

## A. Training Details and Hyperparameters

### A.1. MetaMorph Training

We follow the training recipe outlined in prior studies [46, 63], using a two-stage training approach. First, we pretrain a two-layer MLP with a GELU activation [22] as the adapter between the visual tokens and the LLM. We train this adapter on Cambrian adapter data while excluding all data points sourced from LAION [56]. Next, we finetune the entire model, excluding the vision backbone, using the instruction tuning data described in Sec. 2.2 and detailed in Appendix C.

We use DeepSpeed [53] Zero-3 to train our model on H100 GPUs. Detailed training hyperparameters for all experiments are provided in Tab. 2. We conduct all of the experiments with 1 epoch.

### A.2. Diffusion Visualizer Training

We leverage pretrained diffusion models such as Stable Diffusion 1.5 [55]. We use a 2-layer MLP projector to align the SigLIP embedding dimension with the cross-attention dimension in the pretrained diffusion model. The first layer applies a linear transformation to map the input dimension to 2048, followed by layer normalization [7] and a ReLU activation. The second layer reduces the 2048-dimensional features to the output dimension through a linear transformation, followed by a final layernorm.

We set the batch size to 2112. The learning rate schedule begins with a logarithmic warm-up over the first 2000 steps, gradually increasing from zero to a peak value of  $1.1e-5$ . After this warm-up phase, the learning rate decreases linearly over the next 12000 steps until reaching zero. We use the AdamW [41] optimizer to train our model, with  $\beta$  parameters (0.9, 0.999). We apply a weight decay of 0.01.

During diffusion training, we freeze the VAE encoder and Siglip encoder, only training the projector and the diffusion unet. The CFG level is set to 0.7. This is because we start with a pretrained diffusion model and aim to transform the conditioning from CLIP text to SigLIP image embeddings. A higher CFG level ensures the model maintains high image quality while gradually adapting to the new conditioning in the remaining fraction. Empirically, this approach achieves the best balance between adaptation and image quality. For the training datasets, since we finetune the diffusion model to condition on SigLIP image embeddings, training this model does not require text descriptions for conditioning. Instead, we use images curated through in MetaCLIP [76] and train this diffusion model to visualize the visual tokens generated by MetaMorph.

### A.3. Evaluation Benchmarks

For evaluation, we use nine ImageQA, one VideoQA and two generation benchmarks:

- **MMBench** [40]: A comprehensive benchmark spans across 20 multimodal ability dimensions.
- **Seed** [19]: A benchmark focusing on visual tasks for multimodal understanding, consists of 19k multiple choice questions with accurate human annotations.
- **V\*STAR** [72]: A VQA benchmark designed for testing details in high-resolution images.
- **MMVP** [65]: A benchmark for evaluating “CLIP-Blind” pairs in Vision Language Models.
- **MMM** [78]: A benchmark designed to evaluate multimodal models on extensive multi-discipline tasks requiring college-level subject knowledge and deliberate reasoning.
- **ChartQA** [45]: A large-scale benchmark involving visual and logical reasoning over charts.
- **TextVQA** [58]: A benchmark designed to evaluate models’ ability to read and reason about text in images to answer questions.
- **ScienceQA** [44]: A multimodal benchmark for answering science-related questions requiring integration of visual and textual data.
- **RealWorldQA** [74]: A benchmark focused on real-world multimodal reasoning tasks.
- **MV-Bench** [33]: A benchmark contains a comprehensive video understanding benchmark, which covers 20 challenging video tasks that cannot be effectively solved with a single frame.
- **FID Score** [24]: A metric for evaluating the quality of generated images by comparing their feature distributions with real images.
- **CLIP Score** [23]: A benchmark metric that uses CLIP embeddings to measure alignment between generated images and their corresponding text descriptions.

## B. Ablation Studies on Visual Prediction Objective

We compare our approach to the commonly used L1 regression loss, which has been widely adopted in contrastive self-supervised learning methods [8, 31]. For this comparison, we train MetaMorph, based on LLaMA-3 8B, using datasets described in Sec. 2.2. We highlight that cosine similarity and L1 loss influence the embedding outputs differently: cosine similarity enforces normalization, while L1 loss does not. This discrepancy in output normalization prevents a direct and fair comparison in terms of generation performance. Consequently, our analysis focuses exclusively on VQA performance.

