 2019; Tito et al., 2021; Zhang et al., 2024b; Liu et al., 2023b; Ying et al., 2024; Fu et al., 2024). The seminal BLIP2 (Li et al., 2023a) firstly introduces Q-Former to adapt encoded image features as potential language tokens for LLM-based caption prediction. Following works (Liu et al., 2024a; Team et al., 2023; Dong et al., 2024c) improve the visual component by replacing VIT (Dosovitskiy et al., 2020) or scaling the input image resolution, while Zhu et al. (Zhu et al., 2023) extends BLIP2 by employing emergent open-source LLMs (Touvron et al., 2023a; Chiang et al., 2023), endowing current VLMs with significantly better instruction following and problem solving abilities. LLaVA/LLaVA-1.5 (Liu et al., 2024b, 2023a) further remove Q-Former and point out that simple MLP projection layers present impressive performance in aligning image representation with LLMs. Some works also highlight the importance of collecting high-quality cross-modal alignment data for improving the consistently scaling VLMs (Bai et al., 2023; Wang et al., 2023b; Li et al., 2023b).

Multi-modal Dataset Construction The scarcity of high-quality human-labeled data inspires the synthesis of cross-modal data (Wang et al., 2024; Chen et al., 2023a; Rasheed et al., 2023; Wang et al., 2023a; Li et al., 2023e; Lu et al., 2023; Dong et al., 2024a; Chen et al., 2024c). Among them, Wang et al. (2023b) propose the AS-1B data generation pipeline and open-sourced high-quality dense captions on 1B images. GLaMM (Rasheed et al., 2023) further extends AS-1B by introducing about 10 specialists of different functionalities including grounding, tagging, and in-context learning. These specialists enable pixel-wise grounded dense captions for each image. However, the expensive human annotation required in AS-1B and the complicated construction pipeline in GLaMM have greatly limited the potential of data scaling. In this work, we try to answer whether synthetic data can improve VLMs on classical VQA benchmarks (Fu et al., 2024; Ying et al., 2024; Chen et al., 2024b) to avoid tedious data collection.

Recent progress in synthetic data generation for LLMs (Huang et al., 2023a; Li et al., 2023c; Wang et al., 2022, 2023c) shed light on the possibility of Multi-modal data construction by leveraging consistency in generation to filter invalid data. Wang et al. (2022) presents the consistent reasoning path generation demonstrating better performance in COT. Li et al. (2023c) uses the generator-validator consistent data for training and can effectively improve LLMs on various tasks. Zhang et al. (2024b) further shows that the generator-validator consistency in most VLMs is prone to be correct.

Code Representation for Visual Tasks Code representations can formally encode various structure information in a scene. Eureka (Ma et al., 2023) and Text2Reward (Xie et al., 2023a) parse a scene into Python codes and encourage LLMs to generate programmable dense rewards. ViStruct (Chen et al., 2023b) takes the first step in visual code intelligence by decomposing the code-visual representation into multiple components including object recognition, object grounding, attribute detection, relation detection, and event detection. Chen et al. (2023b) further introduces a curriculum learning approach to endow VLMs with the aforementioned four abilities. However, the heavy dependency on supervised human-labeled datasets and the complicated curriculum learning pipeline limits its potential. This work investigates an effective data-constructing pipeline based on code-vision representation.

3 Method

Our data construction pipeline shares some similarities with GLaMM (Rasheed et al., 2023), where both methods focus on the region-level caption of the whole image. *W2C* further extend GLaMM to support generation-validation consistency filtering by exploring different organization formations of the labeled elements and present how VLMs boost themselves on basic multi-modal understanding tasks.

To make a comprehensive and systematic exposition of our *W2C* entire pipeline, the following will be divided into three parts for discussion:

(1) Visual Concepts Extraction in Section 3.1, (2) Self-Instructed Information Extraction in Section 3.2, (3) Information Filtering via Self Consistency in Section 3.3, (4) Structured formatting in Section 3.4. The overview of our construction pipeline is shown in Figure 2 and all the used instruct prompts are shown in Appendix A.1.

3.1 Visual Concepts Extraction

To build a fully covered concept list for each image I in images dataset D_{raw}, we prompt VLMs to generate both general captions (for a concise overview of the image) and detail captions (to bootstrap as many visual concepts as possible in caption) using specific instruct prompts, p_g and p_d. We use beam search to encourage the VLMs to provide as many visual concepts as possible to improve the generation diversity. The captions obtained as follows:

$$o_g, o_d = f_{\text{VLM}}(I, p_g), f_{\text{VLM}}(I, p_d) \quad (1)$$

where o_g, o_d denote the general captions and detail captions. Since the visual concepts are mainly composed of noun phrases, we employ the NLTK toolkit (Bird, 2006) to extract all noun phrases denoted as $N = \{N_1, N_2, ..., N_k\}$ from o_g and o_d. This process can be represented as $\mathbf{N} = NLTK(o_g, o_d)$.

We use Grounding DINO to map the extracted noun phrases to the bounding box areas of the current image, where part of the false positive noun phrases are filtered as they fail to be mapped with corresponding areas in the image. Here we denote the filtered visual concepts as $\mathbf{C} = \{c_1, c_2, ..., c_k\}$, and their corresponding bounding boxes as $\mathbf{B} = \{b_1, b_2, ..., b_k\}$, which is formulated as follows:

$$\mathbf{B}, \mathbf{C} = f_{\text{DINO}}(I, \mathbf{N}) \quad (2)$$

3.2 Self-Instructed Information Extraction

Region-level Captions We crop image I for each visual concept c_i with its corresponding bounding box b_i to obtain detailed caption and prompt the VLMs to provide a general caption centered on c_i. Additionally, to encourage the VLMs for providing more concrete details about the properties of c_i, we instruct the VLMs to include the color and material of c_i in the caption. Denote the description prompt for region-level caption as $p_{\text{desc}}(c_i)$ and the image cropped by b_i as $I(b_i)$. The region-level caption for each visual concept c_i is formulated as:

$$o_{\text{desc}}(c_i) = f_{\text{VLM}}(p_{\text{desc}}(c_i), I(b_i)) \quad (3)$$

