

# BACON: Improving Clarity of Image Captions via Bag-of-Concept Graphs

## – *Supplementary Material* –

Zhantao Yang<sup>1,2\*</sup>, Ruili Feng<sup>2\*◇</sup>, Keyu Yan<sup>2</sup>, Huangji Wang<sup>1</sup>, Zhicai Wang<sup>2</sup>  
Shangwen Zhu<sup>1</sup>, Han Zhang<sup>1,2</sup>, Jie Xiao<sup>2</sup>, Pingyu Wu<sup>2</sup>, Kai Zhu<sup>2</sup>, Jixuan Chen<sup>2</sup>  
Chen-Wei Xie<sup>2</sup>, Yue Yang<sup>3</sup>, Hongyang Zhang<sup>4</sup>, Yu Liu<sup>2</sup>, Fan Cheng<sup>1†</sup>

<sup>1</sup>Shanghai Jiao Tong University, <sup>2</sup>Alibaba group

<sup>3</sup>University of Pennsylvania, <sup>4</sup>University of Waterloo

{ztyang196, ruilifengustc, yankeyu66, zhushangwen6, hzhang9617, jiexiao916, wpy364755620}@gmail.com

zhicaiw@outlook.com {kaizhustc, chenjixuan.cjx, eniac.xcw, ly103369}@alibaba-inc.com

yueyang1@seas.upenn.edu hongyang.zhang@uwaterloo.ca chengfan85@gmail.com

<https://ztyang23.github.io/bacon-page>

## Overview

The supplementary materials consist of four sections:

- **The first section is the details of BACON** (see Sec. 1), which provides additional details about the implementation of BACON, accompanied by a complete example. The implementation of BACON comprises three parts:
  - We introduce the VLM-readable string format mentioned in Section 3.2 of the main text (see Sec. 1.1.1)
  - We introduce the specific approach to applying in-context learning (ICL) (see Sec. 1.1.2)
  - We discuss how BACON-style captions facilitate integration with Grounding DINO to add grounding capability (see Sec. 1.1.3).
  - We provide detailed numerical experiments to show why VLM prefers element-wise outputs (see Sec. 1.1.4)
- **The second section covers the details of dataset construction** (see Sec. 2).
- **The third section focuses on LLAVA(BACON)-CAPTIONER** (see Sec. 3), including three parts:
  - We provide the training details of LLAVA(BACON)-CAPTIONER (see Sec. 3.1)
  - We compare the output distribution of LLAVA(BACON)-CAPTIONER and GPT-4V, demonstrating that LLAVA(BACON)-CAPTIONER can effectively replace GPT-4V to generate BACON-style captions (see Sec. 3.2)
  - We highlight the interesting capabilities of LLAVA(BACON)-CAPTIONER beyond generating BACON-style captions, such as interactively editing BACON-style captions, converting ordinary captions into BACON-style, and arranging bounding

boxes for all objects.(see Sec. 3.3)

- **The final section presents additional experimental details** (see Sec. 4), including five parts:
  - We provide additional details of the evaluation of LLAVA(BACON)-CAPTIONER on the open-vocabulary scene graph generation benchmark, with descriptions of the dataset and evaluation metrics (see Sec. 4.1)
  - We provide more details about the user study, which explains how to extract important object nouns from the captions during the user study (see Sec. 4.2)
  - We outline how BACON-style captions assist LLaVA in zero-shot region-based question answering (see Sec. 4.3)
  - We provide methodological specifics for using BACON-style captions with SAM-2 for dense video captioning, along with more examples (see Sec. 4.4)
  - We describe how BACON-style captions help SDXL with image generation, accompanied by additional examples. (see Sec. 4.5)
  - We additionally provide the experiments on BACON-style captions enhancing the multi-modal understanding capabilities of VLMs. (see Sec. 4.6)

## 1. Details of BACON

In this section, we begin by providing comprehensive details about the implementation of BACON in Sec. 1.1. This encompasses a thorough explanation of the VLM-readable string format mentioned in Section 3.2 of the main paper, details on the application of ICL techniques, and additional details on using Grounding DINO to obtain bounding boxes within the BACON-style captions. Lastly, we present a



Figure 1. An example of the VLM-readable string format.

complete example of BACON-style caption that was omitted from the main paper to save space.

## 1.1. Detailed method of implementing BACON

### 1.1.1. The VLM-readable string format

As discussed in Section 3.2 of the main paper, we convert the BACON-style caption into a VLM-readable string format to enable VLM generation. In this section, we introduce the string format, with an example illustrated in Fig. 1. This format is designed by sequentially concatenating all basic elements and separating them with special symbols. Specifically, we denote main titles with %% and subtitles with &&. When listing objects, we enclose additional details such as category, description, and color in parentheses (), with each detail separated by a semicolon ";". The name of an object is marked with <>. In describing relationships, we use <> to indicate objects and [] for the predicate. Furthermore, we use <> to highlight important objects within the context, which serves multiple purposes. One of its functions is to post-process the output from GPT-4V, allowing us to remove foreground information from the background description by deleting sentences where the foreground objects appear, or conversely, eliminating background information from the foreground description. By utilizing these special symbols to separate different sections, we can effortlessly organize the VLM output into a BACON-style caption using regular expressions.

### 1.1.2. The details of implementing ICL technique

As discussed in Section 3.2 of the main paper, we implemented the ICL technique to guide VLMs in responding with the desired string format outlined in Sec. 1.1.1. In practice, we discovered that GPT-4V does not require an extensive array of examples to understand the required format. Instead, incorporating just a few strategically chosen key examples within the instruction is sufficient. This approach allows us to process ICL within a single conversation, significantly reducing the costs associated with implementing BACON on the expensive GPT-4V. The final instruction is illustrated in Fig. 3, where we have highlighted the crucial examples in orange. Among the provided examples, some are specific while others are more general. We observed that general examples are particularly effective for straightforward structural elements. For instance, brief notations like 'lines 3-4' or 'lines 6-7' sufficiently illustrate the use of special symbols within a section, eliminating the necessity for expansive examples. In lines 16-17, we provide a general example that effectively clarifies the structure of each object, greatly reducing errors made by GPT-4V. To simplify the grasping of object details, we utilized a general example in lines 18-19, which proves adequate for generating simple sentences. Similarly, in lines 21-22, a general example suffices to guide GPT-4V on the fundamental pattern for depicting relationships. Lastly, the general example presented in lines 22-23 helps to prevent GPT-4V from erroneously generating two-way relationship pairs.