In Tab. 3, we compare models trained using L1 loss and cosine similarity loss. Our analysis reveals that training with cosine similarity results in better average performance and outperforms L1 loss on most benchmarks. Notably, these vision loss functions affect only tasks requiring visual predictions and do not directly influence VQA tasks, as the VQA training data does not include image token responses. This

Experiment	Backbone LLM	Data		Adapter			Instruction Tuning		
		Adapter	Instruction Tuning	lr	wd	bs	lr	wd	bs
Sec. 3 (LLaMA-3 8B)	LLaMA-3 8B	Cambrian Adapter Data*	Sec. 3 Experiment Setting	4.90e-5	0.0	768	6.93e-5	0	1536
Sec. 3 (LLaMA-3.1 8B)	LLaMA-3.1 8B	Cambrian Adapter Data*	Sec. 3 Experiment Setting	4.90e-5	0.0	768	6.93e-5	0	1536
Sec. 3 (LLaMA-3 70B)	LLaMA-3 70B	Cambrian Adapter Data*	Sec. 3 Experiment Setting	4.90e-5	0.0	768	4.90e-5	0	768
MetaMorph	LLaMA-3.1 8B	Cambrian Adapter Data*	All Data from Sec. 2.2	4.90e-5	0.0	768	6.93e-5	0	1536

Table 2. **Implementation details and hyperparameters for all experiments.** \*We exclude data points in LAION [56] from Cambrian adapter data.

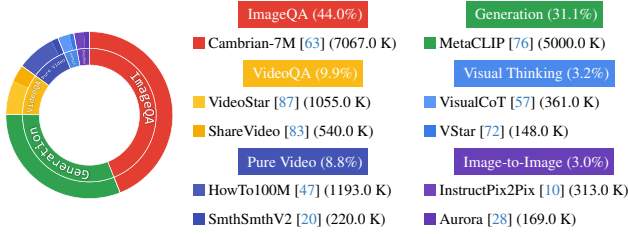


Figure 11. **Data composition.** **Left:** The inner circle shows the distribution of MetaMorph data. **Right:** All the data sources and categories in the MetaMorph data.

improvement is potentially because training with cosine similarity enhances visual generation, which in turn contributes to better visual understanding.

To further investigate, we compare our method—incorporating a broader range of non-VQA data alongside Cambrian-7M—with a baseline trained exclusively on Cambrian-7M. The results show that combining broader dataset with cosine similarity loss leads to better performance across multiple benchmarks. This finding reinforces our earlier observations in Sec. 3: enhancing visual generation capabilities contributes to improved visual understanding, highlighting the benefits of leveraging non-VQA data.

## C. Data

### C.1. Data Composition

We summarize the categorization of data and the number of samples for each source in Fig. 11. This diverse dataset is curated to showcase that an LLM can be finetuned across a variety of tasks, where each task contributes to and enhances the performance of others, as discussed in Sec. 3.1.

### C.2. Data Preprocessing

As discussed in Sec. 2.2, we use a wide range of data, spanning from visual question answering tasks to unlabeled video data. Here, we detail the preprocessing steps applied to each data source to convert them into instruction-tuning-style QA conversations.

**ImageQA.** We use Cambrian-7M [63], a dataset already curated in instruction tuning format. An example entry looks

like the below:

#### Example from ImageQA

##### Prompt:

<image\_start><image><image\_end> What is the animal in the image?

##### Response:

It is a burmilla cat.

**VideoQA.** We use VideoStar [87] and ShareVideo [12], both curated in an instruction tuning format. For each video, we extract frames at a rate of one frame per second and input these frames into the LLM. An example QA entry for an 8-second video is structured as follows:

#### Example from VideoQA

##### Prompt:

<image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>  
 <image\_start><image><image\_end>

What's the color of the dog in this video? (a) white (b) yellow (c) black Please only answer a single letter and nothing else

##### Response:

b

**Generation data.** We use image-text pairs in MetaCLIP [76]. The original data consists of images paired with corresponding text descriptions. We add system prompts and define answering formats, transforming the image-text pairs into question-answer formats suitable for instruction tuning.