OCR information Unlike previous methods that mainly use OCR tools (PaddleOCR, 2023) to enhance the OCR capabilities, *W2C* acquire the OCR

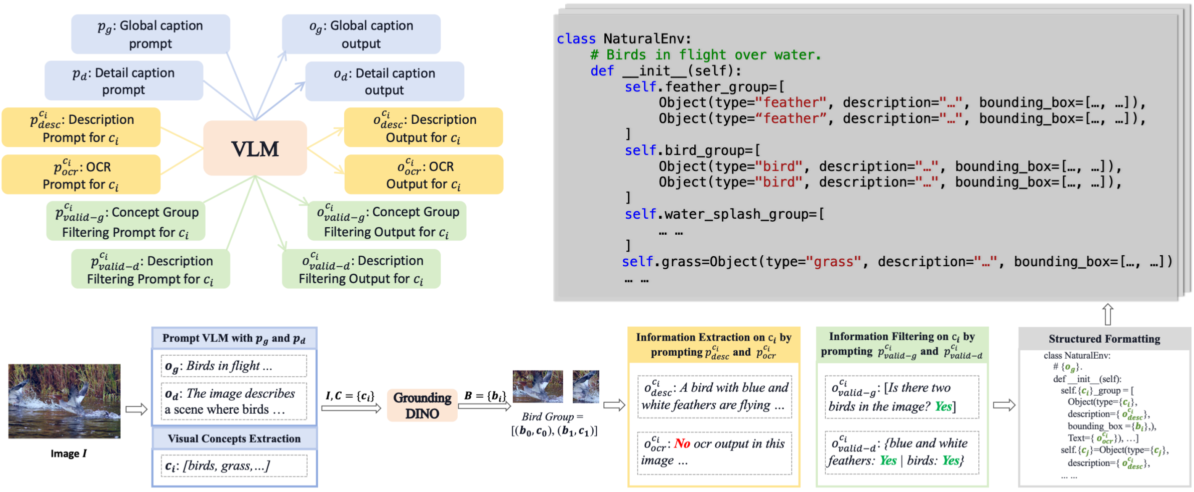

Figure 2: The data construction pipeline for *W2C* . Our pipeline utilizes both VLM and an object detector model to furnish structured data with region-specific awareness, detailed entity captions, and comprehensive global information. The VLM is iteratively invoked to generate the caption and perform consistency filtering to obtain high-quality data. The visual concepts set is obtained from the captions by the NLTK toolkit, c_i here represents a visual concept from the set. The instruction prompts are all predefined templates.

information via instructed prompt to guide VLMs for existing VLMs have better capability in reading text in complex natural scenarios. Given the OCR instruct prompt $p_{ocr}(b_i)$, the OCR information in each bounding box area b_i is formulated as follows:

$$o_{\text{ocr}}(b_i) = f_{\text{VLM}}(p_{\text{ocr}}(b_i), I(b_i)) \tag{4}$$

3.3 Information Filtering via Self Consistency

Our consistency filtering strategy is inspired by the similar generator-validator consistency findings in ConBench (Zhang et al., 2024b), where different instruct prompts may lead to in-consistent captions of visual concepts, and the highly consistent generations are prone to be correct ones. In this paper, we propose to filter the visual concepts via generation-validation consistency, where we change the region-level captions into multiple visual question answering problems for both counting filtering and caption reranking.

Counting Filtering via Consistency Different from AS-1B, we introduce Grounding DINO in our construction process, which can naturally filter part of the plausible visual concepts as these concepts usually fail to find corresponding bounding boxes in the image. However, Grounding DINO introduces new challenges for counting problems, as visual concepts c_i might be mapped to multiple boxes that have a large overlap due to inappropriately designed hyper-parameters. To prevent the effect by plausibly mapped (b_i, c_i), we group all the c_i that has the same name into $\tilde{\mathbf{C}} = \{\tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_i, ..., \tilde{c}_t\}$, and calculate the existing times for each $\tilde{c}_i$ as $\{n_1, n_2, ..., n_i, ..., n_t\}$. We then merge all the boxes for each $\tilde{c}_i$ (which might contain multiple visual concepts with the same name) into $\tilde{\mathbf{B}} = \{\tilde{b}_1, \tilde{b}_2, ..., \tilde{b}_i, ..., \tilde{b}_t\}$, for a box $\tilde{b}_i$ we crop the image and prompt the VLMs to check whether the group element $\tilde{c}_i$ exist n_i times in the image via instruct prompt $p^{\tilde{c}_i}_{\text{valid-g}}$:

$$o_{\text{valid-g}}(\tilde{c}_i) = f_{\text{VLM}}(p_{\text{valid-g}}(\tilde{c}_i), I(\tilde{b}_i)) \tag{5}$$

Caption Re-ranking via Consistency To provide better region-level captions for a given visual concept, we use beam search to bootstrap multiple caption candidates. To select the best candidate, we again leverage the generator-validator consistency. Specifically, denote the beam size as b, for the given visual concept c_i, we get a list of caption candidate $[o^1_{\text{desc}}(c_i), o^2_{\text{desc}}(c_i), ..., o^b_{\text{desc}}(c_i)]$. We use NLTK to parse these captions and collect all the visual concepts that are contained in these captions. Taking n as the total number of extracted concepts in the captions of c_i, we get a new visual concept list denoted as $[c^1_i, c^2_i, ..., c^k_i, ..., c^t_i]$.

