Auto Cherry-Picker: Learning from High-quality Generative Data Driven by Language

Supplementary Material

6. Implementation Details of ACP

In this section, we provide implementation details of ACP. Following our pipeline, we introduce details of Data Priors in Sec. 6.1, Scene Graph Generator in Sec. 6.2, Image Generator in Sec. 6.3, and Image Filter in Sec. 6.4.

6.1. Data Priors

Data Priors Construction. Data Priors P is constructed from open-source datasets D_P, where each sample contains an image caption and associated annotations. We use LLMs to parse each caption and extract object combination (s_i, o_i) along with their relationship r_i, forming $\{(s_i, o_i), r_i\}_{i=1}^{N}$. For each extracted item $\{(s_i, o_i), r_i\}$, we retrieve the corresponding bounding boxes from annotations as $(l_s, l_o)_k^i$. By iterating over the entire D_P, we gather all relevant layout pairs corresponding to $\{(s_i, o_i), r_i\}$ into a list L_i. Thus, each data prior is formulated as $p_i = ((s_i, o_i), r_i, L_i)$, aligning with Eq. 1.

Reference Layouts Selection from Data Priors. Reference layouts selection is based on the object combination (s, o) and their relationship r. We extract reference layouts from P that share the same category and relationship.

6.2. Scene Graph Generator

Choice of LLMs. We conduct experiments using a series of LLMs as the scene graph generator in our ACP pipeline on a limited data scale. Specifically, we employ Qwen-1.5-14B, Qwen-1.5-72B, and Qwen-1.5-110B to generate scene graphs. Each model produces 15K training examples from the same input object lists. These examples are then amalgamated with the original training data for COCO detection tasks. We apply them separately to a Mask R-CNN baseline under a standard $1\times$ training schedule. Table 7 illustrates that the performance of different LLMs is comparable in downstream tasks. We opt for the smaller LLM, Qwen-1.5-14B, for the experiments described due to its faster inference speed.

Prompts. We provide our full prompts for the scene graph generator, including the description generator and layout generator.

Details of Layout Generation. We derive the input from the previous description and prompt LLMs to generate a layout for each object. The input follows a dictionary format, e.g., {"objects": ["xx-1", "xx-2", "xx-3"], "caption": "xxx"}, and the output is a list of dictionaries, e.g., [{"object": "xx-1", "layout": [x,y,w,h]}]. The Raw Layout in Fig 3(a) represents the complete image layout, encompassing all object layouts for single-image generation.

Table 7. Different LLMs as scene graph generator in ACP on COCO detection task.

Scene Graph Generator	$AP^{mask}\uparrow$	$AP^{box}\uparrow$
Qwen1.5-14b	34.5	37.8
Qwen1.5-72b	34.5	37.8
Qwen1.5-110b	34.2	37.7

Prompt for Description Generator:
Task Description:
Your task is to generate a detailed description based on an object list. The description should be a structured representation of a scene detailing its various elements and their relationships. The description consists of: 1. attributes of objects: The attributes should be descriptive of the color or texture of the corresponding object. 2. Groups: A group of objects exhibit strong spatial relationships that interact with each other. 3. Relationships: This section illustrates the interactions or spatial relationships between various objects or groups. 4. Caption: Caption should be a simple and straightforward 2-4 sentence image caption. Please include all the objects in the caption and refer to them in '()'. Create the caption as if you are directly observing the image. Do not mention the use of any source data. Do not use words like 'indicate', 'suggest', 'hint', 'likely', or 'possibly'.

You can refer to the following examples as references.
In-context learning examples for Description Generator

Please provide a json format with Description based on the following object list.

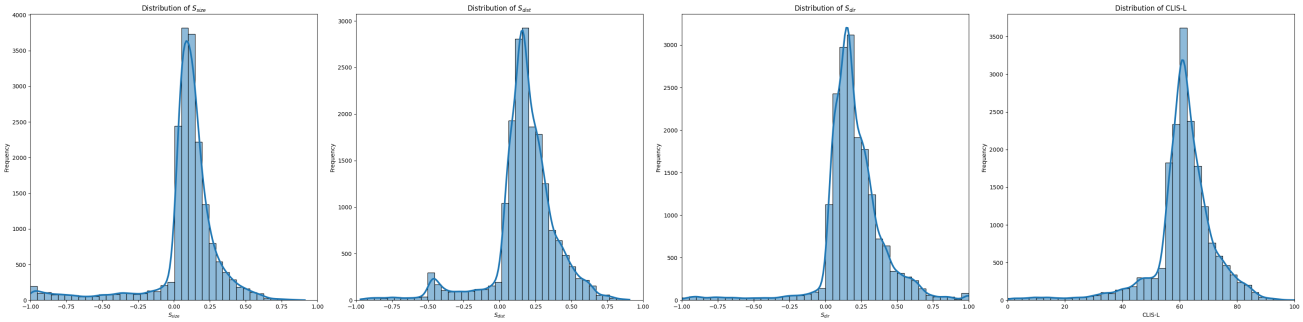

Figure 9. Score distributions of S_{size}, S_{dist}, S_{dir}, and CLIS-L.

