Appendix

Overview

A More Details of CAPability

A.1 Details of Dimension Design

We argue that multi-dimensional evaluation is significant to visual caption evaluation and is more comprehensive than previous work. So how to choose proper dimensions? We refer to existing VQA benchmarks [62, 63, 64, 65] and visual generation benchmarks [31, 32, 33]. VQA benchmarks usually design various types of questions to include multi-dimensional evaluation and analysis of MLLMs. For instance, MMBench [64] defines 20 ability dimensions, including attribute recognition, attribute comparison, action recognition, spatial relationship, physical property, OCR, object localization, image style, image scene, identity reasoning, *etc*. MVBench [64] covers 20 challenging video tasks including action, object, position, count, scene, pose, attribute, character, cognition, *etc*. Due to the flexible design of questions, VQA benchmarks can be naturally built with comprehensive dimensions. Different from the VQA task, the visual caption task does not require specific questions, but inspects the alignment of visual and textual information. Visual generation is the inverse task of visual captioning, as it requires models to generate specific visual content based on detailed textual descriptions. GenEval [31] designs 6 different tasks to evaluate text-to-image alignment, including single object, two object, counting, colors, position, and attribute binding. VBench [32] comprises 16 dimensions, including subject consistency, background consistency, object class, human action, color, spatial relationship, scene, style, *etc*. We follow their explored dimensions to design proper dimensions for visual captioning. Finally, we design 6 views, covering object, global, text, camera, temporal, and knowledge. The object-related view includes object category, object color, object

number, and spatial relation, the global-related view includes scene and style, the text-related view evaluates the OCR capability of captions, the camera-related view covers the camera angle and movement, the temporal-related view contains action and event, and we also design a view to evaluate the knowledge of MLLMs, *i.e.*, character identification.

We believe these dimensions contribute to a comprehensive visual caption benchmarking. However, it is undeniable that there may still be other dimensions that also contribute to caption evaluation. This phenomenon exists in all multi-dimensional benchmarks, but the purpose of our design is to find dimensions that are as comprehensive as possible and sufficient to differentiate model capabilities. As we design 12 dimensions, the evaluation is strong enough to evaluate models from various aspects. We also welcome the proposal of more constructive dimensions.

We explain each dimension in detail about what it represents here.

- **Object category.** This dimension measures the ability of whether models can give a correct description of a specific object in the image. The object is randomly selected from the image.
- **Object number.** Given a kind of randomly selected object existing in several numbers in an image or a video, this dimension measures the ability of whether models can count the object correctly. For videos, models should watch the whole video and dynamically count the number based on the camera.
- **Object color.** Given a kind of randomly selected object in an image, this dimension measures the ability of whether models can correctly describe the color.
- **Spatial relation.** Given two nearby objects in an image, this dimension measures the ability of whether models can correctly describe the spatial relationship of the two objects. We sample 500 images from our collected data, and sample another 500 images from CompreCap [27], with their spatial relationship descriptions.
- **Scene.** Given an image, this dimension measures the ability of whether models can obtain and tell the global scene of the image correctly.
- **Camera angle.** Given an image, this dimension measures the ability of whether models can obtain and tell the camera angle correctly when shooting the image.
- **OCR.** Given an image, this dimension measures the ability of whether models can recognize and tell the text appearing in the image correctly.
- **Style.** Given an image, this dimension measures the ability of whether models can obtain and tell the global style of the image correctly.
- **Character identification.** Given an image, this dimension measures the ability of whether models can recognize the character or the person in the image.
- **Action.** Given a video, this dimension measures the ability of whether models can recognize the action in the video. We use the video data of Dream-1K [28] and re-annotate the action from their annotations.
- **Camera movement.** Given a video, this dimension measures the ability of whether models can obtain and tell the camera angle correctly when recording the video. We search videos by ourselves and cut them into short clips, filtering complex movement composition. We only have simple camera movement in our data, but existing models still perform unsatisfactorily.
- **Event.** Given a video, this dimension measures the ability of whether models can tell a complete event in the video. We refer Dream-1K [28] to design this dimension, and we extract the events from their annotations. Different from other dimensions with atom-level elements, the event is usually composed of subjects and actions, which measures the temporal summarization ability of the model.

A.2 Explanation for One Represents All Strategy

"One represents all" is designed for object selection for object-related dimensions, text for OCR dimension, and action. We further build other dimensions related to attributes and relationships of objects (object color, object number, spatial relation) based on the selected objects. By aggregating results over a large number of samples with random pairing, we achieve statistical coverage across a

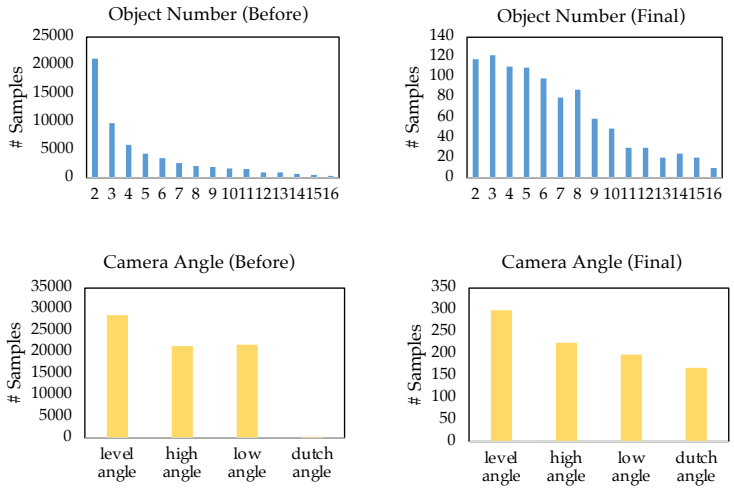

Figure A1: Two examples (object number and camera angle) of data distribution before data balance (pre-annotated) and after data-balanced selection (final human-annotated).

broad spectrum of relationships, granularities, and contexts. Therefore, our design ensures that both single-object properties and inter-object relationships are robustly evaluated at the benchmark level, while providing a practical and scalable way to balance annotation cost and comprehensive coverage.