Following Equation 5, we prompt VLMs to check the existence of each extracted visual concept c^k_i via instruct prompt $p_{\text{valid-c}}(c^k_i)$:

$$o_{\text{valid-c}}(c^k_i) = f_{\text{VLM}}(p_{\text{valid-c}}(c^k_i), I(\tilde{b}_i)) \tag{6}$$

We then manually design a scoring mechanism based on the validation result $o_{\text{valid-c}}(c^k_i)$. Specif-

Algorithm 1 Data Construction and Consistency Filtering Pipeline

Input: Image I from dataset D_{raw}, Instruct Prompts: P_{g}, P_{d}, P_{desc}, P_{ocr}, $P_{\text{valid-g}}$, $P_{\text{valid-c}}$, VLM f_{VLM}, Grounding DINO f_{DINO}.

1: Caption generate.
$o_g, o_d = f_{\text{VLM}}(I, p_g), f_{\text{VLM}}(I, p_d)$

2: Visual Concepts Extraction.
$\mathbf{N} = NLTK(o_g, o_d), \quad \mathbf{B}, \mathbf{C} = f_{\text{DINO}}(I, \mathbf{N})$

3: Compositional Captions.(c_i from $\mathbf{C}$,b_i from $\mathbf{B}$)
$o_{\text{desc}}(c_i) = f_{\text{VLM}}(p_{\text{desc}}(c_i), I(b_i))$

4: OCR information Extraction.
$o_{\text{ocr}}(b_i) = f_{\text{VLM}}(p_{\text{ocr}}(b_i), I(b_i))$

5: Grouping Concepts in $\mathbf{C}$ and $\mathbf{B}$.
$\tilde{\mathbf{C}} = \{\tilde{c}_1, \tilde{c}_2, ..., \tilde{c}_i, ..., \tilde{c}_t\}$
$\tilde{\mathbf{B}} = \{\tilde{b}_1, \tilde{b}_2, ..., \tilde{b}_i, ..., \tilde{b}_t\}$

6: Counting Filtering via Consistency.
$o_{\text{valid-g}}(\tilde{c}_i) = f_{\text{VLM}}(p_{\text{valid-g}}(\tilde{c}_i), I(\tilde{b}_i))$

7: Caption Re-ranking via Consistency.
$o_{\text{valid-c}}(c_i^k) = f_{\text{VLM}}(p_{\text{valid-c}}(c_i^k), I(\tilde{b}_i))$

8: Rule-based Structured Formatting and Counting Filtering to get D_{W2C}.

Output: *W2C* dataset D_{W2C}

ically, for each caption that contains multiple extracted visual concepts, we assign each correct visual concept $o_{\text{valid-c}}(c_i^k)$ = "Yes" to score 1 and each hallucinated visual concept $o_{\text{valid-c}}(c_i^k)$ = "No" to -1. By accumulating the scores in each caption, we select the caption with the highest score in one beam as the final caption for the given visual concept c_i, which is supposed to be the most diverse and correct caption.

3.4 Structured Formatting and Filtering

As shown in Figure 2, we organize the structured information into code format to fully represent the region-level information of an image. Inspired by Eureka (Ma et al., 2023) and Text2Reward (Xie et al., 2023b), we organize the information as a structured representation into the Python format due to its generality and conciseness. The organization is achieved by the following three rules.

- One general caption o_g of the whole image as the comments of each image Class.
- Each visual concept is an attribute for the image class. For each visual concept c_i, we get their corresponding bounding box b_i and their caption $o^{c_i}_{\text{desc}}$ and $o^{c_i}_{\text{ocr}}$. Such visual concept is then organized into one single attribute: $\{\text{caption}:o^{c_i}_{\text{desc}}, \text{text}:o^{c_i}_{\text{ocr}}, \text{bbox}:b_i\}$.
- Grouping visual concepts with the same name. To make the representation code more concise, we group the visual concepts with the same name in a list $\tilde{c}_i\prime = [c_i^1, c_i^2, ...]$.

By integrating these rules, we get the final code representation of each image, which is then followed by the rule-based filtering strategy that filters out counting in-consistent samples.

In conclusion, by denoting the final dataset as D_{W2C}, the whole data construction pipeline is depicted in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets For the data construction pipeline, we strictly use the images in the ShareGPT4V dataset for our self-instructed approach validation in a fair comparison. Since the original ShareGPT4V dataset contains duplicate images, We remove the repeated images in the original 102K data and get about 87K original images. We follow the practice of LLaVA-1.5 (Liu et al., 2023a) to adopt a two-stage training approach consisting of prompt tuning (PT) and instruct tuning (IT). For the experiments on low resolution setting, we follow the LLaVA-1.5 to use training dataset $\text{LLaVA}_{558\text{k}}$ for PT stage and $\text{LLaVA}_{665\text{k}}$ for IT stage on LLaVA-1.5 training stages. As the specific mixture ratio details of the LLaVA-NeXT data were omitted, we directly utilized the entire training set from each of the following datasets in the IT stage, forming a mixture of datasets including: $\text{LLaVA}_{665\text{k}}$ (Liu et al., 2023a), DocVQA (Tito et al., 2021), ChartQA (Masry et al., 2022) and ShareGPT4V (Chen et al., 2023a) on high resolution setting.

To comprehensively assess the effectiveness of our constructed dataset, we evaluate the model on widely adopted multi-modal benchmarks and grouding benchmarks, including TextVQA (Singh et al., 2019) (without providing OCR tokens), DocVQA (Tito et al., 2021), ChartQA (Masry et al., 2022), MME (Fu et al., 2024), MMT Bench (Ying et al., 2024), MMStar (Chen et al., 2024b), ScienceQA (Lu et al., 2022), POPE (Li et al., 2023d), GQA (Hudson and Manning, 2019), RefCOCO (Kazemzadeh et al., 2014), RefCOCO+ (Mao et al., 2016) and RefCOCOg (Mao et al., 2016). These benchmarks provide a comprehensive assessment of multiple perspectives on multi-modal VLM performance.