Prompt for Layout Generator:
Task Description:
Your task is to generate a layout based on a detailed description. The layout is a list of json with 'object' and 'bbox'. 'object' refers to the object name in the prompt provided, while 'bbox' is formulated as [x,y,w,h], where "x,y" denotes the top left coordinate of the bounding box. "w" denotes the width, and "h" denotes the height. The bounding boxes should not go beyond the image boundaries. The six values "x,y,w,h,x+w,y+h" are all larger than 0 and smaller than 1.

You can refer to the following examples as references.
In-context learning examples for Layout Generator

Please provide a json format with Layout based on the following prompt.

6.3. Image Generator

We use InstanceDiffusion [74] as the image generator. We follow the default settings of SDXL used in InstanceDiffusion to refine synthetic images for our image generator. Regarding the number of synthetic images, we follow the approach used in StableRep [69] and SynCLR [68], generating four images for each scene graph.

6.4. Image Filter

For caption model F_C in Eq. 9, we adopt a pre-trained VLM, Qwen-VL [1], which has strong perception abilities. For F_{sim} function in Eq. 9, we prompt an LLM to assign a score based on text similarity. Unlike direct comparisons of text embeddings, LLMs can weigh different parts of the descriptions according to their significance. For instance, a mismatch in categories results in a lower score than a mismatch in attributes.

Prompt for Caption Model:
You are my assistant to evaluate the correspondence of the image to a given text prompt.
Briefly describe the image within 50 words. Focus on the objects in the image and their attributes (such as color, shape, texture), spatial layout, and action relationships.

Prompt for Similarity Compution:
You are an intelligent chatbot designed to evaluate the correctness of generative outputs for question-answer pairs.
Your task is to compare the predicted answer with the correct answer and determine if they match correctly based on the objects, and their actions, relationships. Here's how you can accomplish the task:
———
##INSTRUCTIONS:
- Focus on the objects mentioned in the description and their actions and relationships when evaluating the meaningful match.
- Consider synonyms or paraphrases as valid matches.
- Evaluate the correctness of the prediction compared to the answer.

Please Evaluate the following answer pair:

Correct Answer: answer
Predicted Answer: pred

Provide your evaluation in the JONSON format with the 'score' and 'explanation' key. The score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. The explanation should be within 20 words.

7. Deployment Details on Downstream Tasks

7.1. Templates for Multi-modal Downstream Tasks

Localization:
Question:
1. Where is the object described {attribute} located in the image in terms of the bounding box?
2. What is the location of object described {attribute} in terms of the bounding box?
3. Localize the object described {attribute} in terms of bounding box.
4. Provide a bounding box for the object described {attribute}.
5. Generate a bounding box for the object described {attribute}.
6. Describe the object located at {layout}.
7. Provide a caption for the object at {layout}.
8. What is at location {layout} in image?
Answer:
1-5: It is located at {layout}.
6-8: There is a {attribute}.

Attribute-binding:
Question:
1. What is the color of {obj}?
2. What color is the {obj}?
3. What color do you think the {obj} is?
4. Which color is the {obj}?
5. What is the number of {obj}?
6. What is the total count of {obj} in the image?
Answer:
1-4: {color}.
5-6: {number}.

Relation:
Question:
What is the relationship between the subject described {attribute1} and the object described {attribute2}?
Answer:
{subject} {relation} {object}.

We provide templates for constructing question-answer pairs for multi-modal downstream tasks. For perception, we design two types of tasks: localization and attribute-binding. Localization tasks necessitate that models pinpoint an object detailed in the instructions or, alternatively, describe an object situated at a specific location. Attribute-binding tasks require models to identify precise attributes of an object within a given location or give a precise number of the target object. For reasoning, we craft relation reasoning tasks. These tasks require models to deduce the relationship between a specified subject and object based on the provided description.

7.2. CLIS Settings

CLIS-L Computation. We detail CLIS-L computation as follows:

• Penalty Function. To filter out noise data identified by a low score in any of the three metrics, S_{size}, S_{dist}, and S_{dir}, we introduce a penalty function f and a score threshold t. The function f is a linear transformation that maps scores below t from 0 to t to a range of -1 to t.

• Weight. To balance the impact of the three metrics (size, distance, and direction) in Eq. 4, we apply Z-score normalization to each. The distribution of scores across these three metrics and CLIS-L is shown in Fig. 9.