We focus on keeping the randomness of element selection, thus covering the whole visual content in a statistical sense, based on the law of large numbers. Therefore, we can get the ability to evaluate the thoroughness of the generated captions by calculating the hit. The details about our efforts to ensure the randomness are as follows. 1) To minimize bias and keep randomness in the pre-annotation stage, we utilize multiple SOTA models (GPT-4o, Gemini-1.5-pro, Qwen-VL-Max) to generate candidate objects and take the union of their outputs. We then use code (Python's random library) to randomly select one object from this union. 2) During manual annotation, as humans tend to select the most obvious objects in the image, annotators are not allowed to reselect or change the target object if the pre-annotation is incorrect, thus mitigating the bias; they are only permitted to judge correctness and filter out incorrect samples. This ensures that neither model nor human bias influences which element is selected for annotation, thus keeping randomness.

A.3 Details about Human Subjectivity Controlling

We acknowledge that differences in human annotator judgment can introduce subjectivity, particularly in interpreting synonyms or resolving ambiguous cases. To reduce this, annotators are required to flag any samples for which they have low confidence. All low-confidence samples are then independently annotated by two additional annotators, and the final decision is determined by majority vote among the three annotations. We also have a spot check and verification process. For each dimension, we ask other annotators (or ourselves) to randomly select 20% of the samples for verification. If the annotation accuracy falls below 97%, we hold a meeting to highlight the incorrect annotations, revise all annotations for that dimension, and repeat this process until the annotation accuracy meets our requirements.

A.4 Details of Data Balance and Final Distribution

The purpose of data balance is to suppress the long-tail distribution, thus ensuring there are a certain number of samples of different difficulties in the benchmark. Fig. A1 shows two examples of the comparison of pre-annotated data distribution and final human-annotated data distribution. For object number and camera angle dimension, we first randomly sample nearly 75K samples and conduct model pre-annotation. The 75K samples consist of approximately 1/3 SA-1B [50], 1/3 COYO-100M [51], and 1/3 crawled from multiple public datasets and websites. For object number dimension, we select all samples with counting from 2 to 16 same objects within an image. As shown in Fig. A1 (left upper), the counting follows the long-tail distribution, there are fewer images

Table A1: The overlap among each dimension.

Obj. Cate.	Obj. Num.	Obj. Color	Spa. Rel.	Scene	Cam. Ang.	OCR	Style	Cha. Iden.	(D) Obj. Num.	Cam. Mov.	Act./Event
11.3%	10.8%	10.5%	2.5%	4.6%	4.6%	2.6%	0.5%	0%	0%	0%	0%

with more objects within an image. Therefore, we conduct data-balanced sampling. Specifically, we separately sample images for different counts, thus forcing the number of each count to be more balanced after the human correcting and filtering process. For camera angle dimension, the dutch angle data is rare, therefore we keep all dutch angle data, and sample the same number of the other three categories. After the human correcting and filtering, the number of these categories varies slightly. The situation in other dimensions is also similar.

As we consider the counts, categories, *etc*. For each dimension to conduct the data balance and not consider the data source (*i.e.*, from SA-1B, COYO-700M, or crawled by ourselves) during this process, the final data source distribution for dimensions of object category, object number, object color, scene, camera angle, OCR varies. For data of object-related dimensions (object category, object number, object color, spatial relation), global-related dimensions (scene and style), and a small part of camera angle and OCR dimensions, we collect the base data in a hybrid way. Our approach involves first performing pre-annotation for these dimensions on a large pool P_{all} of images (100K). We then filter out those samples that are not suitable for the corresponding dimension independently, donated as P_i^{pre} (nearly 40K - 80K), where i represents each dimension. After that, we conduct balanced sampling and manual annotation, resulting in approximately 1K samples per dimension, donated as P_i^{final}. For spatial relation, we directly choose 1/2 CompreCap [27] and 1/2 SA-1B images as SA-1B is more likely to contain high-resolution images with complex object relationship scenes. For style, we choose all realistic images from SA-1B, and crawl animated, special effect, and old-fashioned images by ourselves, all art-related images are from Wikipaintings [53]. For character identification, we use all images from the public dataset, *i.e.*, Wukong-100M [52] rather than crawling to ensure proper copyright. For dynamic object number, we directly use data from VSI-Bench [49]. For action and event, we directly use videos from Dream-1K [28]. We crawl all videos for camera movement dimension by ourselves, as there is little camera movement data in existing datasets. While there may be some overlap among pre-annotated P_i^{pre}, the average of overlapping samples in the final P_i^{final} is 3.95%, as shown in Tab. A1. We thank all public datasets and benchmarks, their excellent images, videos, and annotations provide much convenience for building our CAPability.

A.5 Details about Annotators

We employed 24 outsourcing annotators to complete the annotation task, including 14 female annotators and 10 male annotators. Their ages range from 24 to 30 years old, and all are from mainland China. Each annotator holds at least an associate's or bachelor's degree, demonstrating their expertise level. In total, the annotators labeled 23,824 samples, averaging approximately 993 samples per person. Among these, 3,323 samples were annotated by three different annotators, and the final result was determined through voting.