Loss	Image QA									
	AVG	MMBench <sup>EN</sup>	SEED	RealworldQA	MMVP	SQA	MMMU	VStar	ChartQA	TextVQA
None (VQA Only)	55.50	73.11	69.96	55.69	41.33	80.39	37.29	46.60	35.16	59.96
L1 Loss	53.83	72.17	69.28	57.25	34.67	79.00	34.00	45.55	32.40	60.17
Cosine Sim	55.93	73.78	71.36	55.03	44.00	79.83	35.29	47.64	36.60	59.79

Table 3. **Comparison of different loss functions.** Training with cosine similarity loss enables the model to effectively utilize non-VQA data, which in turn enhances its visual understanding.

#### Example from Generation data

**Prompt:**

Generate an image of a puppy.

**Response:**

Here is an image based on your request:  
<image\_start><image><image\_end>.

#### Example from Image to image data

**Prompt:**

<image\_start><image><image\_end> Make this a rainy day.

**Response:**

<image\_start><image><image\_end>.

Different from the ImageQA and VideoQA, we require the model to predict the visual tokens in the response.

**Visual Thinking Data.** We explore incorporating vision as part of the model’s reasoning process to enhance its answers. As a preliminary step, we experiment with the Visualization-of-Thought [57] and VStar [72] datasets. Originally, these datasets were designed to teach models how to utilize external tools and APIs, such as segmentation or zoom-in cropping. In this work, we aim to integrate these visual skills directly into the model’s inference steps. We use system prompts, such as “*think visually before you answer the question*”, to activate this visual reasoning mode. Here is an example:

#### Example from Visual thinking data

**Prompt:**

Think visually before you answer the question: what is the color of the hat in the image  
<image\_start><image><image\_end>.

**Response:**

Let’s consider this visually:  
<image\_start><image><image\_end> black.

In this example, the model first predicts a zoomed-in view of the key region in the image necessary for answering the question, and then it provides the answer.

**Image to image data.** The model is required to predict a transformed image based on a given text description. We use Aurora [28] and InstructPix2Pix [10] datasets to train this capability. An example is presented below:

Here, the model is expected to predict the visual tokens of the transformed image as per the specified prompt.

**Pure video data.** We explore commonly used open-source video datasets in instruction tuning: SomethingSomethingV2 [20] and HowTo100M [47]. We design the following tasks from the pure video:

1) Forward Frame Prediction. In this task, the model is presented with the initial frame of a video sequence and must predict the subsequent frames at fixed time intervals. An example is presented below:

#### Example of Forward Frame Prediction

**Prompt:**

<image\_start><image><image\_end> Can you predict what happens in the next 3 frames, each 5 seconds apart?

**Response:**

<image\_start><image><image\_end>  
<image\_start><image><image\_end>  
<image\_start><image><image\_end>

2) Partial Sequence Completion. This task requires the model to complete a video sequence when given only a subset of frames while maintaining temporal coherence:

#### Example of Partial Sequence Completion

**Prompt:**

```
<image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end> Can you pre-
dict the 2 missing frames in this 5-second-interval se-
quence?
```

**Response:**

```
<image_start><image><image_end>
<image_start><image><image_end>
```

3) Reverse Temporal Prediction. This task challenges the model to reconstruct the preceding frames given the final frame of a sequence:

#### Example of Reverse Temporal Reasoning

**Prompt:**

```
<image_start><image><image_end> Work back-
wards to predict the previous 4 frames, each 5 seconds
apart.
```

**Response:**

```
<image_start><image><image_end>
><image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end>
```

4) Temporal Sequence Reordering. In this task, the model receives a shuffled sequence of video frames and must reconstruct their correct temporal order:

#### Example of Temporal Sequence Reordering

**Prompt:**

```
<image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end>
Arrange these frames in their correct temporal sequence.
```

**Response:**

```
<image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end>
<image_start><image><image_end>
```

Each task is designed to train the model’s temporal understanding and visual reasoning capabilities.