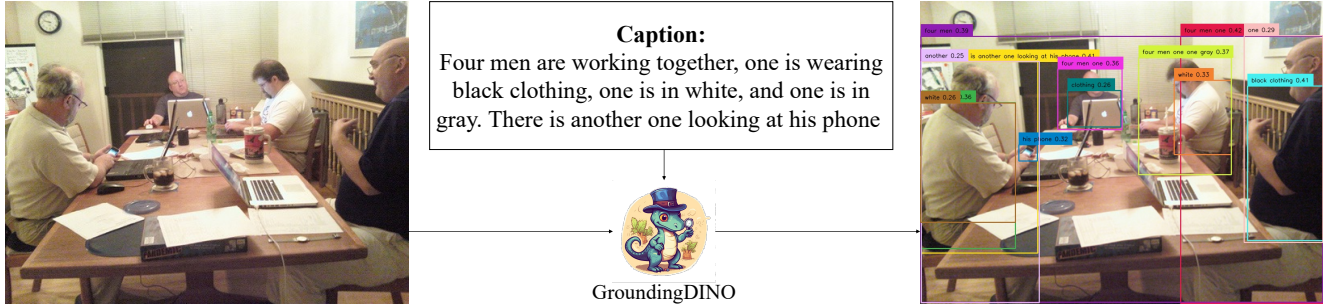


Figure 2. **An example of Grounding DINO performing the object detection task** illustrates its struggles with ambiguous labels and challenges in differentiating between individuals within the same category.

However, our stringent requirements concerning content and structure pose significant challenges, even for GPT-4V. As a result, it occasionally makes mistakes, such as omitting special symbols, even when general examples are provided. This highlights the necessity of including specific examples to ensure that GPT-4V accurately comprehends the required structure. For instance, when it comes to numbering items within the same category, we introduced a specific example in lines 11-12. Without this explicit reference, GPT-4V tends to overlook numbering, despite our earlier instructions in lines 10-11. Additionally, we noticed that while GPT-4V performs well with the format of the first section, it often falters in the second and third parts, complicating the transformation of data into a dictionary format. By offering just one clear example in lines 26-45 for these sections, we significantly increase the likelihood that GPT-4V will generate the correct structure. The implementation of the ICL technique has effectively ensured that nearly all of the 100,000 data entries we’ve collected are formatted correctly and can be seamlessly translated into a dictionary format.

### 1.1.3. Why BACON enables the introduction of Grounding DINO to add grounding capability

As discussed in Section 3.2 and Section 4.2.1 in the main paper, Grounding DINO is the leading grounding model, but it faces some issues. 1) First, it lacks the ability to understand long and complex sentences, resulting in its inability to accept more than a single noun as input. When the input is an image description, Grounding DINO struggles to extract nouns, which can lead to bizarre labels. For example, as illustrated in Fig. 2, Grounding DINO produces ambiguous labels such as “one,” “four men one gray,” and “another.” 2) The second issue, which is more significant, is Grounding DINO’s difficulty in differentiating between individuals of the same category. As shown in Fig. 2, while Grounding DINO correctly identifies four people, it remains challenging to determine which individual corresponds to each bounding box, often resulting in vague labels like “four men one.”

Fortunately, the structured format of BACON-style cap-

tions enables Grounding DINO to overcome these two challenges. For the first issue, BACON-style captions provide an accurate and comprehensive object list that serves as a reliable source of nouns. Regarding the second issue, as introduced in Figure 3 of the main paper, by leveraging the list of objects provided by BACON-style captions, along with detailed descriptions for each object, it becomes possible to utilize CLIP to make precise distinctions among different individuals sharing the same category label.

### 1.1.4. Numerical experiments of why VLMs may prefer element-wise outputs.

As discussed in Section 3.1 in the main paper, VLMs may prefer element-wise outputs for two reasons: 1) stronger attention maps of output tokens corresponding to specific image regions, and 2) higher semantic consistency scores in outputs with repeated requests. In this section, we present detailed numerical experiments, with the results illustrated in Figure 2 of the main paper.

**Numerical experiments.** We conduct numerical experiments to show these two key insights. Specifically, we compare a commonly used image captioning prompt, “Please describe this image in detail.”, with several questions targeting basic elements, including “Please list all objects in this image.”, “Please describe the object name in detail.”, and “What is the color information of object name?”. We randomly select 1,000 images from the MSCOCO dataset [6] and ask LLaVA these four questions, repeating each one 10 times with different random seeds. We then calculate the mean values of attention maps in the target regions and the semantic consistency scores.

For attention analysis, we extract objects and their corresponding target regions (segmentation masks) from the MSCOCO dataset annotations. For each object, we calculate the mean values of their attention maps within the target region across all output tokens. We then use the maximum value among all tokens as the final score for that object. As shown in Figure 2 (b) in the main paper, the questions

1 Hello, I would like to ask for your help in describing an image. Please note that I would like the description to be as detailed as possible. Please strictly respond  
2 following my instructions and do not print any redundant words.

3 This description needs to include three parts. The title of each part should be “%%Part1: Overall description%%”, “%%Part2: List of objects%%”, and “%%Part3:  
4 Relationships%%”. All important nouns in your response have to be bounded by ‘<’ and ‘>’!

5 The first part is an overall description of the image. Your answer to this part should consist of three parts, one sentence to describe the style of the image, one  
6 sentence to describe the theme of the image, and several sentences to describe the image. The titles of these parts are ‘&&Part1.1: Style&&’, ‘&&Part1.2:  
7 Theme&&’, ‘&&Part1.3: Global description of background&&’, ‘Part1.4: Global description of foreground&&’. The global description should be as detailed as  
8 possible and at least 150 words in total. If there is text content in the image, you can also describe the text, which should be bound by quotation marks. All  
9 important nouns in your response have to be bounded by ‘<’ and ‘>’!

10 The second part is to list all the objects in the image, as many as possible, in order of importance. Note that any object should not be a part of other objects. Note  
11 that the listed object should not be the plural. If there are multiple individuals of the same category of objects, please list them separately. For example, if there  
12 are three apples in the picture, they should be listed as ‘Apple 1,’ ‘Apple 2,’ and ‘Apple 3,’ respectively. Additionally, the objects should be classified into two  
13 categories: living and inanimate objects. Living refers to creatures such as humans, cats, dogs, and plants, while other lifeless objects belong to the category of  
14 inanimate objects. Finally, each object should have a very detailed description, with more important objects receiving more detailed descriptions. Each  
15 description should be at least 30 words and the important nouns in it have to be bounded by ‘<’ and ‘>’. You should also identify whether this object belongs to  
16 the foreground or background. You should additionally provide a sentence to describe the color information of the object. Therefore, the format for listing each  
17 object should be ‘Object Name (Category (Living/Inanimate); foreground/background; Description; Color information)’. Specifically, the detailed description of  
18 an object should focus on its part and its action. All descriptions should be in the forms of, object’s + part + verb + object/adjective or object + is + present  
19 participle. The description should be detailed as well as possible, and try to describe all parts of this object. You should specifically notice if there is a sky, tree,  
20 sun, or other object in the background of the environment. All important nouns in your response have to be bounded by ‘<’ and ‘>’!

21 The third part is to describe the relationships between all the objects in pairs. Please list them one by one. Additionally, please describe the relationship between  
22 object A and object B in the format of ‘Object A’ + ‘Action’ + ‘Object B’. Please don’t print the same relation twice. For example, if there is “A relation B”, you  
23 shouldn’t print ‘B relation A’ again. All important nouns in your response have to be bounded by ‘<’ and ‘>’!