We specifically instructed annotators on how to handle such cases during the annotation process for each dimension. For example, in the color dimension, if the main object is primarily one color with minor secondary colors (such as a white airplane with some painted markings), only the primary color should be annotated. If the object exhibits multiple primary colors, annotators were asked to label up to three colors if the object can be clearly described within this limit; otherwise, the sample was filtered out. As a result, each object in our dataset is annotated with at most three colors.

A.6 Ethical Impact

The geographic and cultural backgrounds of the data are diverse. For data obtained from other public datasets, the distribution of potential bias in our benchmark largely follows that of the original datasets, as we performed random sampling. For the data we collected ourselves, there is a certain degree of cultural bias towards mainland China and East Asian cultures, although mainstream Western

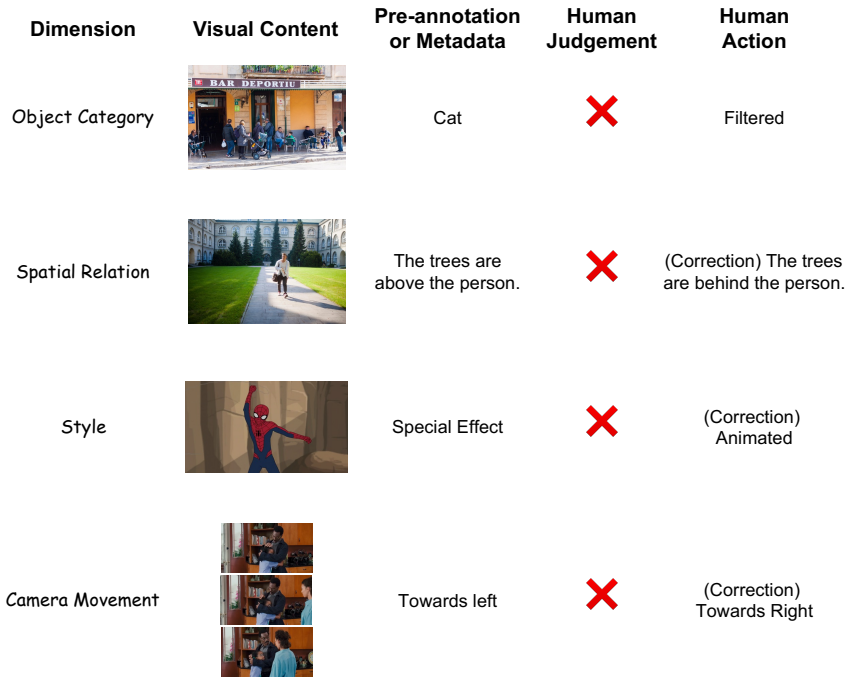

Figure A2: Bad case examples of pre-annotation or metadata, human annotators filter or correct the wrong pre-annotation or metadata one by one.

cultures are also represented to some extent. However, coverage of cultures from other regions is relatively limited.

A.7 Discussion about the Designed Metrics

We define two metrics, *precision* and *hit*, to evaluate both the correctness and thoroughness. The *hit* is similar to the meaning of *recall*, which measures how many visual elements in the image/video can be described correctly. However, the *recall* is usually related to the TP/FP/FN/TN system [66], which is inconsistent with our evaluation situations. To avoid ambiguity and misunderstanding, we name our metric as *hit* rather than *recall*.

We give the analysis from the TP/FP/FN/TN perspective here. The inconsistency comes from the conflict of the situation definition. In our pipeline, there are three situations when evaluating: 1) MIS, 2) COR, 3) INC, but the TP/FP/FN/TN system does not define the situation of MIS. In the TP/FP/FN/TN system, the *precision* and *recall* are defined as:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{A1}$$

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{A2}$$

The existence of MIS leads to the ambiguity of the definition of FN and FP. If there is no existing MIS, we can only calculate the accuracy without considering *precision* or *recall*. Now we try to analyze with MIS. Based on the definition of TP, we can map our COR to TP with no doubt, as it correctly predicts the answer. If we want to calculate the *precision*, we can map our INC to FP, as it wrongly predicts the answer. Therefore, the precision can be calculated by Eq. 1 or Eq. A1, which is consistent. When we consider recall, $TP+FN$ should be the number of all ground truths, as there are no negative samples (TN) in our annotation, which means $TP+FN = S(\text{COR}) \cup S(\text{INC}) \cup S(\text{MIS})$. This leads to S(INC) being included in both FP and FN. As the TP/FP/FN/TN system does not define the MIS, the TP/FP/FN/TN-based *precision* and *recall* cause contradiction. Therefore, we name our metric of $S(\text{COR})/(S(\text{COR}) \cup S(\text{INC}) \cup S(\text{MIS}))$ as *hit* rather than *recall* to avoid ambiguity as it does not fit the TP/FP/FN/TN-based definition.

Figure A3: Examples of visual content and annotations for each dimension. We outline some visual elements by the red box in the image or video to make them easier to identify.

In addition to the TP/FP/FN/TN system, there are also other ways to define *precision* and *recall*. MUC-7 [67] defines the *precision* and *recall* with the situation of MIS. Apart from COR, INC, MIS, which own the same meaning as ours, MUC-7 also defines SPU, which represents the number of spurious, and equals 0 in our situation. MUC-7 defines the *precision* and *recall* as follows:

$$\text{Precision} = \frac{S(\text{COR})}{S(\text{COR}) + S(\text{INC}) + S(\text{SPU})}, \tag{A3}$$

$$\text{Recall} = \frac{S(\text{COR})}{S(\text{COR}) + S(\text{INC}) + S(\text{MIS})}. \tag{A4}$$

Based on this kind of definition, our *hit* equals the "*recall*".

