# Olympus: A Universal Task Router for Computer Vision Tasks

**Yuanze Lin**♣  **Yunsheng Li**♠  **Dongdong Chen**♠
**Weijian Xu**♠  **Ronald Clark**♣  **Philip H. S. Torr**♣

♣University of Oxford  ♠Microsoft

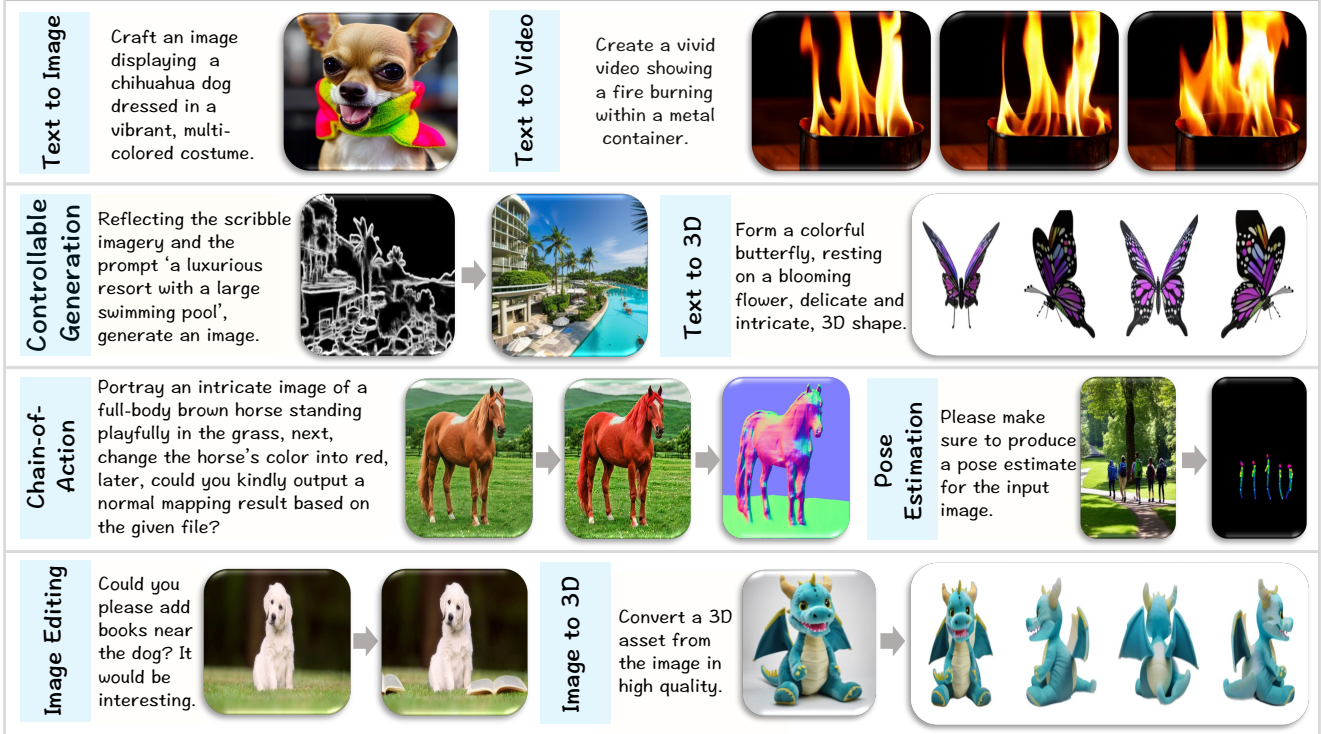http://yuanze-lin.me/Olympus_page/

Figure 1. With its versatile capabilities, *Olympus* addresses a broad spectrum of vision tasks across images, videos, and even 3D content, in fact, it can cover over 20 different tasks.

## Abstract

*We introduce Olympus, a new approach that transforms Multimodal Large Language Models (MLLMs) into a unified framework capable of handling a wide array of computer vision tasks. Utilizing a controller MLLM, Olympus delegates over 20 specialized tasks across images, videos, and 3D objects to dedicated modules. This instruction-based routing enables complex workflows through chained actions without the need for training heavy generative models. Olympus easily integrates with existing MLLMs, expanding their capabilities with comparable performance. Experimental results demonstrate that Olympus achieves an average routing accuracy of 94.75% across 20 tasks and precision of 91.82% in chained action scenarios, showcasing its effectiveness as a universal task router that can solve a diverse range of computer vision tasks.*

## 1. Introduction

*"If I have seen further it is by standing on the shoulders of Giants."* – Isaac Newton

Multimodal Large Language Models (MLLMs) have made significant strides in advancing understanding, generation and reasoning across diverse domains. For example, in understanding, MLLMs like LLaVA [47] excel in visual question-answering (VQA) [4], effectively integrating visual and textual data. In generation, diffusion models [26, 56, 58] have achieved exceptional re-

sults in text-to-image, text-to-video and text-to-3D generation [9, 19, 27, 42, 43, 58, 60, 64, 77].