24 I will provide you with an example of the last two parts of a description to show you the desired format. You should only focus on the format of this example  
25 instead of the content of it. You should use the same format to respond.

26 “%%Part2: List of objects%%  
27 <Woman> (Living; foreground; The <woman>’s <hair> is bundled in a <scarf>. Her <torso> is covered with a <black shirt>. Her <lower body> is clad in <blue  
28 jeans>. Her <legs> move through the <water>. Her <right hand> holds a pair of <shoes>; Color information: <black> shirt, <blue> jeans, <orange> scarf.)  
29 <Water> (Inanimate; foreground/background; The <water> floods the <street>, reflecting the <sky> and <surrounding objects>; Color information: <murky  
30 blue-grey>.)  
31 <Building 1> (Inanimate; background; The <building> has a <façade> with <doors> and <windows>, showing signs of <water damage>; Color information:  
32 <pale yellow>.)  
33 <Building 2> (Inanimate; background; This <building> is similar to <Building 1> but with a <red> roof visible above the <flood>; Color information: <light  
34 orange> walls, <red> roof.)  
35 <Vehicle 1> (Inanimate; background; A <vehicle> is partially submerged, showing only the <roof> and <upper parts>; Color information: <white>.)  
36 <Vehicle 2> (Inanimate; background; Another <vehicle>, also partially submerged, with a <visible logo>; Color information: <silver>.)  
37 <Sky> (Inanimate; background; The <sky> is filled with <clouds>, implying recent or ongoing <precipitation>; Color information: <gray>.)  
38 %%Part3: Relationships%%  
39 <Woman> [is walking through] <Water>.  
40 <Woman> [is moving away from] <Camera>.  
41 <Water> [reflects] <Sky>.  
42 <Water> [surrounds] <Vehicles>.  
43 <Buildings> [line] <Street>.  
44 <Vehicle 1> [is submerged by] <Water>.  
45 <Vehicle 2> [is submerged by] <Water>.

Figure 3. **The instruction** for GPT-4V to obtain the BACON-style caption from an image. We highlight the parts involving specific examples in orange.

targeting basic elements yield more pronounced attention maps compared to the standard prompt for description, indicating VLMs understand their meaning more firmly.

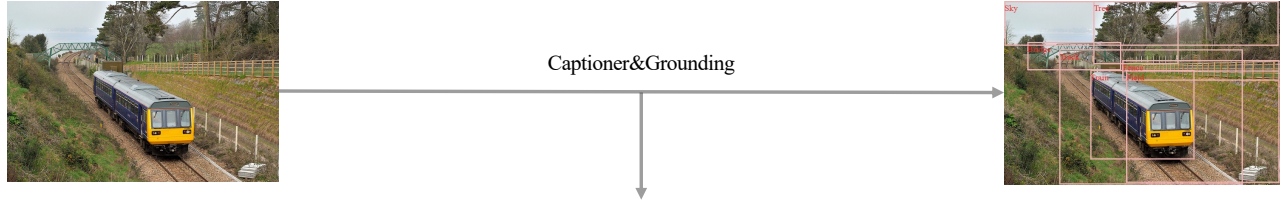
For the semantic consistency analysis, we compute the similarity evaluated by T5 [9] of answer sub-sentences among  $n$  independent repetitions of question-answering. For each answer  $a_i$ ,  $1 \leq i \leq n$ , we split it into  $k_i$  sub-sentences based on punctuations. We then compute the semantic consistency score between  $a_i$  and  $a_j$  by  $S(a_i, a_j) = \frac{1}{2}(F(a_i|a_j) + F(a_j|a_i))$ , where  $F(a_i|a_j) =$

$\sum_{m=1}^{k_i} s(a_i^m, a_j)$ . Here,  $s(a_i^m, a_j)$  is defined as

$$s(a_i^m, a_j) = \begin{cases} 1 & \text{if } \max_{1 \leq n \leq k_j} \frac{\langle T5(a_i^m), T5(a_j^n) \rangle}{\|T5(a_i^m)\|_2 \|T5(a_j^n)\|_2} \geq \rho \\ 0 & \text{else} \end{cases} \quad (1)$$

where  $\rho$  is a threshold.  $s(a_i^m, a_j) = 1$  indicates that a similar sub-sentence of  $a_i^m$  can be found in  $a_j$ . As illustrated in Figure 2 (c) in the main paper, the three questions targeting basic elements exhibit much higher semantic consistency scores than the normal request for image description,





#### Overall description:

**Style:** 'The image is a photograph with a realistic style.'

**Theme:** 'The theme of the image is transportation, specifically a train traveling through a rural landscape.'

**Background description:** 'The background of the image features a rural landscape with elements of nature and infrastructure. There is a bridge with green metal railings crossing over the train tracks. Beyond the bridge, a fence made of wooden posts and rails encloses a field. The field appears to be grassy with some patches of bare earth. The sky is overcast, with a pale, diffused light suggesting an overcast or cloudy day.'

**Foreground description:** 'In the foreground, a train is captured in motion on the tracks. The train is painted in a blue and yellow color scheme. The train has multiple carriages, and the windows reflect the surrounding environment. The tracks are made of steel rails with wooden sleepers, and they run parallel to a grassy embankment on the left side of the image.'

#### Object list:

- Train:** Category: inanimate, foreground  
Description: 'The train's body is long and sleek, with windows lined along its side. The front car has a curved nose with a destination sign and headlights. The train is composed of several carriages connected together.'  
Color: blue and yellow  
Position: [200, 160, 441, 367]
- Track:** Category: inanimate, foreground  
Description: 'The track consists of parallel steel rails supported by wooden sleepers. It stretches into the distance, guiding the train.'  
Color: rusty brown rails, brown sleepers  
Position: [128, 112, 553, 425]
- Bridge:** Category: inanimate, background  
Description: 'The bridge spans over the tracks with a structure made of metal beams and railings. It appears functional and unadorned.'  
Color: Description railings  
Position: [54, 95, 271, 160]
- Fence:** Category: inanimate, background  
Description: 'The fence is constructed of wooden posts and rails, enclosing the field and providing a boundary.'  
Color: natural wood tone  
Position: [274, 137, 638, 184]
- Field:** Category: inanimate, background  
Description: 'The field is predominantly grass-covered, with some areas of bare soil. It is bordered by the fence and trees.'  
Color: green grass, brown soil  
Position: [283, 161, 638, 421]
- Tree:** Category: inanimate, background  
Description: 'The trees have bare branches, indicating a lack of leaves which could suggest a seasonal change.'  
Color: dark brown branches  
Position: [207, 0, 404, 146]
- Sky:** Category: inanimate, background  
Description: 'The sky is overcast, with a uniform light grey color, suggesting cloudy weather.'  
Color: light grey  
Position: [1, 0, 636, 103]

#### Relationship:

- <Train,Track>: 'is traveling on'
- <Train,Bridge>: 'is passing under'
- <Bridge,Track>: 'spans over'
- <Fence,Field>: 'encloses'
- <Field,Tree>: 'is bordered by'
- <Field,Fence>: 'is bordered by'
- <Tree,Field>: 'is standing in'

Figure 4. A complete example of BACON-style caption.

meaning that the VLMs are very confident and clear in what the answers should be.