However, as the TP/FP/FN/TN system is too famous and standard to define the *precision* and *recall*, we finally decide to use the "*hit*" rather than "*recall*" to avoid misunderstanding.

A.8 Benchmark Examples

Examples of human annotation process. We show some visual cases of the human annotation process in Fig. A2. All of examples are with wrong pre-annotation or metadata. Human annotators check the pre-annotation/metadata one by one and filter/correct the mistakes for each dimension.

Figure A4: Examples of visual content and converted QA annotations for each dimension. The visual content is the same as Fig. A3. We outline some visual elements by the red box in the image or video to make them easier to identify.

Examples of annotations. We show some visual cases with our annotations in Fig. A3. We outline some visual elements by the red box in the image or video to make them easier to identify. We collect our data from various sources, and we crawled some visual content from the Internet by ourselves, ensuring diversity.

Image

```
IMAGE_PROMPT = "Please describe the image in detail. Your description should follow these rules:\n"\
"a) You should describe each object in the image in detail, including its name, number, color, and spatial relationship between objects.\n"\
"b) You should describe the scene of the image.\n"\
"c) You should describe the camera angle when shooting this image, such as level angle, high angle, low angle, or dutch angle.\n"\
"d) You should describe the style of the image, such as realistic, animated, special-effect, old-fashioned and so on.\n"\
"e) If there are any texts in the image, you should describe the text content.\n"\
"f) If you know the character in the image, you should tell his or her name.\n"\
"Directly output your detailed description in a elaborate paragraph, instead of itemizing them in list form. Your description: "
```

Video

```
VIDEO_PROMPT = "Please describe the video in detail. Your description should follow these rules:\n"\
"a) You should describe each events in the video in order, especially focusing on the behavior and action of characters, including people, animals.\n"\
"b) You should describe each object in the video in detail, including its name, number, color, and spatial relationship between objects.\n"\
"c) You should describe the scene of the video.\n"\
"d) You should describe the camera movement when shooting this video, especially the direction, such as pan left, track right, tilt up, boom down, zoom in, dolly out, and so on.\n"\
"e) You should describe the style of the video, such as realistic, animated, special-effect, old-fashioned and so on.\n"\
"f) If there are any texts in the video, you should describe the text content.\n"\
"g) If you know the character in the video, you should tell his or her name.\n"\
"Directly output your detailed description in a elaborate paragraph, instead of itemizing them in list form. Your description: "
```

Figure A5: The image prompt and video prompt for all models when inferring captions.

Examples of converted QA pairs. As we directly annotate the visual elements in the image or video rather than the caption sentence, we can easily convert our annotation into the format of question-answer (QA) pairs, and we name it as CAPability-QA. We use CAPability-QA to evaluate the QA accuracy and the *know but cannot tell* ($K\bar{T}$) metric. In Fig. A4, we also show the same visual cases as Fig. A3 for each dimension with converted QA format. Most of the dimensions are converted to the format of a multiple-choice QA task with several options, and the object color, OCR, and character identification dimensions are designed as open-ended QA tasks.

B More Experimental Analysis

B.1 Implementation Details

We use 4x80G GPUs to run all open-sourced model inference. We use `transformers` to deploy LLaVA-OneVision, InternVL2.5, VideoLLaMA3, and NVILA, use `vLLM` to deploy Qwen2VL and Qwen2.5VL, as their official repositories suggested. For all our evaluated model, we follow their official configurations to run the inference. We set the temperature of all open-source models to 0, while keeping the default for closed-source APIs. All maximum output token length is set to 8192. We list the configurations as follows.

LLaVA-OneVision. We set `anyres-max-9` for image, and uniformly sample 32 frames for video.

Qwen2VL and Qwen2.5VL. We keep the default minimum and maximum image pixels in package `qwen_vl_utils`, which is $4 * 28 * 28$, and $16384 * 28 * 28$, respectively. We also keep default video settings, the fps is set to 2.0, the maximum frames are 768, the minimum video pixel is $128 * 28 * 28$, and the maximum video pixel is $768 * 28 * 28$.

InternVL2.5. We use the official video and image process function and uniformly sample 32 frames for video.

VideoLLaMA3. We use image model for image dimensions and video model for video dimensions. The fps is set to 1, and the maximum frames are 180 for videos.

NVILA. We use the official image and video process function in `VILA` repository, and uniformly sample 8 frames for videos, as suggested in the official config.

GPT-4o. Due to the maximum frame number limits of GPT API, we uniformly sample 50 frames for videos, and keep the original spatial size of both images and videos, sending them to the API server.

Gemini-1.5-pro and Gemini-2.0-flash. As Gemini API supports video, we directly send the original image and video to the API server. For very few videos with too large file size, we downsample the fps to 3, and send the downsampled video to the API server for connection stability.