Implementation Details In this paper, we employ two types of leading methods: LLaVA-

Method	GQA	MME.	POPE	SQA^I	MMS.	MMT.	Text.	Doc.	Chart.
				Low resolution setting					
LLaVA-1.5-7B*	62.3	1468	**86.2**	68.2	32.4	48.6	47.6	-	-
+ShareGPT4V	**63.4**	**1507**	86.0	69.0	**34.3**	49.3	**47.9**	-	-
+*W2C*	62.8	1503	85.6	**69.8**	33.5	**49.4**	46.6	-	-
LLaVA-1.5-13B*	63.7	**1574**	85.7	72.1	33.5	51.1	**49.0**	-	-
+ShareGPT4V	**64.0**	1537	**86.1**	72.0	33.9	50.9	48.8	-	-
+*W2C*	**64.0**	1547	85.7	**72.6**	**36.1**	**51.7**	48.9	-	-
				High resolution setting					
LLaVA-NeXT-7B	**64.2**	1473	87.3	67.9	34.6	48.2	63.9	75.4	62.0
+ShareGPT4V*	64.0	1513	85.8	**68.5**	33.7	49.5	**64.2**	75.1	62.2
+*W2C*	**64.2**	**1516**	**87.5**	68.3	**35.8**	**50.1**	63.7	**76.5**	**63.0**
LLaVA-NeXT-13B	65.3	1545	87.1	70.1	37.2	50.6	**67.6**	78.1	**66.2**
+ShareGPT4V*	65.3	1574	87.1	70.1	**37.5**	50.4	67.0	78.4	63.8
+*W2C*	**65.5**	**1597**	**87.5**	**70.7**	37.1	**51.4**	65.2	**79.1**	65.6

Table 1: Visual Question Answering benchmarks of *W2C* on LLaVA1.5 and LLaVA-NeXT under different combination of IT datasets. The best results are **bold** and the second results are underlined. *: our reproduction of LLaVA-1.5 and LLaVA-Next, which achieves comparable performance with the original papers. −: LLaVA-1.5 does not support benchmarks that requires high input resolution. Abbreviations: SQA^I(ScienceQA), MMS.(MMStar), MMT.(MMT-Bench), Text.(TextVQA), Doc.(DocVQA), Chart.(ChartQA).

1.5 (Liu et al., 2023a) uses a CLIP-pretrained ViT-L/14 (Radford et al., 2021) as a vision encoder, a projector and an LLM, and LLaVA-NeXT (Liu et al., 2024a) increases the input image resolution by applying an adaptive image cropping strategy to concatenate all vision tokens. To ensure a fair and comprehensive comparison Table 1 and Table 2 present results both excluding and including the ShareGPT4V dataset, as well as results from the incorporation of our dataset.

We have reproduced LLaVA-NeXT with a learning rate of ViT to 1/10 of the base learning rate for the reason that LLaVA-NeXT only publishes their evaluation code. The learning rate for the PT stage is set to $1e^{-3}$ and the IT stage is set to $2e^{-5}$ for both Vicuna-7B and Vicuna-13B backbone LLM. We use 16 A100 for experiments on VLM training. We freeze the vision encoder during training on the LLaVA-1.5 and only freeze the vision encoder on the PT stage during training on the LLaVA-NEXT following the original paper. We show more training details in the Appendix B.1

Data Processing Details During the data construction pipeline, we employ NLTK (Bird, 2006) tool to extract noun phrases from the captions, and the resulting set of phrases is then post-processed using WordNet (Miller, 1995) to remove duplicates and filter out inaccurately named entities. The total amount of final data after consistency filtering will not be completely consistent for different VLMs and we show the details in Appendix B.1. The checkpoints of the VLM we used in our data processing are the original checkpoints of the official release. For LLaVA-1.5, which is not trained with the ShareGPT4V dataset, LLaVA-NEXT is trained with part of the ShareGPT4V dataset. The detailed GPU hours can be found in Appendix B.2 and we show the visualization of our *W2C* samples in Appendix B.3.

4.2 Main Results

Effectiveness of *W2C* data improve various VLMs in Visual Question Answering benchmarks We show a quantitative comparison results of the trained VLMs with and without the ShareGPT4V dataset, as well as *W2C* for replacement of the ShareGPT4V during the IT training stage in Table 1. *W2C* consistently improves the performance on different settings in both LLaVA-1.5 and LLaVA-NeXT. Especially, in the high resolution setting, our *W2C* presents impressive performance improvement on multi-modal visual understanding benchmarks such as MMT Bench, MMStar, and MME. Specifically, *W2C* can bring improvement in 7 out of 9 benchmarks on LLaVA-NeXT-7B and 6 out of 9 on LLaVA-NeXT-13B. Especially, on LLaVA-NeXT-13B, *W2C* improves DocVQA by 0.7 ANLS, ChartQA by 1.8 accuracy, MMT Bench by 0.8 accuracy and MME by 23 points compared to the reproduction results of LLaVA-NeXT.