• Percentile Operation. We use percentile operation in Eq. 4. We first compare the percentile operation with the average operation. Given that multiple reasonable layouts can correspond to the same description, not all layouts from the data priors P provide the necessary information for accurate assessment. For example, in the description 'a person holds an umbrella', it would be unreasonable to evaluate a synthetic layout where the umbrella is in the person's right hand using ground truth layouts from P where the umbrella is in the left hand. To compare these two operations quantitatively, we conduct experiments. We construct a test set of 10K samples generated by ACP, each containing two objects in the scene graph. We first swap the layouts of the two objects to determine if CLIS-L can assign a higher score to the original layout. Additionally, assuming that good layouts can produce better images, we compare the images selected by the two different CLIS-L calculation methods. Specifically, we use these two methods to independently select the top 25% highest-scoring data (2.5K examples) and calculate the FID score for the corresponding images. Table 8 shows that percentile operation in CLIS-L outperforms average operation. We then compare the percentile operation with the max operation. Since we use LLMs to construct data priors P, it may cause some errors in P. Thus, percentile operation is more robust against similar errors in synthetic layouts.

CLIS Distribution and Setting. For visual perception tasks, we emphasize image quality by applying a threshold to CLIS-I. Specifically, we set an instance-level threshold of 60. Images in which all instances fall below this threshold are excluded from the training set. Fig. 10 shows the original distribution of instance-level CLIS-I (left) and overall CLIS-I (middle), as well as the distribution of CLIS-I after filtering (right). The instance filtering ratio is approximately 50%, while the image filtering ratio is around 15%. For multi-model perception and reasoning tasks, we apply both a threshold for CLIS-I and an additional threshold of 50 for CLIS-L to ensure the generated layouts are reasonable. The

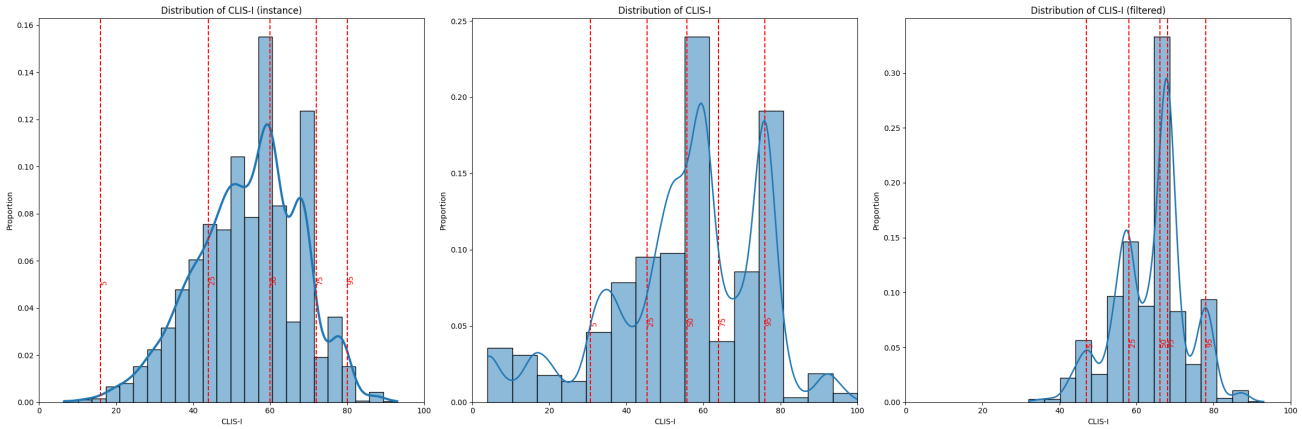

Figure 10. Score distribution of CLIS-I.

Table 8. Comparison of Max operation and Average operation in CLIS-L.

Operation	Accuracy↑	FID↓
Average	58.8	61.4
Percentile	61.0	55.5

CLIS-I threshold remains consistent with that used in visual perception tasks.

8. Details of Experiments Setup

8.1. Baseline Settings

Our specific baseline settings in experiments are as follows:

• Mask R-CNN baseline. We follow the same setup outlined in [19]. Specifically, we adopt ResNet-50 [23] with FPN [48] backbone, using the standard 1× training schedule.

• CenterNet2 baseline. We follow the setup outlined in [87]. Specifically, we use two configurations: 1) ResNet-50 with a 1× training schedule, and 2) Swin-B with a 4× training schedule. We employ the AdamW optimizer and utilize repeat factor sampling with an oversample threshold of 10^{-3}.

• Grounding-DINO baseline. We follow the setup outlined in [88]. Specially, we use the model pretrained on Objects365 [62], GoldG [35], GRIT [55], and V3Det [72] with Swin-T [51] as the backbone. The fine-tuning process uses the standard 1× training schedule. We use AdamW [52] optimizer with a weight decay of 0.0001. The initial learning rate is 0.00005, dropped by 10× at the 8th and 11th epochs.