### C.3. Potential Image Leakage in Testing Data

When selecting data sources, we carefully choose those that do not overlap with the testing sets of our evaluation data, such as COCO [35]. However, given that the data used in a Sec. 2.2 is composed of numerous sources, some degree of

data leakage may be inevitable. As discussed and analyzed in a prior work [63], even when image overlap occurs, it does not necessarily imply that the exact image-question pairs have been encountered during training. Unlike traditional unimodal computer vision research, where an image alone constitutes a data point, the multimodal paradigm treats each image-text (question-answer) pair as a distinct and unique data point.

## D. Generating Visual Tokens

Here, we include the quantitative results of all the experiments in Sec. 3.

### D.1. Results of Samples Needed to Unlock Visual Generation

Tab. 4 presents the quantitative results corresponding to Fig. 2, which examines generation performance under two conditions: training exclusively on generation data and joint training with all other data described in Sec. 2.2. The results demonstrate that the model can develop the ability for visual generation with a relatively modest amount of data when trained jointly with understanding tasks. In contrast, teaching this skill in isolation requires a substantially larger dataset.

In Tab. 5, we present the quantitative results corresponding to Fig. 3, which investigates the impact of joint training on generation data in combination with various types of data outlined in Sec. 2.2. The results show that joint training with visual understanding data—specifically ImageQA and VideoQA—provides the most significant improvement in visual generation performance.

### D.2. Results of Joint training Different Understanding and Generation Data

In Tab. 6, we present the numerical results of joint training with varying scales of understanding data (1M, 4M, 7M) and generation data (200k, 500k, 1M, 2M, 3M, 4M). These findings demonstrate that increasing the amount of understanding data yields more substantial improvements in both understanding tasks (e.g., VQA performance) and generation tasks (e.g., FID scores and CLIP scores) compared to increasing the amount of generation data. These results, consistent with our analysis in Sec. 3.2 and Sec. 3.3, highlight that understanding data play a more pivotal role in enhancing performance across both task types.

### D.3. Results of Training on Different LLMs

We present the results of training with 7M VQA data and 1M generation data across various LLM backbones, including LLaMA-3 8B, LLaMA-3.1 8B, and LLaMA-3 70B. As shown in Tab. 7, which corresponds to the results in Fig. 6, we observe that stronger LLM backbones lead to improvements in both visual understanding and visual generation.

Joint train With Other Data	# of Generation Data	FID Score
Yes	1k	68.5
No	1k	115.0
Yes	5k	19.2
No	5k	116.4
Yes	10k	18.7
No	10k	111.0
Yes	50k	17.1
No	50k	111.8
Yes	200k	15.2
No	200k	110.7
Yes	200k	14.7
No	200k	93.7
Yes	1M	14.4
No	1M	52.8
Yes	3M	15.1
No	3M	39.2
Yes	5M	14.3
No	5M	27.7

Table 4. **Results of training solely on generation data vs. joint training with additional data.** These results correspond to Fig. 2. Joint training with additional data significantly improves generation performance. At 5,000 samples, the model begins to generate reasonably accurate visual tokens, indicating that visual generation is an ability unlocked through the learning of other tasks.

These findings further support the conclusion that visual understanding and generation are reciprocal processes, where advancements in one drive enhancements in the other.

## E. More Examples of MetaMorph

We provide additional examples of MetaMorph in Fig. 12 and Fig. 13. These examples illustrate how MetaMorph extends beyond the capabilities of typical MLLMs by leveraging learned skills to perform novel tasks such as visual reasoning and visual transformation. In Fig. 12, when prompted with the question “*What is the type of hat?*”, MetaMorph first generates visual tokens related to hats and then answers correctly with “*top hat*”. The model also demonstrates the ability to perform image transformations, such as creating a cartoon version of an image or altering it to appear as day-time. Additionally, we showcase examples of MetaMorph solving implicit puzzles, such as interpreting “*a rearrangement of the letters in the word 'tca'*”, before generating the corresponding visual tokens of cats.