Object Number

```
object_number_user_prompt = "Given an image caption and the number of an object with format {object: number} as follows:\n"\
f"Image Caption: {caption}\n"\
f"Object Number: {{{object_category}: {object_number}}}\n"\
f"Please analyze the image caption. Determine whether the provided object number is correctly described in the caption, and explain why. You may need to count in the caption to determine how many the provided objects it describes.\n"\
"Give score of 0 if the caption does not mention the specific number of provided object (including the use of words such as 'some' and 'various' in the caption rather than giving specific numbers) or not mention the provided object. Give score of 1 if the caption describes the object number correctly. Give score only of -1 if the caption gives the wrong number.\n"\
"Output a JSON formed as:\n"\
"{\"object_number\": \"copy the provided {object: number} here\", \"score\": \"put your score here\", \"reason\": \"give your reason here\"}\n"\
"DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only output the JSON. Do not add Markdown syntax. Output:"
```

Camera Movement

```
camera_movement_category_explains = [
"left: the camera angle swings left (pan left), or the camera moves left (track left)",
"right: the camera angle swings right (pan right), or the camera moves right (track right)",
"up: the camera angle swings up (tilt up), or the camera moves up (boom up)",
"down: the camera angle swings down (tilt down), or the camera moves down (boom down)",
"in: camera pushes toward the subject (dolly in), or enlarges the frame (zoom in)",
"out: camera moves away the subject (dolly out), or expands the visible area, makeing the subject appear smaller (zoom out)",
"fixed: camera is almost fixed and does not change",
]
camera_movement_categories = [c.split(":")[0] for c in camera_movement_category_explains]
camera_movement_user_prompt = "Given a video caption, your task is to determine which kind of camera movement is included in the caption.\n"\
f"Video Caption: {caption}\n"\
f"Please analyze the video caption and classify the descriptions of camera movement into the following categories: {camera_movement_categories}\n"\
f"Here are the explanations of each category: " + '\n'.join(camera_movement_category_explains) + "\n"\
"If the caption explicitly mentions one of the above camera movement categories, write the result of the category into the 'pred' value of the json string. Note do not infer the camera movement categories from the whole caption. You should only search the descriptions about the camera movement. If there is no description of the camera movement in the video caption or the description does not belong to any of the above categories, write 'N/A' into the 'pred' value of the json string.\n"\
"Output a JSON formed as:\n"\
"{\"pred\": \"put your predicted category here\", \"reason\": \"give your reason here\"}\n"\
"DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only output the JSON. Do not add Markdown syntax. Output:"
```

Figure A6: Two prompt examples for different types of evaluation sub-tasks. The example of object number represents dimensions with open-ended descriptions, and the example of camera movement represents the dimensions with specific categories.

Table A2: The referring ratio of all models, which only reflects the referring ratio of each dimension without considering the accuracy.

Methods	Obj. Cate.	Obj. Num.	Obj. Color	Spa. Rel.	Scene	Cam. Ang.	OCR	Style	Cha. Iden.	(D) Obj. Num.	Act.	Cam. Mov.	Event	Avg.
LLaVA-OV-7B	96.6	33.9	60.8	54.7	86.7	79.7	73.6	98.8	5.0	34.5	92.2	62.4	30.9	62.3
Qwen2VL-7B	97.6	30.2	56.8	51.8	88.7	98.0	77.0	99.9	4.8	20.0	94.0	72.0	31.7	63.3
NVILA-8B	97.0	34.3	64.9	52.7	85.3	85.1	74.5	98.1	7.4	22.1	80.6	48.6	21.5	59.4
InternVL2.5-8B	97.0	38.1	60.9	55.2	87.3	100.0	84.6	100.0	20.3	22.7	91.4	94.8	32.9	68.1
VideoLLaMA3-7B	95.1	34.0	62.5	56.4	85.6	93.1	74.7	98.5	5.0	9.2	96.5	83.1	34.5	63.7
Qwen2.5VL-7B	96.7	26.8	63.7	55.3	89.4	99.2	86.7	100.0	11.3	20.9	92.5	98.6	35.0	67.4
LLaVA-OV-72B	95.8	35.2	63.1	54.1	87.4	93.3	74.9	99.5	11.0	31.2	92.6	69.0	31.7	64.5
Qwen2VL-72B	97.4	35.8	64.3	56.9	89.4	99.6	83.0	100.0	6.8	21.6	93.5	77.1	34.7	66.2
InternVL2.5-78B	97.2	41.7	65.3	57.0	86.7	100.0	85.5	100.0	21.3	25.7	88.2	63.3	28.8	66.2
Qwen2.5VL-72B	95.6	43.3	69.2	62.3	90.7	100.0	91.8	100.0	31.4	24.4	94.8	99.4	38.9	72.5
GPT-4o (0806)	96.0	44.5	73.5	61.6	88.2	100.0	88.8	100.0	35.1	29.4	93.4	99.4	44.5	73.4
Gemini-1.5-pro	96.1	55.3	77.0	69.0	88.1	99.9	91.4	100.0	67.5	48.9	90.5	100.0	48.6	79.4
Gemini-2.0-flash	96.1	39.0	67.2	58.2	87.5	100.0	93.2	99.9	46.2	30.4	92.0	99.6	44.6	73.4

B.2 Prompts of Inference and Evaluation

Inference prompt. We use the same prompts for all models to produce the visual captions. The image prompt and video prompt are shown in Fig. A5. To decrease the inference difficulty, we prompt the models to output the information of all our designed dimensions with a detailed caption. Despite this, the models still show a huge difference in the hit rate of each dimension, which may be due to the variety of training data related to the caption.

Evaluation prompt. As we divide the evaluation of dimensions into two types: 1) dimensions with specific categories (*i.e.*, style, camera angle, and camera movement), 2) dimensions with open-ended descriptions. Therefore, we design two kinds of templates for evaluating, and fine-tune them within each dimension. In Fig. A6, we take the object number dimension and camera movement dimension as examples, to show our prompts for evaluation. For dimensions with specific categories, we ask GPT-4-Turbo to extract the key information and classify the caption into our pre-defined categories or the 'N/A' class. The correct classification is considered positive, the wrong one as negative, and the 'N/A' result is considered a miss. For dimensions with open-ended descriptions, we ask GPT-4-Turbo to directly compare the annotations and the caption, and give out the result of positive, negative, or miss with reasons.