Building on these advancements, recent studies [17, 21, 41, 70, 78, 81, 87] have aimed to develop unified architectures capable of performing tasks across various domains. Notably, Emu3 [74] and Omni-Gen [79] introduce all-in-one models designed to handle both generation and understanding tasks. However, the integration of distinct domains within a single model continues to present significant challenges. In particular, variability within domains often leads to compromised performance on individual tasks due to conflicts between task objectives, such as those between text and image generation tasks [96]. These conflicts hinder the models' effectiveness and limit their utility in real-world applications.

Another key limitation of all-in-one models lies in their constrained ability to handle a broad spectrum of vision-language tasks across different domains due to differing input and output formats. This restriction presents a substantial bottleneck to scalability, particularly as the range of tasks continues to grow across images, videos, and the emerging 3D domain. Furthermore, extending these models to accommodate new tasks is inherently challenging. Training such comprehensive models with increasing model sizes demands substantial computational resources, and highly complex training methodologies. For instance, Omni-Gen [79] necessitates $104 \times$A800 GPUs and five distinct training stages. These issues underscore the pressing need for modular or task-adaptive frameworks to enhance scalability and efficiency in addressing the increasingly diverse demands of vision tasks. Additionally, all-in-one models often struggle to integrate meticulously designed, task-specific components effectively, reducing their overall efficiency and performance in specialized applications.

This prompts us to explore another alternative approach for seamlessly unifying vision tasks within a single framework. Inspired by HuggingGPT [62] and the advanced contextual understanding of MLLMs, we propose leveraging MLLMs to handle vision-language comprehension *internally* while delegating other tasks *externally*. Although technically straightforward, it represents a foundational and significant step toward advancing unified frameworks for computer vision tasks. Specifically, the MLLM can function as a task router, coordinating with specialized external models to address various tasks and overcome individual model limitations. However, it still faces significant challenges, primarily due to the variability in user prompts across a wide range of tasks and the lack of comprehensive, task-specific instruction datasets essential for effective training and evaluation.

In this paper, we introduce *Olympus*, a unified framework that leverages MLLMs to handle a diverse array of computer vision tasks. Our *Olympus* differs from existing methods [71, 74, 81, 96] which focus on presenting all-in-one

models to solve diverse tasks. To accomplish this, we collected **446.3K** high-quality training instructions and **49.6K** evaluation instructions from GPT-4o [29] named as *OlympusInstruct* and *OlympusBench* respectively, spanning **20** different vision tasks. Furthermore, we designed specific routing tokens tailored to delegate individual tasks. Finally, leveraging these routing tokens and *OlympusInstruct*, our model can even perform a chain of tasks within a single user instruction if needed.

In our experiments, *Olympus* achieves comparable performance to the leading MLLMs on standard multimodal benchmarks [48]. Additionally, it supports over **20** distinct tasks across the domains of image, video, and 3D, as shown in Figure 1. We further investigate the effectiveness of decomposing user instructions to interface with suitable external models, *Olympus* achieves an impressive average routing accuracy of **94.75%** across **20** individual tasks. In chain-of-action scenarios, which involves performing multiple tasks to complete an instruction, our model attains **91.82%** precision. These results highlight the potential of *Olympus*. In summary, our contributions can be included as:

- We introduce *Olympus*, an innovative framework that leverages Multimodal Large Language Models (MLLMs) to perform contextual understanding tasks through their inherent capabilities, while addressing other tasks via allocating external models.
- We develop task-specific routing tokens and enhance MLLMs with chain-of-action capabilities. Our model achieves comparable performances with leading MLLMs on multimodal benchmarks, *Olympus* achieves **94.75%** routing accuracy in single-task scenarios, **91.82%** precision in chain-of-action settings, and solves up to **5** tasks within a single instruction.
- We have curated high-quality instruction datasets named *OlympusInstruct* and *OlympusBench* across **20** computer vision tasks, comprising **446.3K** and **49.6K** samples for training and evaluation respectively. These datasets provide a solid foundation for further exploration and advancement in this domain.