Joint training Data	Data Type	FID Score	CLIP Score
None	-	110.5	5.7
Image-to-Image	Other Visual Data	97.5	6.4
Visual Thinking	Other Visual Data	93.5	6.5
Pure Video	Other Visual Data	84.7	8.1
VideoQA	Visual Understanding Data	26.5	16.1
ImageQA	Visual Understanding Data	18.9	22.0

Table 5. **Impact of joint training 200k generation data with different data types.** These results correspond to Fig. 3. Among the data types analyzed, joint training with visual understanding data has the most significant impact on enhancing visual generation performance.

Data Composition		Image QA										Generation	
# of VQA Data	# of Generation Data	Average	MMBench <sup>EN</sup>	SEED	RealworldQA	MMVP	SQA	MMMU	VStar	ChartQA	TextVQA	FID Score	CLIP Score
1M	200k	46.4	60.0	62.2	50.3	24.0	80.0	38.4	37.4	16.4	48.8	28.3	15.2
1M	500k	48.2	66.4	63.2	50.8	24.3	80.4	39.9	38.7	18.2	51.6	28.1	15.9
1M	1M	49.1	70.1	65.2	52.2	21.3	80.0	39.5	38.7	20.4	54.6	27.3	16.5
1M	2M	49.9	67.8	66.0	50.2	30.3	80.2	38.9	39.0	21.8	54.8	23.1	17.8
1M	3M	51.1	71.3	67.1	55.4	33.0	79.5	38.8	37.4	22.7	55.0	21.1	21.1
1M	4M	51.4	71.1	66.9	52.4	31.0	80.5	39.8	41.1	24.0	56.0	18.4	22.3
4M	200k	53.8	73.1	68.8	55.0	34.7	81.2	38.5	44.0	29.5	59.2	21.4	20.5
4M	500k	53.3	73.0	69.9	55.3	32.7	80.6	40.2	39.3	29.6	58.9	16.0	24.8
4M	1M	54.2	73.8	69.6	54.9	33.3	82.1	36.6	45.6	32.4	59.9	16.0	24.8
4M	2M	53.8	72.8	70.3	55.2	37.3	80.8	36.8	44.0	31.2	56.2	15.6	24.7
4M	3M	54.3	71.8	70.1	57.7	36.0	81.0	38.0	42.9	32.6	59.0	16.1	24.8
4M	4M	54.4	75.2	69.9	56.0	37.3	81.4	38.1	40.8	31.6	59.3	15.3	25.5
7M	200k	55.8	73.1	70.3	55.6	42.0	81.0	40.8	44.0	35.2	60.6	18.2	22.3
7M	500k	55.6	74.4	70.6	56.2	38.7	81.9	37.9	44.0	36.0	60.5	15.2	25.5
7M	1M	55.8	74.3	70.3	56.3	42.7	81.3	36.6	44.5	35.8	60.6	14.5	26.6
7M	2M	55.4	73.9	71.1	56.9	40.0	81.6	35.9	42.4	35.4	61.6	14.8	27.1
7M	3M	55.6	74.2	71.0	57.3	38.0	81.1	40.1	43.5	35.0	60.2	14.2	27.5
7M	4M	56.2	75.4	70.4	55.4	44.0	80.4	39.6	45.0	35.2	60.2	14.9	26.3

Table 6. **Full results of joint training on varying amounts of VQA data (1M, 4M, 7M) and generation data (200k, 500k, 1M, 2M, 3M, 4M).** These results correspond to Fig. 4, Fig. 5, Fig. 7, and Fig. 8, which analyze how different combinations of understanding and generation data impact the model’s visual understanding and generation performance.

Pretrained LLM		Image QA									Generation	
LLM	Average	MMBench <sup>EN</sup>	SEED	RealworldQA	MMVP	SQA	MMMU	VStar	ChartQA	TextVQA	FID Score	CLIP Score
LLaMA-3 8B	55.8	74.3	70.3	56.3	42.7	81.3	36.6	44.5	35.8	60.6	14.5	26.6
LLaMA-3.1 8B	56.7	75.8	70.2	56.2	44.7	81.9	41.2	43.4	36.0	61.3	13.2	27.1
LLaMA-3 70B	60.7	80.7	72.6	58.3	48.7	87.8	48.9	47.1	37.4	65.0	13.8	26.8

Table 7. **Full results of training on different LLMs.** We train 7M VQA data and 1M generation data on different LLM backbones (LLaMA-3 8B, LLaMA-3.1 8B, and LLaMA-3 70B) and measure understanding and generation performance.