## 1.2. Complete examples of BACON-style captions

We provide a complete example of BACON-style caption in Fig. 4

## 2. Detailed method of building ECO

### 2.1. Detailed method of building the training set

The structure of BACON-style captions significantly streamlines the workload of annotation during the training

data collection process. By breaking down complex descriptions into basic elements, many of which require annotators to simply make a straightforward right-or-wrong judgment, the task becomes highly manageable. For larger segments of information, such as background or foreground descriptions, annotators are instructed to independently assess whether each sentence accurately reflects the image. Additionally, they are asked to identify and add any objects missed by GPT-4V. The design of our structure for object descriptions further aids annotators by simplifying the annotation process; they only need to fill in the corresponding information according to the established format.

Table 1. **Comparison of plan task** between LLAVA(BACON)-CAPTIONER (ours) and LayoutGPT [2] on both MSCOCO and test set of ECO.

Dataset	Method	Precision	Recall	mIOU
MSCOCO	LayoutGPT	70.1%	39.7%	4.1%
	LLAVA(BACON)-CAPTIONER	<b>71.2%</b>	<b>41.8%</b>	<b>6.8%</b>
ECO	LayoutGPT	50.8%	29.2%	9.1%
	LLAVA(BACON)-CAPTIONER	<b>51.7%</b>	<b>47.1%</b>	<b>18.4%</b>

## 2.2. Detailed method of building the test set

Despite the impressive capabilities of GPT-4V, it may occasionally overlook certain objects. To achieve maximum accuracy, we employ an entirely different pipeline that involves separately querying each basic element to create the test set for ECO. This method comprises five distinct steps: 1) The Segment anything (SAM) [4] model segments all components within the image, helping to prevent the omission of objects. 2) VLMs identify the names of objects in the masked image obtained from the first step. 3) Using the names identified in the second step, VLMs annotate each object in detail. 4) VLMs generate an overall description of the image based on the list of objects derived from the above steps. 5) Images created by randomly pairing two masked images from the first step are fed to the VLMs to identify the relationship between the two objects in the selected masked images. Annotators are involved in steps 2 through 5. In Step 2, they are responsible for correcting the results returned by the VLMs to accurately identify the object names given the masked images. In Steps 3 and 4, annotators are tasked with verifying the accuracy of each generated sentence. They do not need to add objects, as the Segment Anything (SAM) method [4] ensures that there are no omissions. In Step 5, annotators must assess whether the identified relationships are correct and add any significant relationships that may have been overlooked by the VLMs.

## 3. Details of LLAVA(BACON)-CAPTIONER

### 3.1. Details of training LLAVA(BACON)-CAPTIONER

LLAVA(BACON)-CAPTIONER is fine-tuned on ECO using a pre-trained 13B LLaVA model with the Low-Rank Adaptation (LoRA) technique [3]. The total number of LoRA parameters is approximately 0.5 billion. Training is conducted on NVIDIA A100 GPUs and requires about 100 GPU hours, utilizing a learning rate of  $2 \times 10^{-4}$  for 3 epochs. The batch size used is 16, and the LoRA rank is set to 128. Additionally, we implement a warmup ratio of 0.03 during training.

### 3.2. Comparison between LLAVA(BACON)-CAPTIONER with GPT-4V on obtaining BACON-style captions

We present an analysis of the root words and categories identified in the outputs of LLAVA(BACON)-CAPTIONER, as illustrated in Fig. 7. The results clearly indicate that the output distributions of LLAVA(BACON)-CAPTIONER are very similar to those of GPT-4V. Notably, there is a 100% overlap in the top 100 most frequent nouns, a 99% overlap for verbs, and a 97% overlap for categories detected by both GPT-4V and LLAVA(BACON)-CAPTIONER. This similarity confirms that **LLAVA(BACON)-CAPTIONER can effectively take over from GPT-4V in generating BACON** from images, thereby extending our ECO.

### 3.3. Additional capabilities of LLAVA(BACON)-CAPTIONER

In addition to generating BACON-style captions from images, the trained LLAVA(BACON)-CAPTIONER excels in several additional functions, including interactively editing BACON-style captions by requesting desired changes from the LLAVA(BACON)-CAPTIONER, transforming ordinary prompts into BACON-style captions, and planning the positions of objects within the object list. First, as illustrated in Fig. 6, **the LLAVA(BACON)-CAPTIONER enables interactive editing of BACON-style captions**, thereby influencing the image generation process. Besides and remarkably, without requiring any fine-tuning, **the LLAVA(BACON)-CAPTIONER can convert a standard prompt into a BACON-style caption**. This capability is particularly important for image generation, given the challenges of manually providing BACON-style prompts. Furthermore, **the LLAVA(BACON)-CAPTIONER can effectively arrange the positions of objects within the object list**. Examples of using LLAVA(BACON)-CAPTIONER to organize prompts and arrange the positions of objects for image generation can be found in Figs. 11 and 12. We quantitatively evaluate the planning capabilities of the LLAVA(BACON)-CAPTIONER against the leading LayoutGPT [2] on the MSCOCO dataset [6] and our BACON datasets, employing mIoU, precision, and recall metrics [2]. The results presented in Tab. 1 demonstrate that the LLAVA(BACON)-CAPTIONER outperforms LayoutGPT across both evaluated datasets.

## 4. Additional details of experiments

In this section, we provide more details about the experiments part (Section 4 in the main paper). This includes details of the five experiments discussed respectively in Section 4.1.2 (Analyzing Object & Relation Accuracy using Open-vocabulary scene graph generation), Section 4.1.3