B.3 More Experimental Results

Referring ratio among all models. Apart from the *precision* and *hit*, we can also report another metric, *referring ratio*, which represents the referring ratio about the dimension in visual caption and

Table A3: The average metric of image dimensions and video dimensions.

Models	LLaVA-OV-72B	Qwen2VL-72B	InternVL2.5-78B	Qwen2.5VL-72B	GPT-4o (0806)	Gemini-1.5-pro	Gemini-2.0-flash
image precision	81.4	82.2	78.2	80.7	84.4	81.1	84.6
video precision	59.5	62.9	55.6	65.0	67.6	68.7	67.3
image hit	55.3	57.5	57.9	62.1	65.2	67.7	64.5
video hit	26.9	29.2	22.4	33.7	36.8	43.9	37.6

Table A4: The PPL results of each generated caption by different models.

PPL models	LLaVA-OV-72B	Qwen2VL-72B	InternVL2.5-78B	Qwen2.5VL-72B	GPT-4o (0806)	Gemini-1.5-pro	Gemini-2.0-flash
Qwen3 PPL	3.56	4.60	5.13	5.01	8.64	8.45	8.16
LLaMA3.1 PPL	4.89	6.70	7.75	7.54	11.29	10.76	10.68

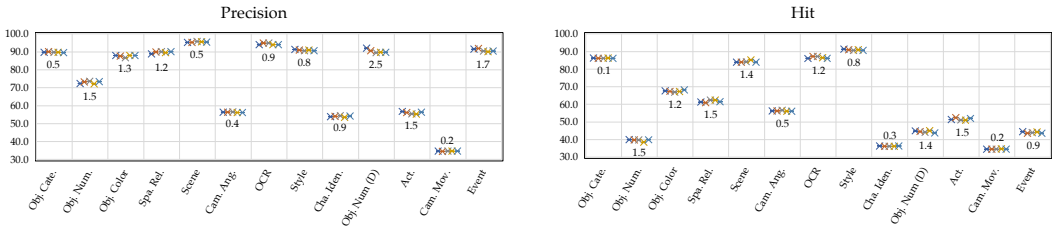

Figure A7: The evaluation of repeating 5 times for Gemini-1.5-pro captions. We tag the fluctuation range beside the data point.

can be calculated as:

$$\text{Referring Ratio} = \frac{|S(\text{COR}) \cup S(\text{INC})|}{|S(\text{ALL})|}. \tag{A5}$$

This metric only considers the pure thoroughness of the caption in each dimension without considering the accuracy.

We report the *referring ratio* in Tab. A2. For example, it is considered a reference if the caption mentions any object for the object category dimension, or mentions any angle information for the camera angle dimension, but for the object number or color dimension, it is only considered a reference if the caption mentions any number or color information of the correct object. We find the referring ratio seems to increase as the size of models increases, which may be due to more knowledge and stronger instruction following ability for larger models. Among all dimensions, the referring ratio of character identification performs the worst, the existing models prefer to keep silent as they usually cannot recognize the person and character well. The closed-source models would be more likely to reveal the names of characters, and we guess this may be due to stronger knowledge and more diverse training data.

Metric analysis between different modalities. Tab. A3 shows the metric analysis between image dimensions and video dimensions. Across all models, performance on image dimensions is substantially higher than on video dimensions, for both precision and hit metrics. Even the best-performing models (GPT-4o, Gemini series) show nearly 20% or greater drop in precision/hit from image to video dimensions. For example, all models perform well on object category, scene, OCR, and style for both precision and hit, but all models cannot achieve a satisfactory level on all video dimensions for hit. This shows that time series modeling and multi-frame information integration are still the main challenges of current MLLMs. It is true that some dimensions may be inherently more difficult to represent in video than in pictures. This may be due to the following two reasons: 1) There is more redundant content with more visual tokens input into MLLMs, handling the long sequence is likely more difficult than a shorter one. 2) Temporal change and movement should be considered for video dimensions, which may lead to confusion for models to recognize. Nevertheless, images can also represent the strength of videos in this dimension to a certain extent (such as OCR), because video understanding also requires a good spatial understanding ability as a prerequisite. The video dimensions we designed are more focused on the challenge of temporality. In the future, we will consider introducing more video data to fully evaluate the video's ability in the spatial dimensions.

The naturalness and coherence analysis. In the current landscape of MLLMs, the naturalness and coherence of generated captions are generally no longer a sufficient or even primary differentiator of

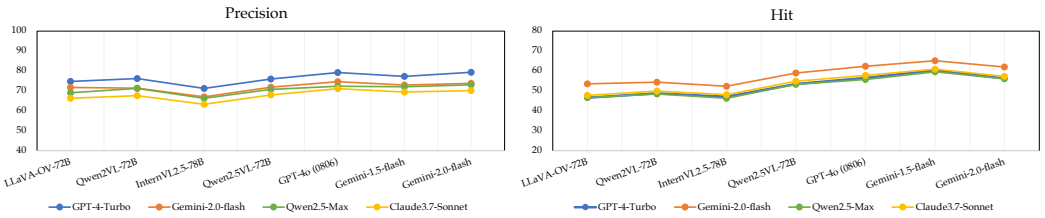

Figure A8: The stability analysis with three different evaluation models on 7 MLLMs' captions. The results on all metrics show a high degree of consistency.