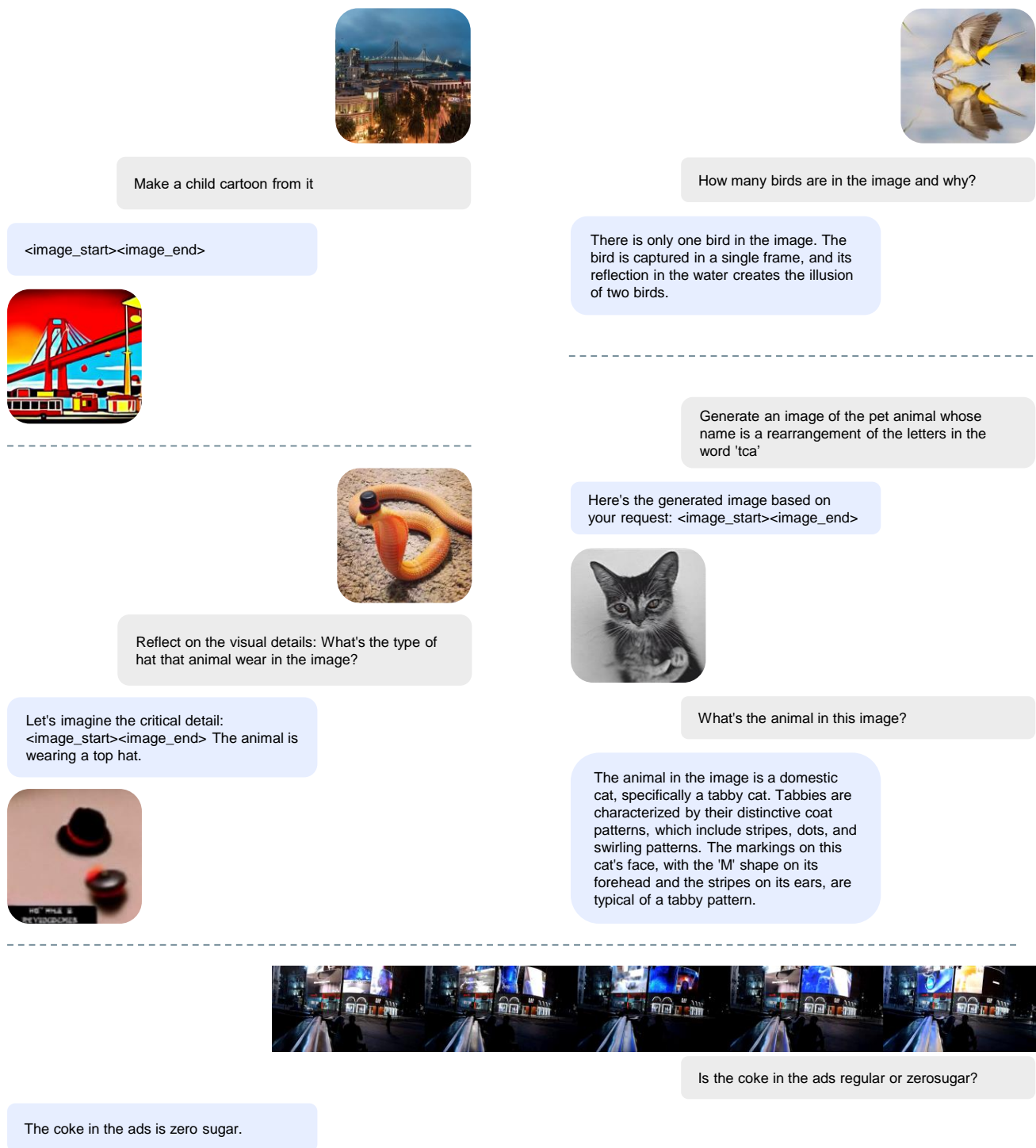


Figure 12. **Examples of MetaMorph (I).** We showcase examples of MetaMorph’s capabilities: transforming images based on prompts (*top-left*), answering challenging questions (*top-right*), integrating visual tokens into reasoning processes (*bottom-left*), implicitly solving puzzles (*bottom-right*) and answering tricky video-qa questions. (*bottom*).