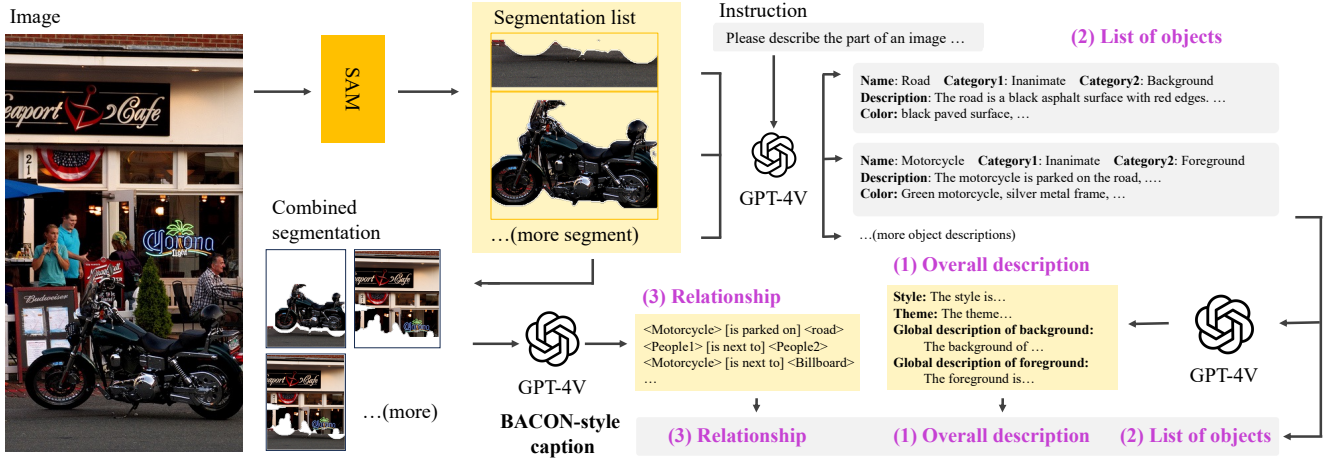


Figure 5. **A detailed overview of the method used to collect the test set of ECO**, segmented into five distinct steps. 1) The SAM model segments all components within the image. 2) VLMs identify the names of objects in the masked image obtained from the first step. 3) Using the names identified in the second step, VLMs annotate each object in detail. 4) VLMs generate an overall description of the image based on the list of objects derived from the above steps. 5) images created by randomly pairing two masked images from the first step are fed to VLMs to identify the relationship between the combined segments. It is important to note that human annotation is required to correct and verify the outputs from steps two through five.

Table 2. **Comparison of the VLM trained on QA data** derived from ECO (BACON-13B, Ours) with other VLMs across 7 general benchmarks.

Model	GQA	SQA <sup>I</sup>	POPE	MMB	MMB <sup>CN</sup>	SEED	MM-Vet
Qwen-VL	59.3*	67.1	-	38.2	7.4	56.3	-
Qwen-VL-Chat	57.5*	68.2	-	60.6	56.7	58.2	-
LLaVA-13B	63.3	71.6	85.9	67.7	63.6	61.6	35.4
VILA-13B	63.3	73.7	84.2	70.3	64.3	62.8	38.8
Ours	<b>63.5</b>	<b>91.3</b>	<b>88.0</b>	<b>74.6</b>	<b>68.2</b>	<b>65.9</b>	<b>41.6</b>

(Analyzing Precision & recall using user study), Section 4.2.2 (Zero-shot region-based question answering), Section 4.2.3 (Multi-object video tracking and dense video captioning), and Section 4.2.4 (Image generation) of the main text. Each experiment is presented in its own subsection, covering aspects that are omitted from the main text, including detailed evaluation metrics, the methods employed by the models to utilize BACON-style captions, and other relevant details.

#### 4.1. Details of evaluating LLaVA(BACON)-CAPTIONER on open-vocabulary scene-graph generation benchmark

As discussed in Section 4.1.2 in the main paper, we use the open-vocabulary scene graph generation benchmark to evaluate the performance of LLaVA(BACON)-CAPTIONER. In this section, we introduce more details about the evaluation method from two perspectives, the dataset and the evaluation metrics.

**Visual Genome.** As introduced in the main paper, Visual Genome (VG) [5] is employed for evaluation. VG is

an open-vocabulary dataset; however, most current scene graph generation (SGG) models typically consider only a limited number of categories. Consequently, researchers often treat it as a dataset with a restricted set of categories. Specifically, they usually identify the 70 or 150 most frequent noun classes, along with the 50 most common predicates, to create a filtered dataset. In our case, since we are working on an open-vocabulary scene graph generation (OV-SGG) task, we treat the VG dataset as an open-vocabulary resource and utilize only the triplets containing nouns or predicates that fall outside the commonly used set.

**Evaluation metrics.** Traditional SGG tasks often utilize recall-related metrics to evaluate performance, specifically assessing how many (subject-predicate-object) triplets present in an image are correctly predicted. Previous models typically perform classification tasks within a fixed set of categories and use the confidence of those classifications to obtain the top K predictions with the highest likelihood. Any ground truth triplets that appear within the top K predictions are considered correctly predicted.

However, in an open-vocabulary setting, the possibilities are virtually infinite, making it challenging to identify the top K predictions. Therefore, we employ the CLIP [8] score to determine whether a prediction is correct. To do this, we construct a string for each triplet in the format “{subject}{predicate}{object}” and calculate the CLIP similarity between the prediction and the ground truth. If we find that a ground truth has a CLIP similarity score with the prediction that exceeds a specified threshold (0.9 in this case), and the Intersection Over Union (IOU) of the subject and object positions between the prediction and the ground

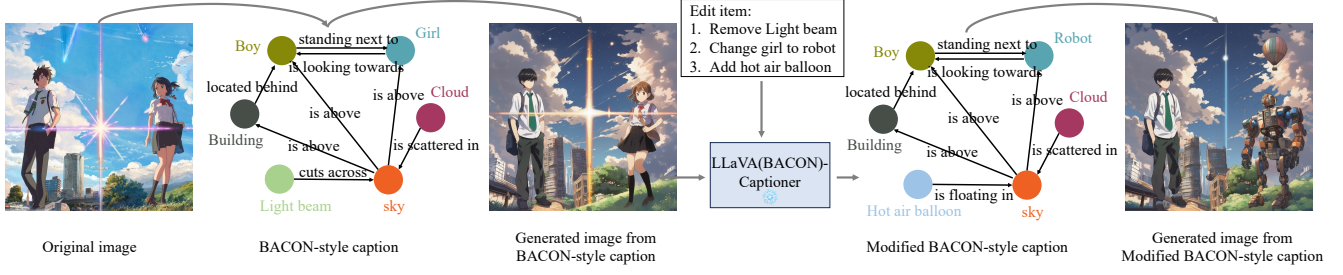


Figure 6. An example of interactively modifying BACON using LLaVA(BACON)-CAPTIONER.