Method	RefCOCO			RefCOCO+			RefCOCOg		
	test-a	test-b	val	test-a	test-b	val	test	val	Avg.
Low resolution setting									
LLaVA-1.5-7B	86.8	72.9	80.0	79.3	60.7	70.7	72.2	72.2	74.4
+ShareGPT4V	87.1	72.7	80.4	79.5	62.2	71.5	72.5	72.2	74.8
+*W2C*	**88.0**	**75.3**	**81.7**	**81.5**	**63.1**	**73.9**	**75.2**	**75.2**	**76.3**
LLaVA-1.5-13B	88.9	75.3	82.3	82.4	65.0	74.3	75.2	74.6	77.3
+ShareGPT4V	89.0	75.6	83.0	82.7	65.6	75.7	75.3	75.0	77.7
+*W2C*	**89.6**	**77.6**	**84.1**	**85.0**	**67.2**	**77.3**	**76.8**	**76.8**	**79.3**
High resolution setting									
LLaVA-NeXT-7B	89.9	78.7	84.8	84.5	68.7	77.0	79.4	78.8	80.2
+ShareGPT4V	89.4	76.8	83.5	82.1	65.9	75.5	77.5	77.6	78.5
+*W2C*	**90.9**	**81.3**	**86.4**	**85.8**	**70.5**	**79.5**	**80.7**	**80.5**	**82.0**
LLaVA-NeXT-13B	**91.7**	81.9	86.3	86.2	71.2	79.5	80.9	80.8	82.3
+ShareGPT4V	91.5	80.8	86.5	86.0	71.1	79.6	79.6	79.8	81.9
+*W2C*	91.1	**83.6**	**87.3**	**86.3**	**72.9**	**81.0**	**81.7**	**81.3**	**83.2**

Table 2: Grounding benchmarks of *W2C* on LLaVA1.5 and LLaVA-NeXT under different combination of IT datasets. The best results are **bold** and the second results are underlined.

Method	format	MMT-Bench	DocVQA	TextVQA	$\text{RefCOCO}_{\text{val}}$	$\text{RefCOCO+}_{\text{val}}$	$\text{RefCOCOg}_{\text{val}}$
LLaVA-NeXT-7B	*single*	49.2	75.4	**63.8**	85.4	78.5	79.5
LLaVA-NeXT-7B	*multi*	48.8	72.0	61.4	82.4	73.8	76.8
LLaVA-NeXT-7B	*code*	**50.1**	**76.5**	63.7	**86.4**	**79.5**	**80.5**

Table 3: Ablation study of *W2C* on using different data organization format. *single/multi/code*: constructed data are organized in single-round conversations/multi-round conversations/python code format.

***W2C* data show impressive performance on Grounding benchmarks** We present the performance of the VLMs on Grounding benchmarks in Table 2. The task of referential expression comprehension necessitates that the model accurately identifies and localizes the object described. Our models demonstrate their exceptional capability for detailed image recognition and localization by undergoing evaluation across various referential expression comprehension benchmarks, including RefCOCO, RefCOCO+, and RefCOCOg. Benefit from the entity-enteric generation of local captions and the presence of local bounding box information, our model achieved an average improvement of 1.5/1.6 average IoU on LLaVA-1.5 7B/13B and 3.5/1.3 average IoU on LLaVA-NeXT 7B/13B.

4.3 Ablation Studies

Our results show advantageous performance in Table 1 and Table 2, but our analysis of these results shows the limitations of the base model's OCR capability on LLaVA-1.5. We proceed with further ablation studies on LLaVA-Next-7B for the constraints on resources, which optimally demonstrate the full benefits of our pipeline and consistency filtering in a comprehensive manner.

Organizing data into the python code format presents better performance We discussed in Section 3.2 the strengths of choosing the code format for the representation of structured data. In Table 3, we quantitatively compare our data format with a single-round dialogue format and a multi-round dialogue format. By using the python code as data construction format, we observe improved performance in both visual grounding benchmarks and visual question answer benchmarks on LLaVA-NeXT-7B. Especially, we improved the MMT-Bench by 0.9/1.3 accuracy and DocVQA by 1.1/4.5 ANLS compared to the *single/multi* data format.

Filtering introduces better downstream benchmarks performance We show the ablation of different consistency filtering choices in Table 4. Similarly, the performance of LLaVA-NeXT-7B on the both visual grounding benchmarks and visual question answering benchmarks highlights the effectiveness and necessity of our consistency filtering approaches. When two filtering strategies are combined, we achieve the best performance by improving DocVQA with 1.0 ANLS, TextVQA

Method	*re-ranking*	*counting*	MMT-Bench	DocVQA	TextVQA	RefCOCO$_{val}$	RefCOCO+$_{val}$	RefCOCOg$_{val}$
LLaVA-NeXT-7B			**50.3**	75.5	62.7	**86.6**	79.0	79.7
LLaVA-NeXT-7B		✓	49.4	76.3	63.4	86.1	78.5	80.4
LLaVA-NeXT-7B	✓		49.4	75.3	63.2	86.5	79.2	79.7
LLaVA-NeXT-7B	✓	✓	50.1	**76.5**	**63.7**	86.4	**79.5**	**80.5**

Table 4: Ablation study of *W2C* when combined the different consistency filtering strategy. *re-ranking*: caption re-ranking. *counting*: counting filtering.

Method	GQA		MME	
	2-shot	4-shot	2-shot	4-shot
LLaVA-1.5-7B				
detail caption	34.79	39.67	1136	1098
code parsing	**41.06**	**43.40**	**1139**	**1169**
LLaVA-1.5-13B				
detail caption	34.00	40.87	1192	1170
code parsing	**39.12**	**43.70**	**1199**	**1224**
LLaVA-NeXT-7B				
detail caption	34.89	40.70	**1174**	1105
code parsing	**40.07**	**45.07**	1154	**1189**
LLaVA-NeXT-13B				
detail caption	31.63	40.07	**1193**	1127
code parsing	**39.80**	**42.83**	1151	**1190**

Table 5: Comparison between detail caption and code parsing ability in few-shot evaluations on MME and GQA without referring to the image.

with 1.0 accuracy, RefCOCO+$_{val}$ with 0.5 IOU and RefCOCOg$_{val}$ with 0.8 IOU. We also achieve comparable results on MMT-Bench and RefCOCO$_{val}$ with little performance degradation.

4.4 Code Parsing Ability Evaluation

We further present better cross-modality equivalence between image and text brought by the new code parsing ability. An ideal caption of the image should enable the ability to question without referring to the image. Therefore, we compare the quality of the code output and widely used detail caption output in the ability to handle downstream tasks via in-context learning on the same Large Language Model.