## 2. Related Work

### 2.1. Vision-Language Understanding

Recent advancements in large language models (LLMs) [7, 73] have catalyzed the development of multimodal large language models (MLLMs) [2, 3, 5, 11, 16, 34, 35, 37, 44–46, 48, 54, 57, 63, 72, 82]. Pioneering multimodal large language models (MLLMs), such as MiniGPT-4 [97], have demonstrated impressive capabilities in processing and integrating multiple modalities. Models like Kosmos-2 [57], LLaVA [48] and LLaVA-OneVision [34] have further enhanced the visual cognitive abilities of MLLMs. Additionally, approaches including LLaVA-Phi [100], Mo-
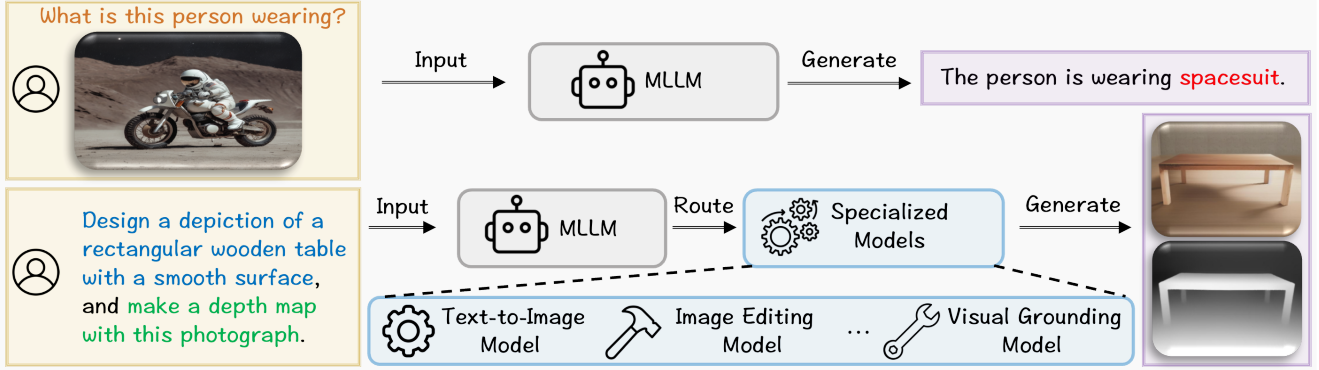
Figure 2. Given the user prompts, a trainable MLLM can perform routing across a wide range of specified models. In this concept, MLLMs can solve multimodal understanding tasks (e.g., VQA) with its inherited capacity, while MLLMs can allocate appropriate specialized models to address multimodal generative and classic vision tasks (e.g., image generation and depth estimation), then aggregate the results and deliver a response to the user.

bileVLM [12], and Mipha [99] focus on refining training methodologies and architectural frameworks to develop more efficient and lightweight MLLMs. Although these models excel in visual perception and multimodal understanding, they are predominantly limited to generating text-based outputs, which restricts their effectiveness across a broader range of vision tasks involving images, videos, and 3D content generation. In this work, we adopt a multimodal model structure following Mipha [99].

## 2.2. Unified Vision-Language Foundation Model

Extensive research [1, 17, 21, 66, 70, 74, 78, 81, 84, 89, 93, 96] has focused on developing unified multimodal language models proficient in both understanding and generating content. Approaches such as [21, 98] integrate continuous embeddings with textual tokens within autoregressive frameworks for image generation. Emu2 [66] combines CLIP ViT [18] image embeddings with text tokens for autoregressive modeling, while Chameleon [71] employs a transformer across diverse modalities with autogressive modeling. Show-o [81] and TransFusion [96] incorporate autoregressive and diffusion modeling within a single transformer. Omni-Gen [79] utilizes a VAE [32] encoder-decoder alongside a transformer to process free-form prompts. Recently, Emu3 [74] trained a unified transformer with next-token prediction across video, image, and text datasets, achieving superior performance on multimodal benchmarks and generation tasks. However, current unified multimodal foundation models predominantly support a narrow range of generative tasks, such as image and video creation or editing, and face significant scalability challenges for broader AI applications. Additionally, their training requires substantial computational resources. To overcome these limitations, we further enhance MLLMs by enabling the seamless integration of domain-specialized models tailored for diverse applications.