• LLaVA-v1.5 baseline. We follow the setup outlined in [46]. We adopt a two-stage training process. For the LLM backbone, we adopt Vicuna-7B [10], Vicuna-13B, and LLama-3-8B [70]. We use an AdamW optimizer with a weight decay of 0. Pre-training for 1 epoch with a 1e-3 learning rate and batch size of 32, and fine-tuning for 1 epoch with a 2e-5 learning rate and a batch size of 16. The warmup ratio of the learning rate is 0.03.

• Stable Diffusion baseline. We use the v1-5 model weight from Huggingface [75].

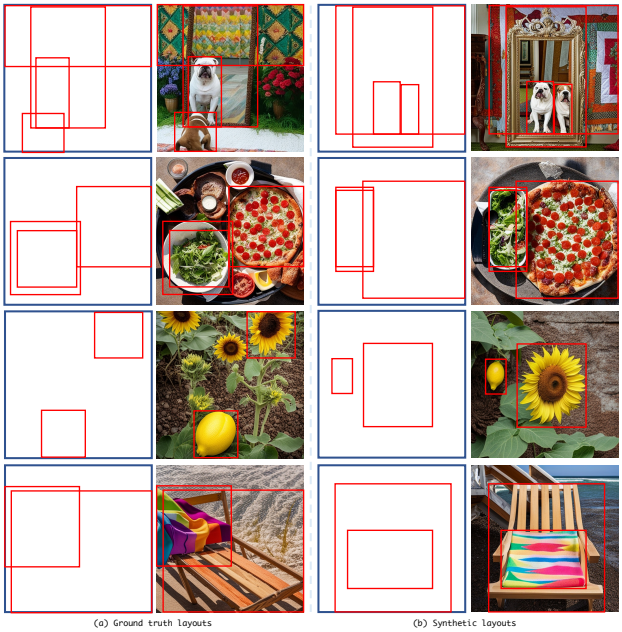

Figure 11. Comparison between ground truth layouts and synthetic layouts from our layout generator.

8.2. Training Settings

We augment the original training set with synthetic examples to co-train downstream models, while annotations for rare categories are excluded in the open-vocabulary setting.