Experimental Setting We conduct experiments on both LLaVA-1.5-7B/13B and LLaVA-NeXT-7B/13B on two widely used Visual Question Answering benchmarks, including GQA and the perception subset of MME. Due to the support of 32k long context and satisfying performance in the open-source community, we use Qwen-1.5-14B (Bai et al., 2023; Team, 2024) as the problem-solving LLM, and prompt it with few shot inputs. Each shot can be represented as a combination of {description, question, answer}. For the detail caption output, we use the models trained with both the original dataset and the ShareGPT4V dataset to improve their detail caption abilities. For the code parsing output, we replace ShareGPT4V with our proposed *W2C* dataset.

The code parsing ability of VLMs presents much better few-shot performance. From Table 5, the code parsing output shows significant improvement when compared with using the detail caption output. On the binary classification task for the visual perception subset of MME, the code parsing ability achieves comparable or better performance in various settings. On the free generation VQA task, GQA, using the code parsing output can bring clear accuracy gain across different model size and architectures. Especially, on the 2-shot evaluation of GQA on LLaVA-NEXT-13B, the code parsing output by model trained with *W2C* achieves 8.2 accuracy improvement compared to baseline, indicating that the code-parsing ability present improved performance in presenting the details of one image.

5 Conclusion

This paper presents *W2C* , an enhanced data construction pipeline that only leverages existing VLMs themselves for detail and compositional captions for an image, which is further organized in Python code format. We present that existing VLMs can improve themselves on the understanding benchmarks in various scenarios, significantly reducing the need for a mix of visual specialists and heavy human annotations. Moreover, additional experiments show that the new code parsing ability of VLMs presents better capability in fully describing the image, with notable improvement in the few-shot evaluation on downstream tasks when the raw images are not provided. Our proposed *W2C* not only enhances the original capabilities on the widely used multi-modal understanding benchmarks but also endows existing VLMs with detailed and executable multi-modal parsing ability.

6 Limitation

Despite the advancements in improved multi-modal understanding benchmarks and new code parsing ability, *W2C* can be further improved in some aspects.

- In this paper, we directly use the ShareGPT4V dataset images for a fair comparison with ShareGPT4V. However, it contains fewer OCR-centric images, limiting the final performance. Further investigation could be taken in studying the performance of *W2C* on more distribution of unlabeled datasets.

- The experiments are mainly conducted on the SOTA open-source VLM structures, i.e., the LLaVA series which use MLP projectors for multi-modal alignment. The effectiveness of *W2C* can be further investigated on other VLM structures.

Given the promising performance of *W2C* on evaluation benchmarks, we would like to explore a more high-quality and diverse data generation pipeline in future investigation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. 2023. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8.

Steven Bird. 2006. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72.

Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. 2024a. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. 2024b. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Lin Chen, Jisong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. 2023a. Sharegpt4v: Improving large multimodal models with better captions. *arXiv preprint arXiv:2311.12793*.

Yangyi Chen, Xingyao Wang, Manling Li, Derek Hoiem, and Heng Ji. 2023b. Vistruct: Visual structural knowledge extraction via curriculum guided code-vision representation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13342–13357.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024c. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. 2024a. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Xilin Wei, Songyang Zhang, Haodong Duan, Maosong Cao, et al. 2024b. Internlm-xcomposer2: Mastering free-form text-image composition and comprehension in vision-language large model. *arXiv preprint arXiv:2401.16420*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. 2024c. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. 2023. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023a. Large language models can self-improve. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Xinyu Huang, Youcai Zhang, Jinyu Ma, Weiwei Tian, Rui Feng, Yuejie Zhang, Yaqian Li, Yandong Guo, and Lei Zhang. 2023b. Tag2text: Guiding vision-language model via image tagging. In *The Twelfth International Conference on Learning Representations*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, et al. 2023b. M3it: A large-scale dataset towards multi-modal multilingual instruction tuning. *arXiv preprint arXiv:2306.04387*.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. 2023c. Benchmarking and improving generator-validator consistency of language models. In *The Twelfth International Conference on Learning Representations*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023d. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2023e. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning. In *NeurIPS 2023 Workshop on Instruction Tuning and Instruction Following*.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024b. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Yuliang Liu, Zhang Li, Hongliang Li, Wenwen Yu, Mingxin Huang, Dezhi Peng, Mingyu Liu, Mingrui Chen, Chunyuan Li, Lianwen Jin, et al. 2023b. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*.

Jianqiao Lu, Wanjun Zhong, Wenyong Huang, Yufei Wang, Fei Mi, Baojun Wang, Weichao Wang, Lifeng Shang, and Qun Liu. 2023. Self: Language-driven self-evolution for large language model. *arXiv preprint arXiv:2310.00533*.

Pan Lu, Swaroop Mishra, Tanglin 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. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Yecheng Jason Ma, William Liang, Guanzhi Wang, De-An Huang, Osbert Bastani, Dinesh Jayaraman, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Eureka: Human-level reward design via coding large language models. In *The Twelfth International Conference on Learning Representations*.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. 2022. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer.

OpenAI. 2023. Chatgpt. https://openai.com/blog/chatgpt/.

PaddleOCR. 2023. Awesome multilingual ocr toolkits based on paddlepaddle. https://github.com/PaddlePaddle/PaddleOCR.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. 2023. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*.

Shuhuai Ren, Aston Zhang, Yi Zhu, Shuai Zhang, Shuai Zheng, Mu Li, Alexander J Smola, and Xu Sun. 2024. Prompt pre-training with twenty-thousand classes for open-vocabulary visual recognition. *Advances in Neural Information Processing Systems*, 36.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Qwen Team. 2024. Introducing qwen1.5.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2021. Document collection visual question answering. In *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pages 778–792. Springer.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Teng Wang, Jinrui Zhang, Junjie Fei, Yixiao Ge, Hao Zheng, Yunlong Tang, Zhe Li, Mingqi Gao, Shanshan Zhao, Ying Shan, et al. 2023a. Caption anything: Interactive image description with diverse multimodal controls. *arXiv preprint arXiv:2305.02677*.