## 2.3. LLM-Based Tools

Large language models (LLMs) [73], trained on extensive datasets, demonstrate exceptional proficiency in zero-shot and few-shot settings, as well as in complex tasks such as mathematical problem-solving and commonsense reasoning. To extend their capabilities beyond text generation, recent research [38, 40, 59, 61, 62, 62, 68, 75] has focused on integrating external tools and models into LLM architectures. Toolformer [61] pioneered this approach by embedding API calls within textual sequences, thereby enabling LLMs to utilize external tools effectively. Building upon this foundation, subsequent studies have incorporated visual modalities: Visual ChatGPT [75] integrates LLMs with visual models such as BLIP [35], Visual Programming [23] and ViperGPT [68] translate visual queries into executable Python code, facilitating the processing of visual data by LLMs. Additionally, HuggingGPT [62] enhances large language models (LLMs) by utilizing them as controllers that direct user requests to specialized expert models, thereby integrating language comprehension with domain-specific expertise.

Although HuggingGPT assigns specific AI models to perform various tasks, it primarily relies on prompt engineering to leverage ChatGPT as an interface for connecting diverse external models without training. In contrast, our approach involves training multimodal large language models (MLLMs) from the ground up, enabling them to internally handle vision-language understanding tasks while designate specialized models to address a wide range of AI tasks.

## 3. *Olympus*

*Olympus* leverages MLLMs as the foundation for various computer vision tasks. For vision-language tasks like visual question answering (VQA), MLLMs utilize their inherent capabilities. For other vision tasks, such as generative tasks
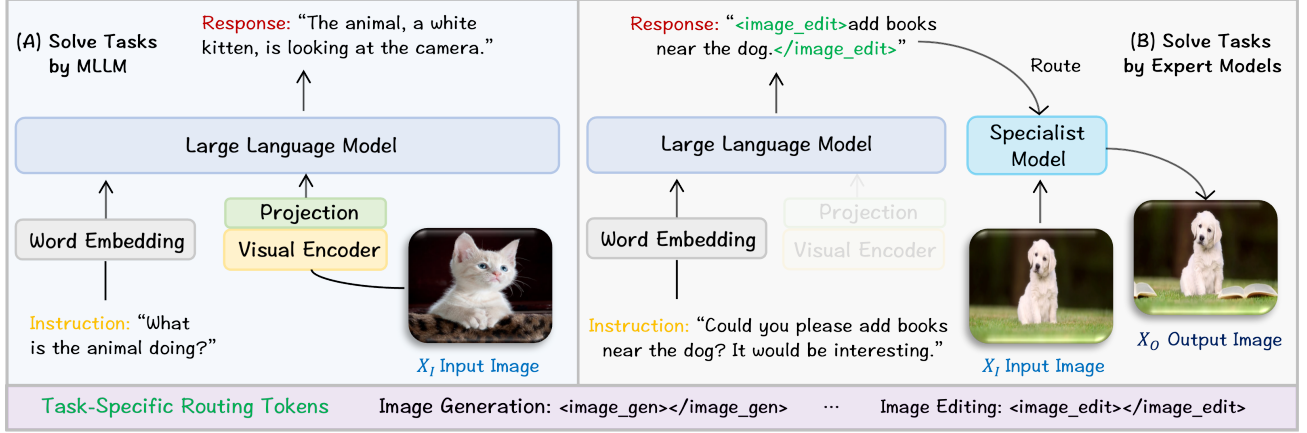
Figure 3. The framework of *Olympus*. It can solve those tasks like VQA through the inherited capacities of MLLM directly. For other tasks, e.g., image editing, *Olympus* can generate the response, which consists of task-specific routing tokens and refined prompts, they are then used to schedule specialist models for addressing diverse user requests.

(e.g., image, video, and 3D generation) and classic vision tasks (e.g., image super-resolution and depth estimation), MLLMs act as intermediaries, routing user instructions to specialized models. As shown in Figure 2, upon receiving a request, the MLLM can autonomously orchestrate the workflow, coordinating expert models to achieve the objective. The following subsections outline the details of *Olympus*.

## 3.1. Instruction Dataset Collection

In order to accurately assign user instructions to the appropriate model, we constructed a high-quality and diverse dataset of user instruction–response pairs using GPT-4o. This dataset comprises **446.3K** training samples, designated as *OlympusInstruct*, and **49.6K** evaluation samples, designated as *OlympusBench*, encompassing **20** distinct tasks. For each task, a specialized prompt was developed to align with the specific context of the task. This involved crafting detailed directives that enable GPT-4o to generate coherent and contextually relevant user requests and responses. An example of the image editing prompt used by GPT-4o to collect user instruction–response pairs is provided in Figure 4.