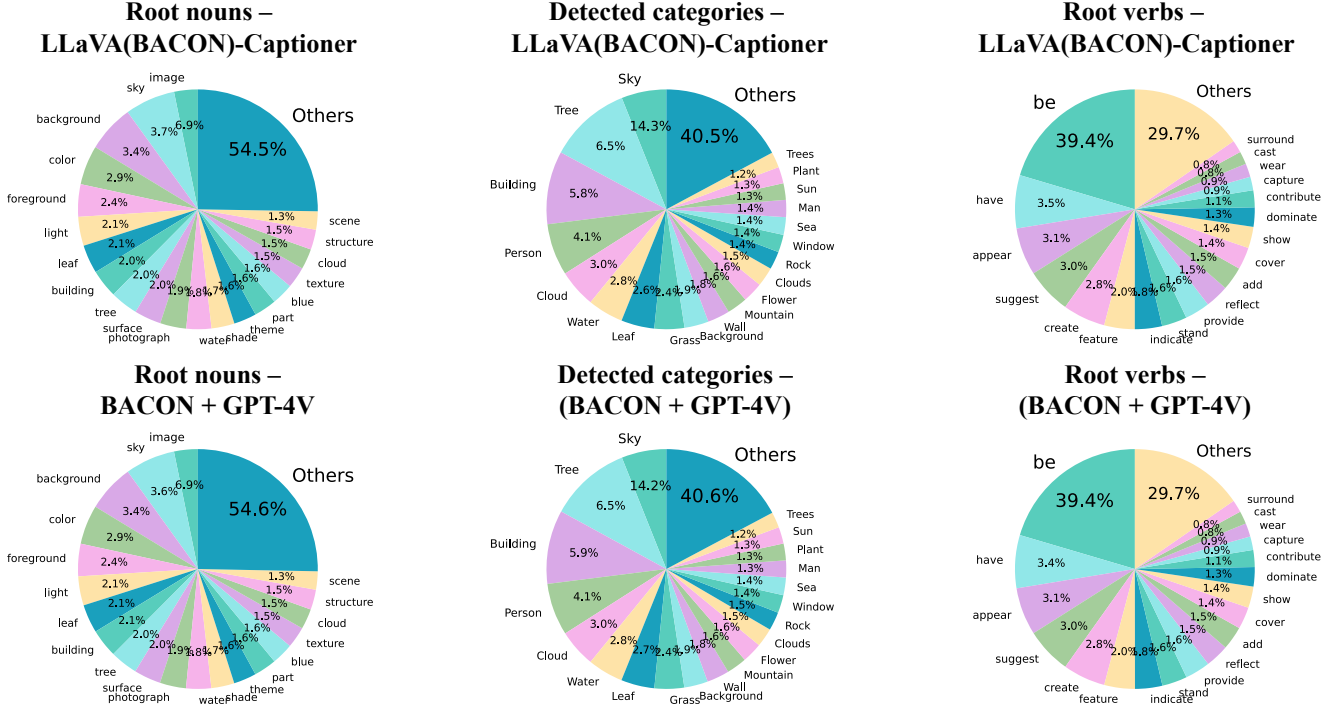


Figure 7. Comparison of the root words and detected categories generated by LLaVA(BACON)-CAPTIONER and GPT-4V on the test set of ECO (certain sections magnified for clearer visualization). The results reveal that the output distribution of BACON closely resembles that of GPT-4V, demonstrating that LLaVA(BACON)-CAPTIONER can effectively replace the expensive GPT-4V in generating BACON-style captions.

truth also surpasses another threshold (0.5 here), we consider it a correct prediction. Consequently, we can calculate the recall score.

#### 4.2. Additional details for user study

As discussed in Section 4.1.3 of the main paper, calculating precision and recall involves identifying the objects predicted by different captioners. For captioners other than LLaVA(BACON)-CAPTIONER, this can be challenging, as directly extracting nouns may include many terms that cannot be considered objects. To address this issue, we utilize VLMs for the task. Specifically, we input the captions into the VLMs and request them to extract the objects contained within. In contrast, this process is straightforward for LLaVA(BACON)-CAPTIONER, as BACON explicitly

provides a list of objects. This emphasizes the superiority of LLaVA(BACON)-CAPTIONER.

#### 4.3. Additional details of BACON-style captions assist LLaVA in zero-shot region-based question answering.

LLaVA lacks the capability to connect information to its corresponding regions without fine-tuning, making it challenging for LLaVA to perform region-based question-answering tasks. Fortunately, BACON-style captions provide bounding boxes for all objects along with their descriptions, enabling the connection between descriptions and regions. In Section 4.2.2 of the main paper, we have introduced the benchmarks used, the evaluation metrics employed, and provided the experimental results. Here,



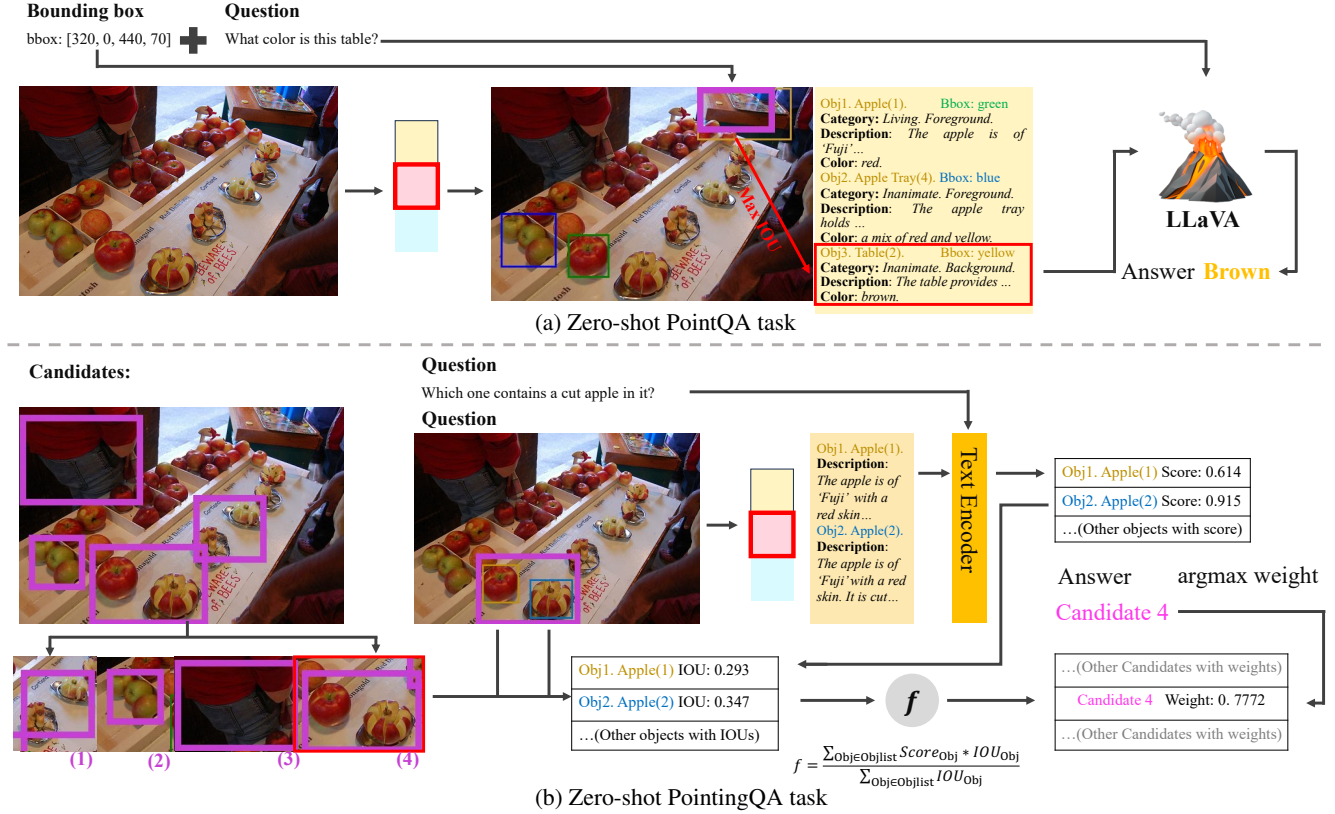


Figure 8. An illustrative diagram depicting how BACON-style captions aid downstream models in executing zero-shot PointQA and zero-shot PointingQA tasks. In (a) the zero-shot PointQA task, the description of the object that has a significant overlap with the target region is used to characterize that region. This regional description is then input into a QA model to answer questions related to the area. In (b) the zero-shot PointingQA task, object descriptions provided by BACON are used to calculate similarity scores with the input question, generating scores for each object. Based on the overlap between object positions and candidate regions, a weighted sum of all object scores is computed to assign scores to candidate regions; the region with the highest score is then selected as the prediction.

we present the detailed method by which LLaVA utilizes BACON-style captions to perform zero-shot region-based question answering, including both zero-shot PointQA and zero-shot PointingQA.