Weiyun Wang, Yiming Ren, Haowen Luo, Tiantong Li, Chenxiang Yan, Zhe Chen, Wenhai Wang, Qingyun Li, Lewei Lu, Xizhou Zhu, et al. 2024. The all-seeing project v2: Towards general relation comprehension of the open world. *arXiv preprint arXiv:2402.19474*.

Weiyun Wang, Min Shi, Qingyun Li, Wenhai Wang, Zhenhang Huang, Linjie Xing, Zhe Chen, Hao Li, Xizhou Zhu, Zhiguo Cao, et al. 2023b. The all-seeing project: Towards panoptic visual recognition and understanding of the open world. In *The Twelfth International Conference on Learning Representations*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023c. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. 2023a. Text2reward: Automated dense reward function generation for reinforcement learning. *arXiv preprint arXiv:2309.11489*.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. 2023b. Text2reward: Automated dense reward function generation for reinforcement learning. *arXiv preprint arXiv:2309.11489*.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, et al. 2024. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi. *arXiv preprint arXiv:2404.16006*.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. 2023a. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*.

Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. 2023b. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*.

Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. 2024a. Recognize anything: A strong image tagging model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1724–1732.

Yuan Zhang, Fei Xiao, Tao Huang, Chun-Kai Fan, Hongyuan Dong, Jiawen Li, Jiacong Wang, Kuan Cheng, Shanghang Zhang, and Haoyuan Guo. 2024b. Unveiling the tapestry of consistency in large vision-language models. *arXiv preprint arXiv:2405.14156*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*.

Zhuofan Zong, Guanglu Song, and Yu Liu. 2023. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758.

A Prompt Templates for *W2C* data construction pipeline

A.1 Prompt Templates

W2C data construction pipeline calls the VLMs repeatedly by using different prompts. We guide the VLMs to accurately answer questions by designing universal prompt templates, thus ensuring better compliance with instruction. All the prompts are shown in Table 6.

B Implementation Details for *W2C* experiments

B.1 Dataset Details

All the creators or original owners of assets used in the paper are credited properly, and the license and terms of use are explicitly mentioned and are respected properly. All datasets we use are from internet open-source datasets under CC-BY licenses and are cited properly.

Data Construction Pipeline Details We incorporate images from the open-source ShareGPT4V dataset, totaling approximately 87K images. For the VLMs in our data construction pipeline, we directly use the official release checkpoints including LLaVA-1.5 and LLaVA-NeXT.

For the cost of our data construction pipeline, we use about 1/1.5 day on 32 A100s GPU for LLaVA-1.5 and about 2/3 days on 48 A100s GPU for LLaVA-NeXT. For the data obtained by *W2C* pipeline, we get 34K from LLaVA-1.5-7B, 33K from LLaVA-1.5-13B, 37K from LLaVA-NeXT-7B, and 29K from LLaVA-NeXT-13B. The reasons for the inconsistency in the amount of data are multifaceted. On the one hand, a minor portion of the data was discarded due to improper handling of anomalous data throughout the processing stage. On the other hand, a significant amount of data was eliminated during the consistency filtering stage owing to inconsistencies detected by the VLMs. Additionally, the generative capabilities of various VLMs vary, and the inherent randomness within VLMs themselves also contributes to these inconsistencies.

Training Details During the training of VLMs, we use different dataset combinations. We utilize the original paper's open-source dataset during both the PT and IT training stages for LLaVA-1.5. In contrast, for the training of LLaVA-NeXT, the lack of disclosure regarding the specific details of the IT stage, we trained using all training set from LLaVA_{665k} (Liu et al., 2023a), DocVQA (Tito et al., 2021), ChartQA (Masry et al., 2022) and ShareGPT4V (Chen et al., 2023a). Furthermore, by aligning our dataset with that of the original study, we achieved comparable experimental results. We use the CLIP-pretrained ViT-L/14 (Radford et al., 2021) as a vision encoder, which input resolution is 336×336. We freeze the vision encoder during training on the LLaVA-1.5 and only freeze the vision encoder on the PT stage during training on the LLaVA-NEXT following the original paper. The experiments of VLM training are all conducted on 16 A100 GPUs.

B.2 Implementation Details of our Pipeline

We employ beam search to fully leverage the powerful language generation capabilities and extensive knowledge base of VLM. This approach enables the generation of an increased number of captions, assisting us in acquiring a broader set of visual concept candidates. Due to the limitation of GPU memory, we set the generation beam to 8 on LLaVA-1.5 and 4 on LLaVA-Next. The learning rate for the PT stage is set to $1e^{-3}$ and the IT stage is set to $2e^{-5}$ for both Vicuna-7B and Vicuna-13B backbone LLM. We set the warmup ratio to 0.03, the PT stage batch size is set to 256 and the IT stage batch size is set to 128. We use model max length 2048 on LLaVA-1.5 and 4096 on LLaVA-Next for its high resolution setting.