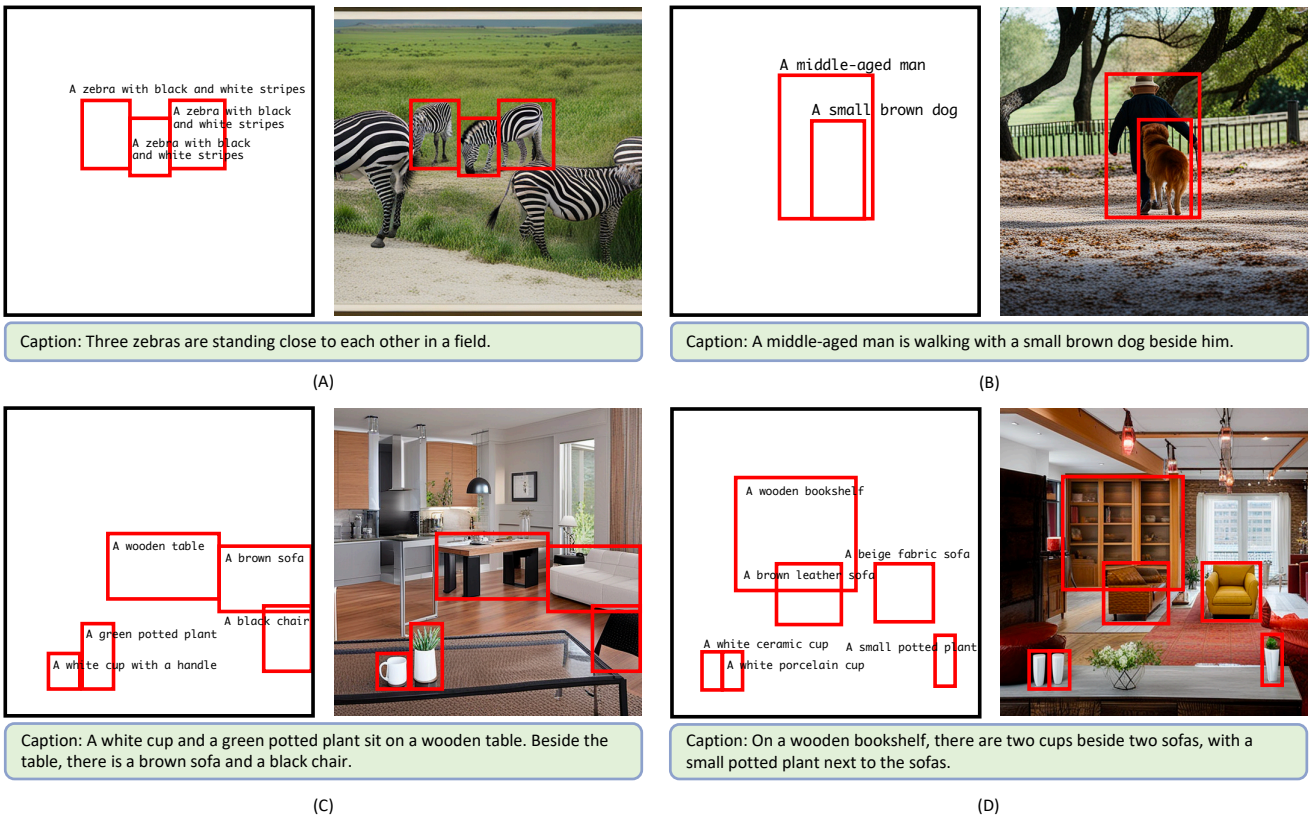

Figure 12. Error analysis of ACP. (A) Numerous objects and (B) overlapping objects for the image generator. (C)(D) complex object combinations for the scene graph generator.

8.3. Evaluation Protocols

For generative metrics, FID is computed with the Inception V3 [66]. We adopt a pre-trained YOLOv8m following [74] for YOLO score [41] and report the standard average precision (AP), which is averaged at different IoU thresholds (from 0.5 to 0.95) across categories.

8.4. Dataset Details

MS-COCO is a common detection dataset containing 80 categories with 118K training images and 5K validation images. In the open-vocabulary setting [3], MS-COCO can be divided into 48 base categories and 17 novel categories, excluding 15 categories without a synset in the WordNet hierarchy.

LVIS is a large vocabulary dataset with 1203 categories, featuring a long-tailed distribution of instances in each category. These categories can be divided into rare(337), common(461), and frequent(405) groups. LVIS training set contains 100K images, with an additional 20K images in the validation set.

The original instruction-following data mixture of LLaVA-1.5 is a total of 665K [46].

9. Limitations and Future Work

Sampling of initial object combinations and evaluating layouts necessitates data priors P, which are resource-intensive to construct. Currently, P is built from the COCO, LVIS, and Filter30K datasets. However, practical limitations in computational resources constrain our capacity to expand P, potentially impacting the accuracy of layout evaluation. Generating high-quality samples through ACP is also computationally demanding, with a portion of synthetic samples to be filtered out. A future research direction involves developing computation-efficient methods to generate high-quality samples or devising strategies to learn from low-quality samples efficiently.

Additionally, simple experiments presented in Table 8 and Fig. 1(a,b) indicate that better layouts contribute to improved image quality. Thus, another potential direction for future work is to use layout metrics to optimize computational resources in the generation process. We encourage more future studies focusing on the design of generation metrics.

10. Consistency with Human Preference

In Fig. 5, we present images generated from the same scene graph. The image quality consistently improves as the CLIS increases, confirming its alignment with human judgment. To comprehensively evaluate consistency with human preferences, we additionally carry out a user study with 20 subjects. Each subject is shown 40 pairs of images, with each pair generated from the same scene graph with different CLIS scores. The subjects are asked to evaluate the image pairs based on the following criteria:

- Q1. choose the image that has the best **visual** quality.
- Q2. choose the image that is better **aligned** with the annotation, including bounding boxes and text descriptions.

A total of 1535 responses are collected. The results show that samples with higher CLIS get 66.1% for Q1 and 94.7% for Q2. This indicates that higher CLIS aligns well with human judgments on visual quality and annotation alignment.

11. Efficiency-Effectiveness Analysis

ACP demonstrates its efficiency: (1) Detectors efficiently utilize synthetic samples from ACP. For instance, X-Paste uses 100K synthetic images, double the size of ACP's synthetic dataset. (2) Synthetic data from ACP is richly annotated with detailed object attributes and relationships, making it readily applicable to various downstream tasks. (3) ACP significantly reduces the cost of data generation compared to manual collection and annotation, particularly for rare categories.

12. Error Analysis

Synthetic errors may arise in large-scale generation due to:

- Scene Graph Generator. LLMs often struggle with rare or complex object combinations, leading to inaccurate layouts.
- Image Generator. Diffusion models frequently fail when objects overlap or when rendering a large number of objects.

As shown in Fig. 12, errors in (C) and (D) originate from the scene graph generator. When confronted with complex object combinations, LLMs may generate implausible layouts. For instance, in (C), the cup and plant should appear on the wooden table, and in (D), the two cups belong on the bookshelf. Errors in (A) and (B) arise from the image generator. Diffusion models tend to struggle when handling (A) numerous objects or (B) overlapping objects.

13. Visualization Results

13.1. Ablation Study on Layout Generator

We present visualizations of images generated using both ground truth layouts and synthetic layouts. As shown in Fig. 11, images generated with synthetic layouts exhibit comparable or even better to those generated with ground truth layouts. Notably, ground truth layouts tend to overlap more, leading to low-quality results from diffusion models. Furthermore, synthetic layouts are more likely to be centered in the images, which helps reduce the occurrence of distracting objects in the generated images.