**Zero-shot PointQA** Zero-shot PointQA is designed to answer questions related to specific regions of an image solely based on the image caption instead of the image itself. Given a target area, we can create a description relevant to that location by combining the descriptions of different objects based on their positions. Specifically, as illustrated in Fig. 8(a), we calculate the Intersection Over Union (IOU) between the target area and the positions of all objects. By combining the descriptions of objects with large overlaps, we generate a description that is closely linked to the target area. Subsequently, we input this region-based description into the question-answering model to derive the answer to the question.

**Zero-shot PointingQA** The Zero-shot PointingQA task involves selecting the most appropriate region from a set of candidate areas based on a given textual request and the image’s description, rather than the image itself. As illustrated

in Fig. 8(b), the method consists of three steps: 1) First, we compute the CLIP similarity between each object’s description and the input textual prompt, resulting in scores for each object. The more relevant an object is to the text description, the higher its score will be. 2) Next, we calculate scores for each candidate region by weighting the sum of object scores based on the overlap between the candidate region and the object’s location. The greater the overlap with the candidate area, the larger the contribution of that object’s score. 3) In the final step, we select the region with the highest score as the answer.

#### 4.4. Additional details of BACON-style captions assist SAM-2 in dense video captioning.

**Methods.** As discussed in Section 4.2.3, BACON-style captions can assist SAM-2 in performing dense video captioning tasks by providing bounding boxes as indicators and supplying the descriptions that SAM-2 lacks. While this method is a reliable approach for video captioning, it still encounters two challenges. The first challenge involves managing newly appeared objects in the video, and the sec-



Figure 9. An example of LLAVA(BACON)-CAPTIONER assists SAM-2 on dense video captioning.

ond is how to ensure that the caption content evolves as the video progresses. To address these challenges: 1) We begin by uniformly sampling several frames from the video and employing tracking from each selected frame to monitor the entire video. We then consolidate the tracking results across different frames and assign the same ID to objects with exceptionally high mask overlap. 2) Additionally, we utilize T5 [9] as the text encoder to compare descriptions of the same object or scene segment across frames. Portions of the text with high similarity scores are deemed stable, while those with low similarity scores are identified as having changed.

**More results.** We provide more examples in Figs. 9 and 10

#### 4.5. Additional details of BACON-style captions assist SDXL in image generation.

**Methods.** Even as one of the most renowned models for text-to-image generation, SDXL often struggles to understand complex prompts and generate precise images accurately. This is primarily because SDXL employs CLIP for text understanding, which limits its ability to comprehend

the text. Fortunately, by breaking down complex texts into basic elements, BACON-style captions can significantly assist SDXL in simplifying complex tasks. The specific method by which BACON-style captions enhance SDXL for image generation can be divided into three steps:

- **Step 1:** Given a natural prompt for generation, the trained LLAVA(BACON)-CAPTIONER converts it into a BACON-style caption and arranges the bounding boxes for all objects listed in the caption.
- **Step 2:** SDXL separately generates all components, including the background and each object. To create the background, SDXL uses the background description. Besides, it relies on the detailed descriptions of objects to generate them.
- **Step 3:** For the generated image of the objects, we extract the main components by segmenting them from the image using SAM [4]. These components are then combined according to their arranged positions from Step 2. Finally, the combined image is refined using commonly employed refinement methods, including Anydoor [1], Collage Diffusion [11], inpainting [10], and SDEdit [7]. Some meth-



Figure 10. An example of LLaVA(BACON)-CAPTIONER assists SAM-2 on dense video captioning.

ods are applied during the combination process, while others are utilized afterward.

**More results.** We provide more examples in Fig. 13

#### 4.6. BACON-style captions enhance the multi-modal understanding capabilities of VLMs.

In this section, we show that the multi-modal understanding capabilities of VLMs can be further improved by training on BACON-style captions. To demonstrate this, we fine-tune an LLaVA-13B using ECO by utilizing QA data derived from them. Specifically, the structured nature of the captions allows us to automatically generate multiple QA pairs from a single caption, such as 'Is object A to the right of object B?', 'Is object A red?', and 'Please describe object A.' We present comparison results of this VLM against previous models on commonly used benchmarks in Tab. 2. Results suggest that data collected through BACON enhances the multi-modal understanding capabilities of VLMs.