model performance. Linguistic quality is more considered for NLP language models, and modern LLMs consistently generate grammatically correct and fluent sentences [68]. As MLLMs are built from LLMs, the main challenge and research focus of MLLMs has shifted toward evaluating the factual accuracy and relevance of the generated captions, rather than their linguistic quality. For example, the papers of MLLMs do not report any metrics about the naturalness and coherence, such as PPL [19, 24], modern benchmarks for MLLMs do not measure the naturalness and coherence either [69, 65]. To support the view that the naturalness and coherence of generated captions are not the primary challenges, we also further calculate perplexity (PPL) of the generated captions of each model. PPL is a fundamental metric for language models that gauges how "surprised" the model is by a given text, thus reflecting their naturalness and fluency, the lower PPL means better coherence. Specifically, we conduct qwen3-32B and LLaMA-3.1-8B-Instruct to calculate the PPL of generated captions from these models, as shown in Tab. A4. (There seems to be some gap between the closed-source APIs and open-source models. This may be because the training data distribution among open-source models is more similar to Qwen3 and LLaMA3.1 than closed-source APIs.) Among closed-source APIs, the PPLs of them are similar to or lower than GPT-4o. Among open-source models, the PPLs of them are similar to or lower than Qwen2.5VL-72B. Based on the common sense of GPT-4o and Qwen2.5VL-72B can generate coherent and human-like sentences, we can draw a conclusion that the naturalness and coherence are not the main challenge for all these models.

Evaluation stability. To validate the stability and robustness of our GPT-4-Turbo-based evaluation method, we take the inferred caption of Gemini-1.5-pro as the example, run our evaluation 5 times, and the result is shown in Fig. A7. We tag the fluctuation range, *i.e.*, the difference between the maximum and minimum scores, besides the data point. Fig. A7 shows our strong stability, and our average range of precision and recall are 1.1% and 1.0%, respectively. This demonstrates the reliability and interpretability of our evaluation method, which matches annotated elements in the generated captions. Moreover, we introduce three other models, Gemini-2.0-flash, Qwen2.5-Max, and Claude3.7-Sonnet to replace the GPT-4-Turbo in our evaluation pipeline. All of them are the most popular and powerful SOTA language models. We re-run the evaluation process with new evaluation models with the same evaluation prompts. The average result of evaluations for closed-source models and open-source 72B models is shown in Fig. A8. Though the judgment criteria for each model may be slightly different, leading to only a slight difference in the scores, the evaluation results show high consistency as they keep the same rank of these MLLMs as GPT-4-Turbo. This further demonstrates the high reliability, interpretability, and stability of our evaluation methods.

B.4 Visualization of Inference and Evaluation.

In Fig. A9 and Fig. A10, we visualize the inferred caption of Gemini-1.5-pro, GPT-4o (0806), and Qwen2.5VL-72B in object number dimension and camera angle dimension. In Fig. A9, the annotation of the given image is 7 pig trotters. Gemini-1.5-pro refers to the correct number of pig trotters, and we thus give it a positive. GPT-4o recognizes the trotters in the image, but counts with a wrong number, 6, and we thus give it a negative. As for Qwen2.5VL-72B, it says there are 7 chicken thighs in the image, recognizing the wrong object category. However, this is the dimension of the object number, and therefore we only evaluate the correctness and thoroughness of the number, without considering the categories of objects. As Qwen2.5VL-72B does not mention the pig trotters, we give it a miss. In Fig. A10, the annotation of the given image is dutch angle, which means the lens has a certain angle of deflection along the central axis, making the horizon crooked. Gemini-1.5-pro says the image is taken from a medium-high angle, and we classify it into the high angle category, thus

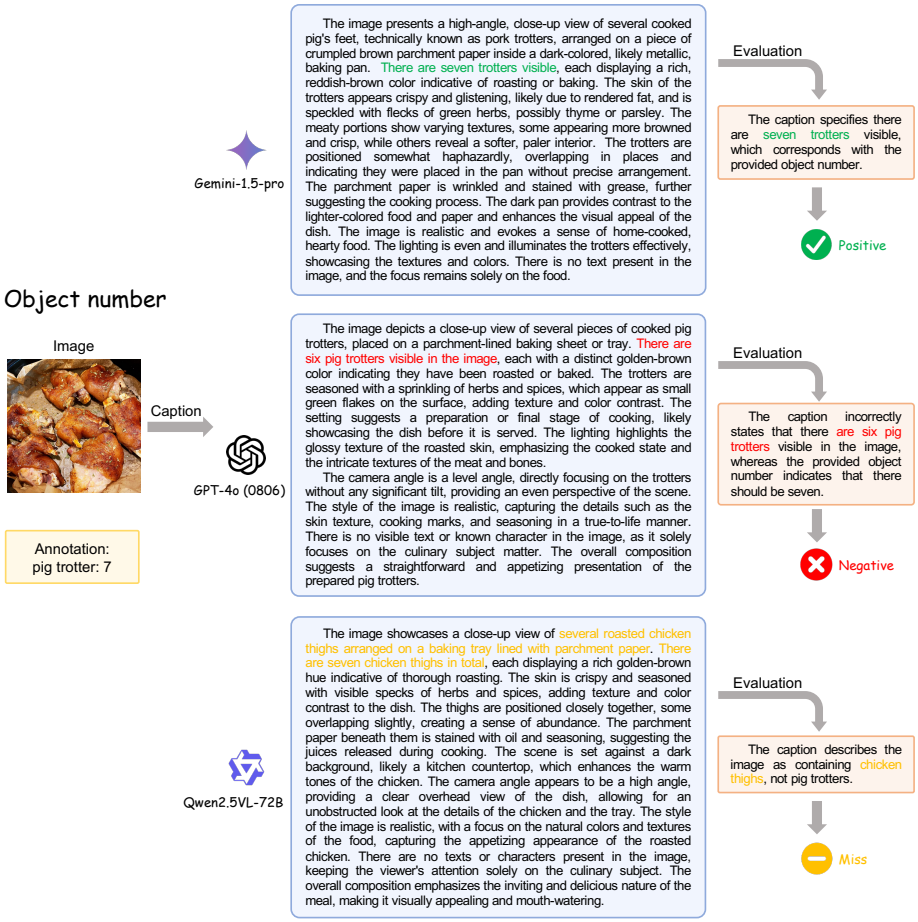

Figure A9: Examples of inference and evaluation on object number dimension. We select the inferred caption from Gemini-1.5-pro, GPT-4o, and Qwen2.5VL-72B as instances.