13.2. Comparison with Other Metrics

CLIS-I. We provide visual results comparing CLIS-I with other metrics. Using the same scene graph from the previous generator, we produce images evaluated with CLIS-I and other metrics, such as CLIP and YOLO scores. As illustrated in Fig 13, CLIS-I demonstrates superior performance in both textual alignment and visual quality.

CLIS-L. We further present visual comparisons of CLIS-L with the spatial detection-based HRS metric [2], similar to those used in T2I-CompBench [31], which applies predefined rules to evaluate fix spatial relationships. To ensure that the relationships being evaluated are spatial and compatible with the HRS metric, we use the HRS spatial compositions benchmark [2]. As shown in Fig 14(A), CLIS-L aligns with the HRS metric in evaluating typical spatial relationships. Both assign high scores to accurate spatial layouts. Fig. 14(B) highlights the advantage of CLIS-L, which assigns low scores to unrealistic or inaccurate spatial layouts, demonstrating its superiority in filtering suboptimal cases. Notably, CLIS-L can also evaluate non-spatial layout relationships, further showcasing its versatility.

13.3. Synthetic Training Examples

Additionally, we showcase visualizations of our synthetic training samples in Fig. 15 and Fig. 16. By leveraging the extensive vocabulary of large generative models, we can produce high-quality training samples for rare categories. These training samples are closely aligned with their respective scene graphs, capturing both detailed attribute descriptions and complex relationships between multiple objects effectively.

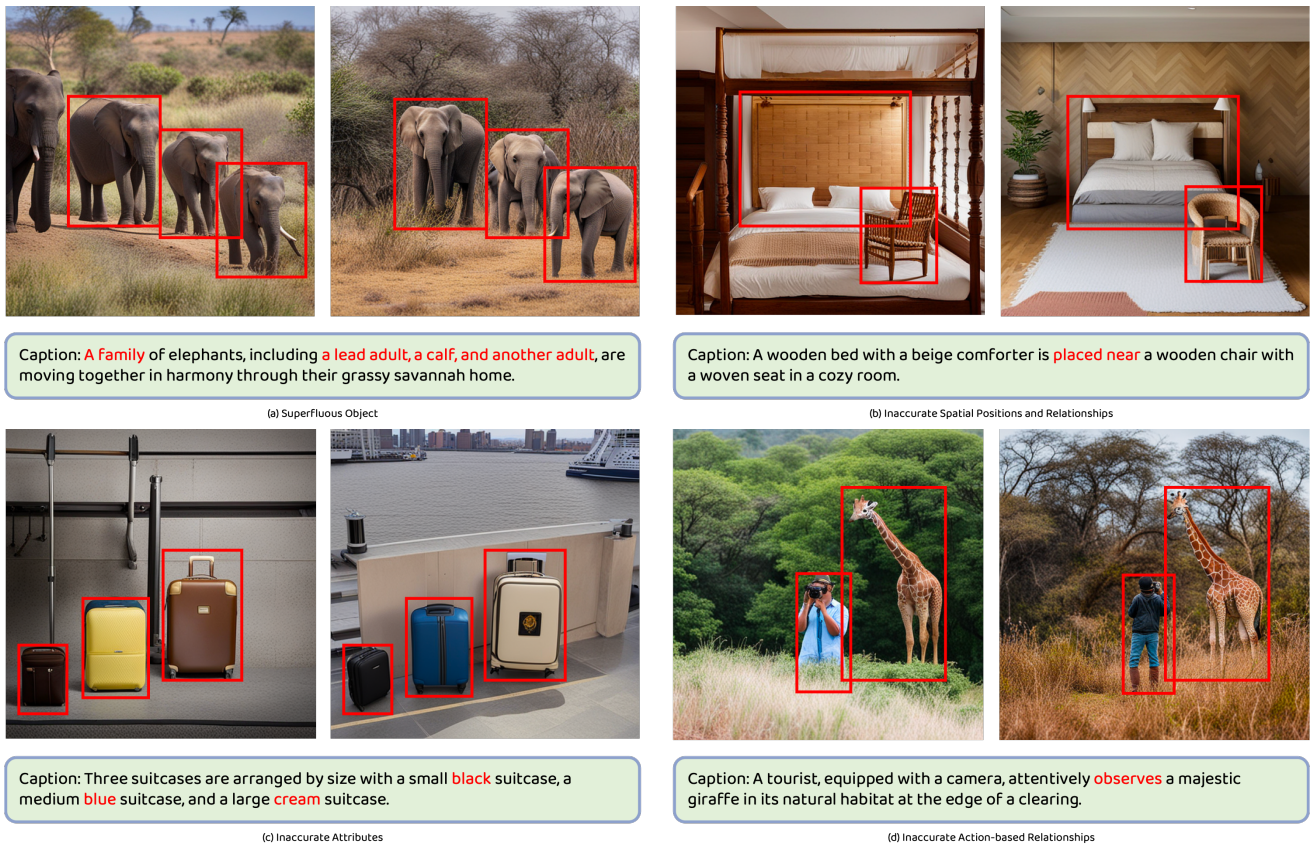

Figure 13. Comparison between CLIS-I and other prevalent metrics. Each pair of images is generated on the same scene graph, with CLIS-I favoring the right image in each pair. In (a) and (b), the CLIP score overlooks the extraneous elephant on the left and the inaccurate spatial arrangement between the chair and bed, respectively. For (c) and (d), the YOLO score fails to assess the detailed attributes or evaluate the semantic relationships between objects.