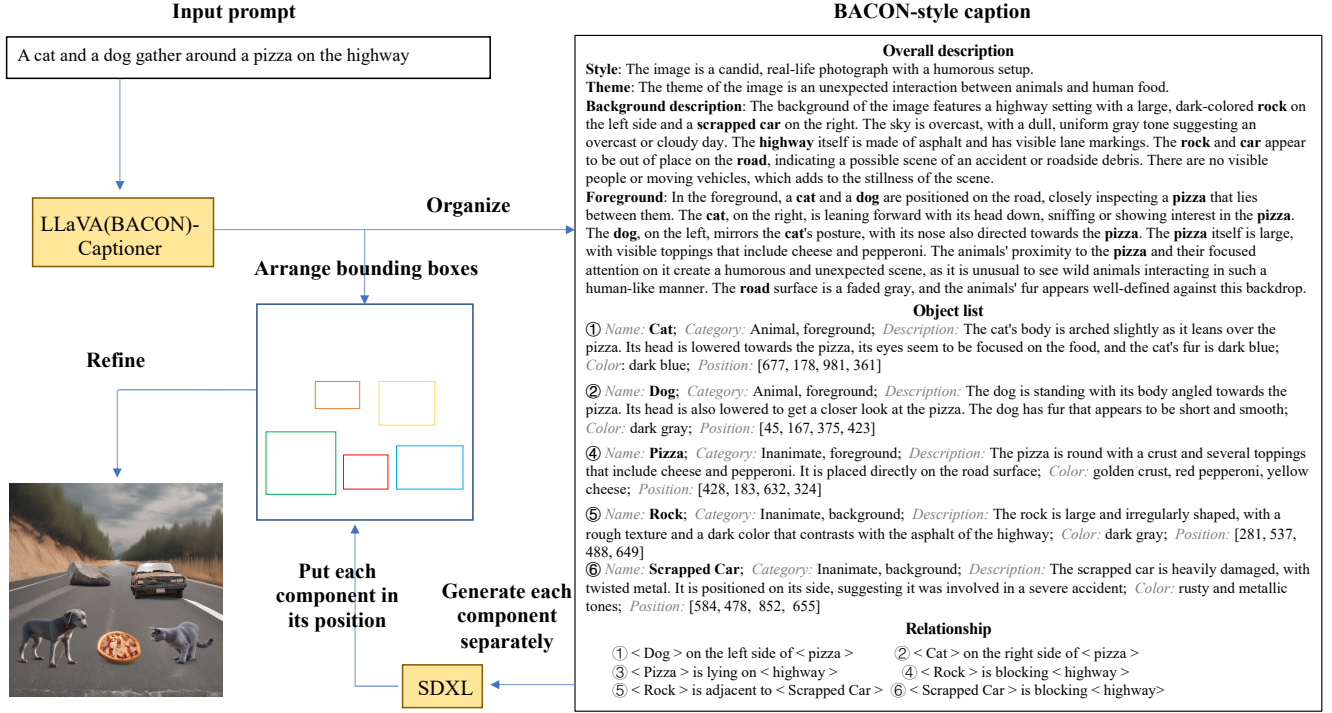


Figure 11. An example of how LLaVA(BACON)-CAPTIONER boosts SDXL for image generation. First, LLaVA(BACON)-CAPTIONER converts the prompt into the BACON-style and arranges the bounding boxes of all objects. Then, each component is generated separately by SDXL and then placed in its arranged position. Finally, the combined image is refined to produce the final image.

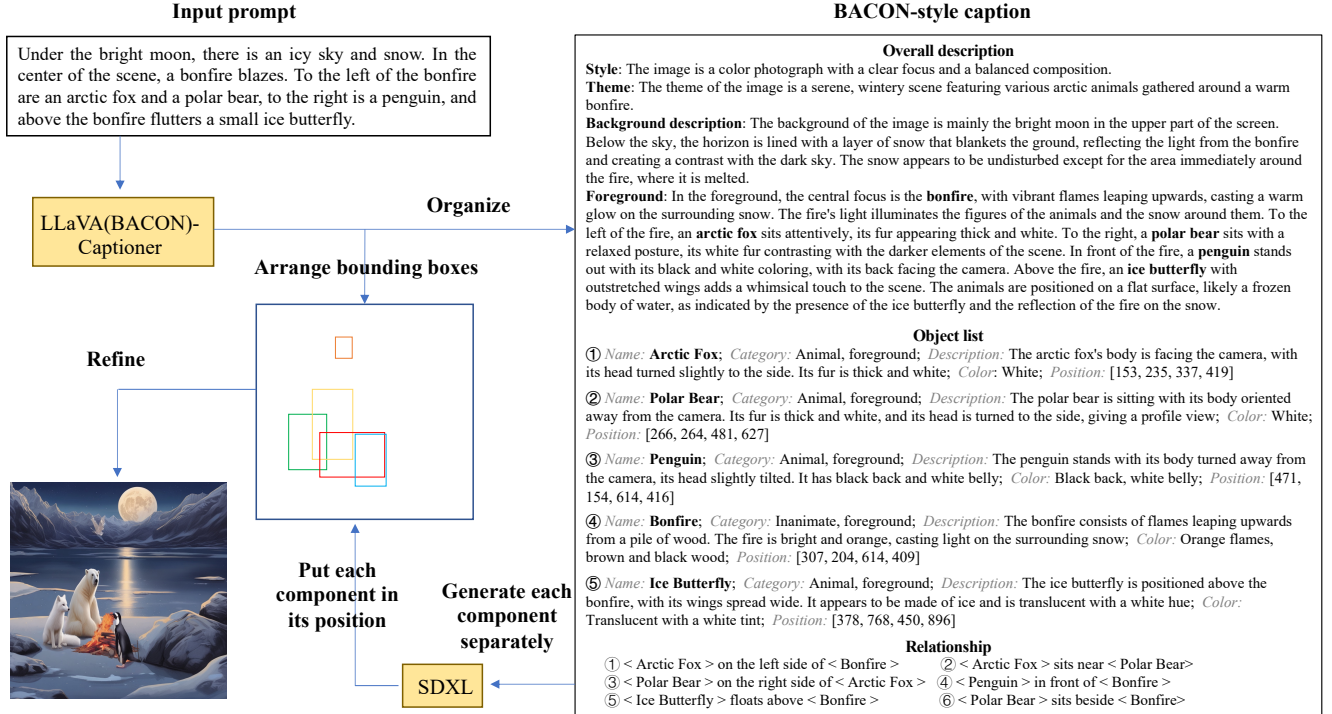


Figure 12. An example of how LLaVA(BACON)-CAPTIONER boosts SDXL for image generation. First, LLaVA(BACON)-CAPTIONER converts the prompt into the BACON-style and arranges the bounding boxes of all objects. Then, each component is generated separately by SDXL and then placed in its arranged position. Finally, the combined image is refined to produce the final image.

















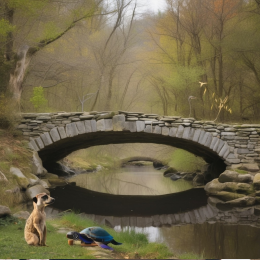



Input Prompt	SDXL	DALL-E 3	LLaVA(BACON)-Captioner + SDXL
In the deep sea, an abandoned large ship full of marine life sank to the bottom of the sea. There are two blue balloons floating in front of the ship. There is a dolphin swimming below the balloon. There is a drifting bottle floating in the deep sea, inside which is a sailboat			
In a yoga studio, there is an artwork of a green jade dragon, with a white cat lying on the right side of the artwork. On the distant ground, against the wall, there is a painting depicting war			
In an abandoned factory building, sunlight filtered in. A technologically advanced spaceship flies over the factory building. Listening to a motorcycle below the spaceship, there is a pink guitar on the ground to the right of the motorcycle.			
On a pink night, there was a pool in the center of the lawn, and a purple sports car was floating on the pool. There was a light bulb on the hood of the sports car, and there was an orange goldfish in the bulb. On the left side of the car is a small, colorful robot			
There is a small river in the forest, and there is a stone bridge on the river. There is a golden praying mantis on the bridge. There is a mongoose standing by the riverbank, and to its right lies a turtle			
In an old-fashioned subway station, there is a emerald green lion, a gray white wolf, and a colorful paper crane standing together waiting for the subway			

Figure 13. Additional examples of BACON boosting SDXL on image generation.

## References

- [1] Xi Chen, Lianghua Huang, Yu Liu, Yujun Shen, Deli Zhao, and Hengshuang Zhao. Anydoor: Zero-shot object-level image customization. In *CVPR*, 2024. [10](#)
- [2] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *NeurIPS*, 2024. [6](#)
- [3] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. [6](#)
- [4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. [6](#), [10](#)
- [5] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017. [7](#)
- [6] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. [3](#), [6](#)
- [7] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021. [10](#)
- [8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. 2021. [7](#)
- [9] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. 2020. [4](#), [10](#)
- [10] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. [10](#)
- [11] Vishnu Sarukkai, Linden Li, Arden Ma, Christopher Ré, and Kayvon Fatahalian. Collage diffusion. pages 4208–4217, 2024. [10](#)