B.3 Data Example

In Figure 3 and Figure 4, we present images from the ShareGPT4V dataset alongside the corresponding annotations we constructed by *W2C* . As shown in these images, the annotations generated entirely by the VLMs accurately describe both the global captions and the detailed captions of local entities within specific areas. Additionally, the OCR text is also encapsulated within the corresponding frames. For multiple entities present in the images, a display of group merging is also conducted.


```
class NaturalEnv:
    # The image shows a large building with an American flag on top, advertising a
country music event.
    def __init__(self):
        self.flag_group=[
            Object(type="flag", description="The flag has a star at its center, resembling
the shape of the state of Tennessee, and is flown against a blue sky backdrop.",
                bounding_box=[0.82,0.23,0.93,0.42]),
            Object(type="flag", description="The flag on the pole is a flat striped flag
with a stars and stripe design.",
                bounding_box=[0.16,0.09,0.22,0.31]),
        ]
        self.banner_group=[
            Object(type="banner", description="This is a stylized image of a huge banner.",
                text=Text(text="grand ole opry, the show that made country music famous"),
                bounding_box=[0.28,0.38,0.58,0.64]),
            Object(type="banner", description="A large rectangular banner featuring the
images of a man playing an acoustic guitar and four other individuals performing with
microphones on a red stage setback.",
                bounding_box=[0.02,0.35,0.11,0.74]),
            Object(type="banner", description="The large banner prominently displays the
sign 'OLE OPRY' in the shape of a red circle with a white border and text.",
                bounding_box=[0.36,0.39,0.51,0.64]),
        ]
        self.bush_group=[
            Object(type="bush", description="This small bush is beautifully trimmed and
has purple flowers adorning it.",
                bounding_box=[0.61,0.65,0.99,0.83]),
            Object(type="bush", description="Large green bush next to a white pole.",
                bounding_box=[0.0,0.65,0.3,0.81]),
            Object(type="bush", description="On the concrete walk in the foreground, there
is a green bush that has been trimmed into an interesting, bushy shape.",
                bounding_box=[0.0,0.66,0.36,1.0]),
        ]
        self.opry_house=Object(type="opry_house", description="The Grand Ole Opry house is
a three-sided building with a light brown roof and orange and white odeon-style
marquee.",
            text=Text(text="grand ole opry house, the show that made country music famous,
grand ole opry"),
            bounding_box=[0.0,0.07,0.98,0.82])
        self.tree=Object(type="tree", description="The tree is tall and green, located on
the side of a building next to a flower bed.",
            bounding_box=[0.85,0.49,1.0,0.78])
        self.entrance=Object(type="entrance", description="The entrance to the building
with a dark wooden door and a black awning.",
            bounding_box=[0.36,0.64,0.5,0.83])
        self.country_music_musicians=Object(type="country_music_musicians",
            description="The poster on the wall shows a country music singer in high
contrast red and blue with vibrant white highlights on his attire.",
            text=Text(text="Grand Ole Opry, The Show that Made Country Music Famous"),
bounding_box=[0.28,0.41,0.35,0.63])
```

Figure 3: Visualization of one *W2C* sample with OCR information.

```
class NaturalEnv:
    # A herd of elephants wading through water with people.
    def __init__(self):
        self.elephant_group=[
            Object(type="elephant", description="The brown elephant is wadding into the
water.",
                bounding_box=[0.45,0.42,0.77,1.0]),
            Object(type="elephant", description="The large elephant on the right has a
muddy side and a long trunk.",
                bounding_box=[0.0,0.58,0.23,1.0]),
            Object(type="elephant", description="The elephant is a large, dark brown
mammal wading in a river.",
                bounding_box=[0.15,0.47,0.35,0.89]),
        ]
        self.people_group=[
            Object(type="people", description="Several people wearing green shirts and
khaki pants are walking on the rocky shore.",
                bounding_box=[0.94,0.06,1.0,0.3]),
            Object(type="people", description="A group of seven men in green shirts and
tan shorts standing together in a sandy area with a wooden pole.",
                bounding_box=[0.89,0.0,0.95,0.21]),
        ]
        self.stick=Object(type="stick", description="A long, slender wooden pole with a
curved shape and a shiny, smooth surface.",
            bounding_box=[0.81,0.55,0.88,0.66])
        self.water_flowing=Object(type="water_flowing", description="Rough surface of the
water shows agitated movement as the elephants bathe in the murky stream.",
            bounding_box=[0.0,0.0,1.0,1.0])
        self.trunk=Object(type="trunk", description="The elephant's trunk is long and
curled at the end.",
            bounding_box=[0.63,0.73,0.77,1.0])
        self.onlooker=Object(type="onlooker", description="One onlooker standing on an
elevated rock with greenish-brown sandal, wearing brown cargo shorts.",
            bounding_box=[0.81,0.0,1.0,0.3])
        self.riverbank=Object(type="riverbank", description="This riverbank is sandy and
rocky, with a cliff-like appearance.",
            bounding_box=[0.63,0.11,1.0,0.37])
        self.stone=Object(type="stone", description="A rectangular, weathered limestone
slab by a river.",
            bounding_box=[0.86,0.67,1.0,0.92])
        self.stick=Object(type="stick", description="The brown stick the person is
holding.",
            bounding_box=[0.04,0.24,0.07,0.45])
```

Figure 4: Visualization of one *W2C* sample without OCR information.

Stage	Prompt
Prompt for Caption	
Global Caption $-$ p_g	Please provide a simple sentence that describes this image accurately.
Detail Caption $-$ p_d	Please describe all the visual concepts in the image in detail, but use concise words with no more than 120 words.
Prompt for Self-Instructed Concept-targeted Captions	
Compositional Caption $-$ p_{desc}	From the image, provide one sentence that describes {e} (you should try your best to include attributes like shape, color or material), especially, using {e} as the beginning of your answer.
OCR Extract $-$ p_{ocr}	List all the text in the image, answer with the ocr tokens only, and answer 'No' with one word if there isn't any.
Prompt for Consistency Filtering	
Caption Re-ranking $-$ $p_{\text{valid-c}}$	Is '{e}' a valid and visible visual concept in the image? Answer yes or no with only one single word.
Counting Group Filtering $-$ $p_{\text{valid-g}}$	Is there {parse times} or more {group key} in the image? Answer yes or no with a single word.
Symbol Explanation	
{e}	means an entity in the final detected entity list of this image.
{parse times}	means the number of times an entity appears in the entity list of this image.
{group key}	means the entity name corresponding to parse times in the entity list of this image.

Table 6: Prompt for *W2C* data construction pipeline.