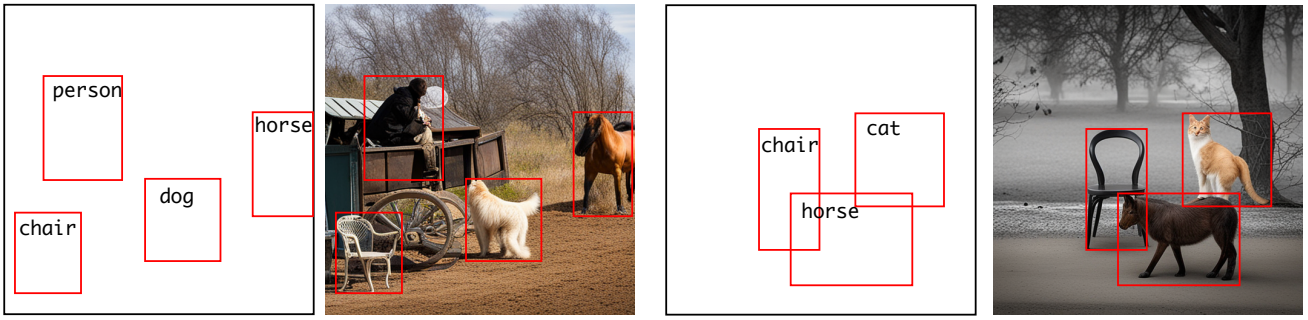

Figure 14. Comparison of CLIS-L and the HRS metric. (A) CLIS-L is consistent with the HRS metric in evaluating typical spatial relations. Both assign high scores to accurate spatial layouts. (B) CLIS-L provides additional filtering capability for problematic cases. For instance, the prompt 'A horse above a cat' is unreasonable in real-world scenarios. 'A cat over a person' is inaccurate as the cat should be positioned higher in the layout.

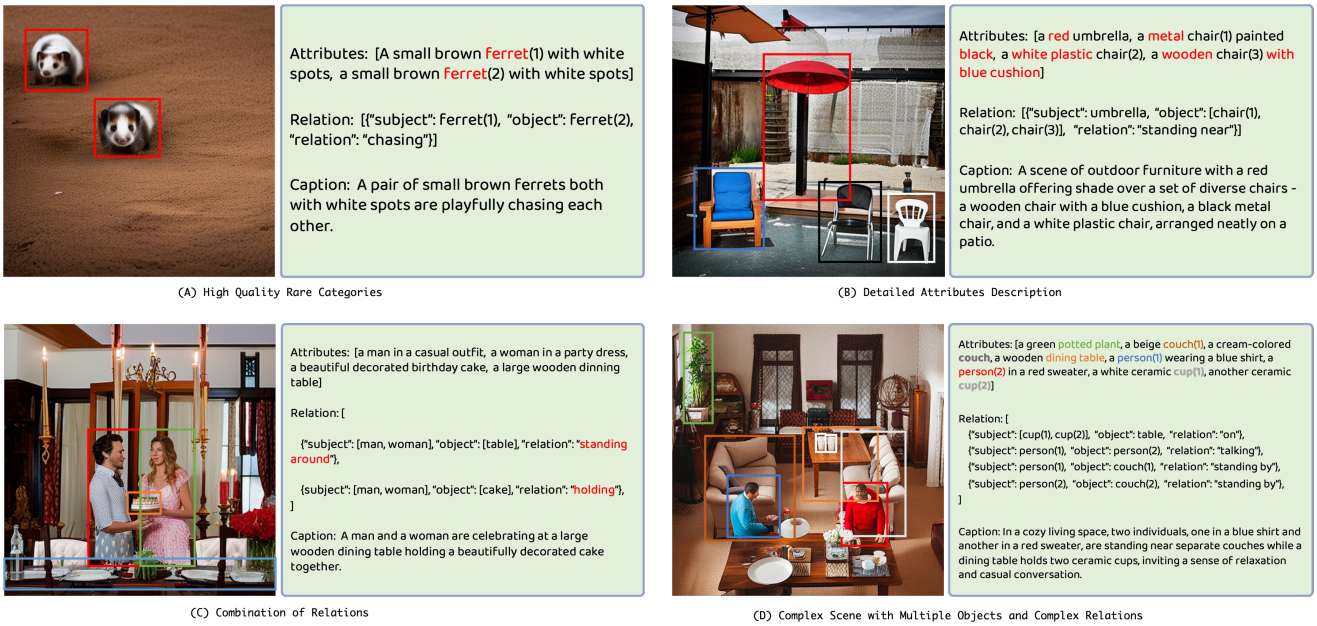

Figure 15. Synthetic training examples from ACP. In settings with imbalanced training data, such as long-tail scenarios, ACP can produce high-quality training examples for rare categories to mitigate this challenge. Additionally, ACP can generate diverse training samples with detailed attributes and relationships within complex scenes.