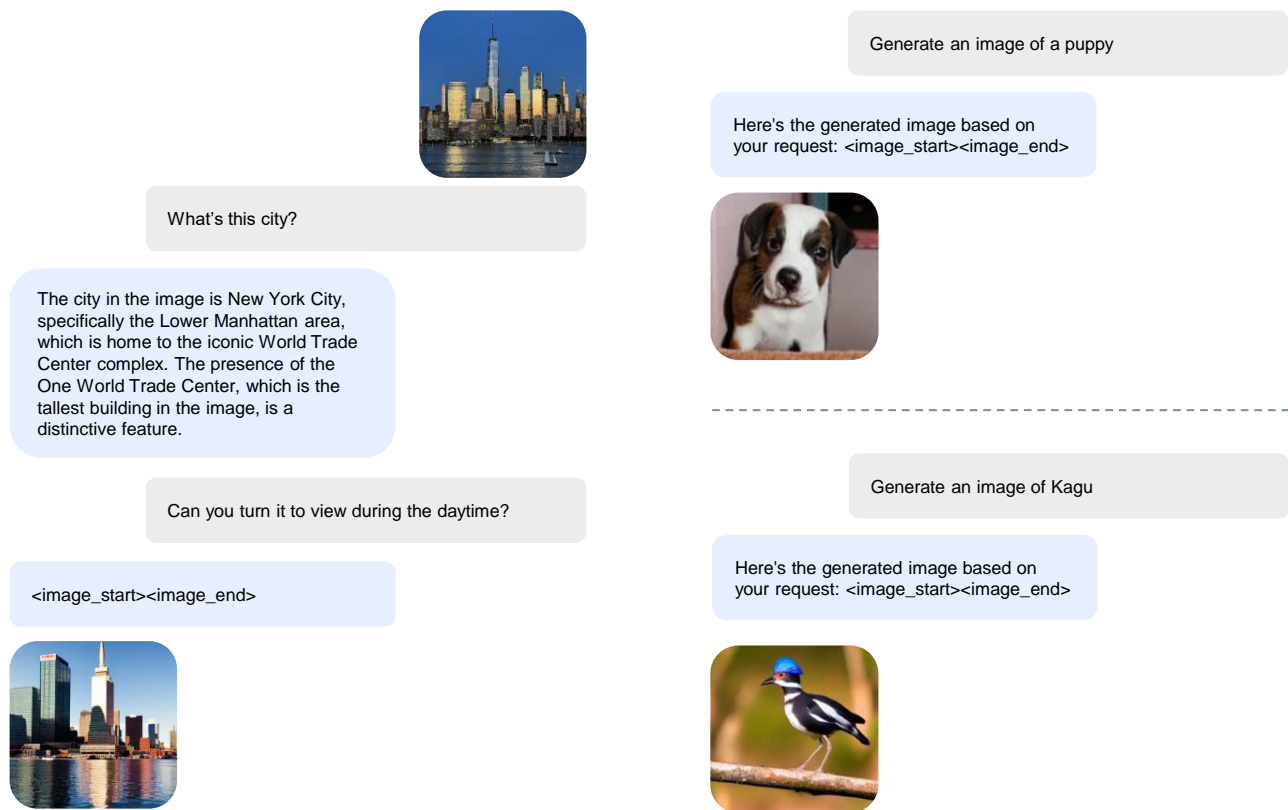


Figure 13. **Examples of MetaMorph (II).** We showcase more examples of MetaMorph’s capabilities: answering questions and transforming images in one conversation (*left*), generating images (*top-right*), and leveraging knowledge in LLMs to generate rare concepts (*bottom-right*).