negative. GPT-4o explicitly points it out as a subtle dutch angle, thus is classified into the dutch angle category, which is positive. Qwen2.5VL-72B describes the image shot from a slightly elevated angle, and it appears to be a level angle, which is also negative. These two figures show our evaluation pipeline, which is precise and reliable.

C Copyright and License

CAPability comprises data from SA-1B, COYO-700M, Wukong-100M, Wikipaintings, VSI-Bench, CompreCap, Dream-1K, and uses some craweled data from inner retrieval system, each governed by its own licensing terms:

- **SA-1B**: SA-1B Dataset Research License[1]
- **COYO-100M**: CC-BY-4.0 License
- **Wukong-100M**: CC-BY-NC-SA-4.0 License
- **Wikipaintings**: BSD 2-clause License
- **VSI-Bench**: Apache-2.0 License
- **CompreCap**: CC-BY-4.0 License
- **Dream-1K**: Apache-2.0 License

[1] https://ai.meta.com/datasets/segment-anything-downloads/

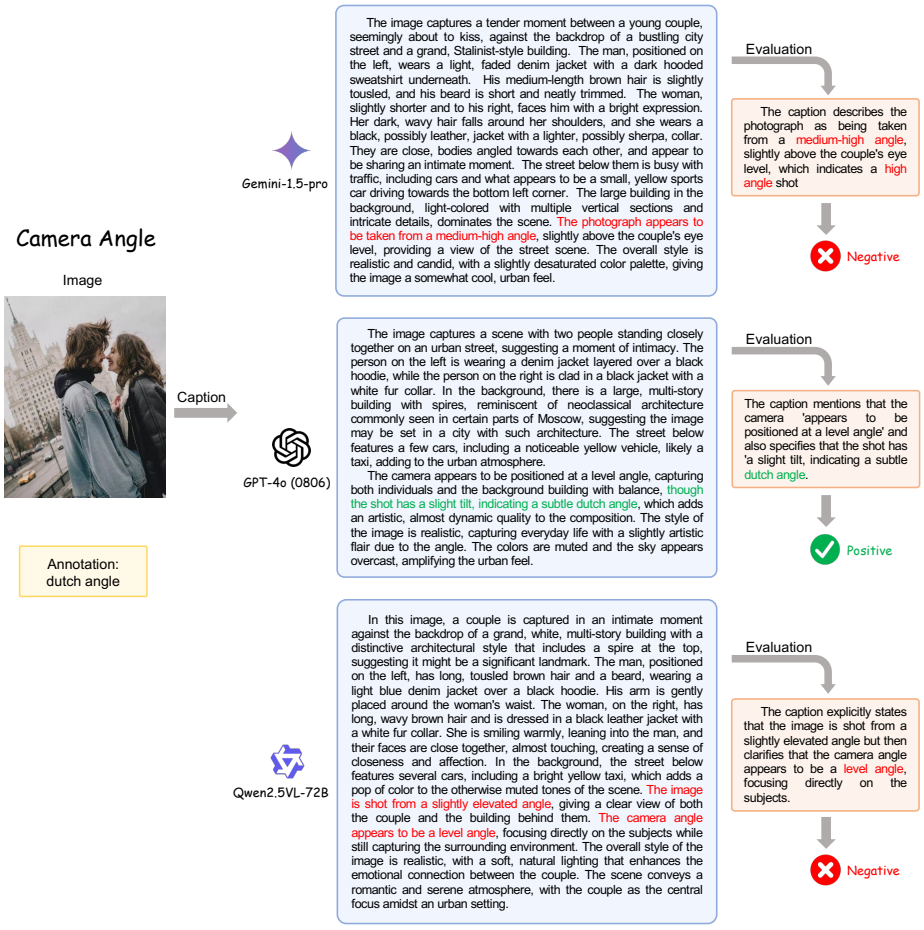

Figure A10: Examples of inference and evaluation on camera angle dimension. We select the inferred caption from Gemini-1.5-pro, GPT-4o, and Qwen2.5VL-72B as instances.

- **Craweled data**: Our craweled data are all retrieved from inner multi-modal retrieval system, which contains various public datasets, and visual contents from websites with CC0 license.

D Limitations

Different from multi-choice VQA benchmarks which evaluate models by definite and explicit choice accuracy, our CAPability is a visual caption benchmark, which still depends on LLMs for evaluation. Therefore, it is still hard to ensure a completely correct evaluation. We try to split the evaluation into several dimensions, thus makes the evaluation as simple and clear as possible. Therefore, the LLM-based evaluation can be more accurate. Due to the LLM language capability limitation, it can still make wrong judgments and requires a carefully designed prompt for constraint.

E Societal Impacts

As our proposed CAPability can perform comprehensive caption evaluations of MLLMs, this work can help LLM users make informed choice, and leads the community to build more and more strong MLLMs. The potential negative impacts are similar to other LLM-related works, The development of MLLMs and MLLMs' benchmarks pose societal risks like the perpetuation of biases, the potential for misinformation, job displacement, *etc*.