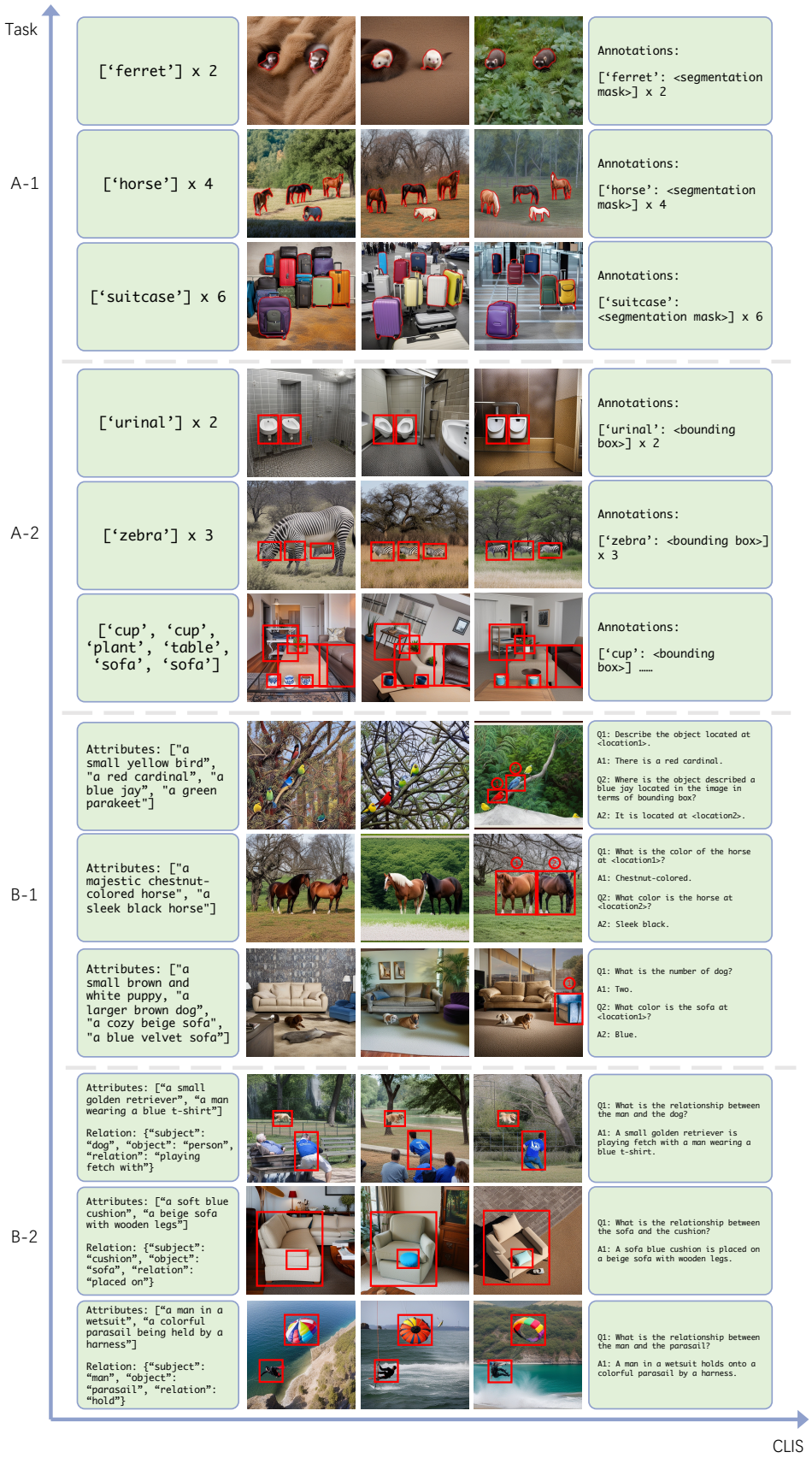

Figure 16. Synthetic training samples of various tasks from ACP. Tasks A-1 and A-2 correspond to Segmentation and Detection, respectively. Tasks B-1 and B-2 pertain to multi-modal perception and reasoning. Given the same input or scene graph on the left, the CLIS of the synthetic training samples increases along the x-axis, with final annotations on the right.