To ensure diversity in user instructions, we incorporated various prefixes and phrases that introduce different language styles, tones, and structures. Additionally, we categorized instruction complexities into three levels: short, moderate, and extended. This stratification allows GPT-4o to produce instructions that vary in length and complexity. Furthermore, we prompted GPT-4o to generate responses that are both practical and direct, thereby enhancing the applicability and clarity of the interactions. We also performed thorough data cleaning by removing duplicate entries and utilizing GPT-4o to remove entries that were contextually or grammatically inappropriate. This purification process ensures the integrity and quality of the dataset. Figure 4 presents examples for image editing task, demonstrating how the task-specific prompts enable GPT-4o to generate instructions with diverse levels of complexity and varied language styles.

Figure 5 illustrates the statistical characteristics of our training and evaluation datasets. Specifically, the training set comprises **381.5K** single-task instruction–response pairs and **64.8K** chain-of-action instruction–response pairs. Similarly, the evaluation set consists of **49.6K** single-task pairs and **7.2K** chain-of-action pairs. The maximum word length across all instructions is **372** words, with an average instruction length of **20.2** words. Responses have an average length of **10.7** words. The **20** covered tasks are categorized into three groups: **(1) Image domains:** image generation/editing, deblurring, deraining, super-resolution, denoising, pose/normal/canny/depth estimation, controllable image generation across six conditions (canny, pose, segmentation, depth, normal, scribble), object detection/segmentation, and visual grounding; **(2) Video domains:** video generation/editing, controllable video generation across the same six conditions, and referring video object segmentation; and **(3) 3D domains:** image-to-3D and text-to-3D generation.

## 3.2. Task-Specific Routing Tokens

As shown in Figure 3, *Olympus* directs user requests to dedicated models using task-specific routing tokens (e.g., image editing). To facilitate MLLMs in predicting appropriate models aligned with users' goals, we design a set of routing tokens specific to individual tasks. For instance, in the domains of image and video generation, we use the routing tokens `<image_gen>···</image_gen>` and `<video_gen>···</video_gen>`, respectively. Given a user instruction such as `"Please craft an image displaying a chihuahua dog dressed in a vibrant, multicolored costume."`, the corresponding response can be `<image_gen>a chihuahua dog dressed in a vibrant, multicolored`

Generate 50 unique user instructions with paired responses for image editing task focused on modifying objects and backgrounds. These instructions should vary significantly in language style, tone, and complexity to capture authentic interactions. Each instruction should range from short instruction into moderate and even extended instructions with multi-sentence descriptions. Occasionally use prefixes to diversify each instruction's phrasing. Responses must be extremely short, direct and practical, effectively addressing each instruction. Refer to the samples below as guidance, and come up with a wide range of prefixes and phrases to develop short, moderate and extended user instructions.

**Prefixes:**
1. Is there a way to. 2. Is it possible to. 3. I would appreciate it if you could. 4. Would you. 5. Could you. 6. Let's. 7. Please.

**Phrases:**
1. replace [object] with [new object]. 2. turn [object] into [new object]. 3. swap [object] with [new object]. 4. erase [object]. 5. let [object] [action]. 6. change [object] to [new object]. 7. remove [object]. 8. add [new object] to [object]. 9. place [new object] beside [object]. 10. make [object] [action]. 11. have [object] [action]. 12. leave [object] on [new object].

**User Instruction Complexities:**
1. Short Instruction: One-sentence requests using concrete objects and actions. 2. Moderate Instruction: Two sentences, adding context for clarity. 3. Extended Instruction: Over three sentences, describing emotions, effects, or scenes for richer context.

**User Instruction-Response Pairs:**
Short Instruction: Is there a way to replace the mountain with a beach?
- Response: Replace the mountain with a beach.
Moderate Instruction: Could you help me change dragonfly to spaceship. I believe this change will give a more inviting feel. -
- Response: Change dragonfly to spaceship.
Extended Instruction: Please replace the old fence in the background with a modern wooden one. The style should be clean and natural-looking, fitting in with a suburban backyard. I'm envisioning a style that's clean and simple, with natural tones that blend seamlessly into a cozy suburban backyard setting. The new fence should add a warm, inviting feel to the space, creating a sense of privacy without feeling closed off.
- Response: Replace the old fence with a modern, clean and natural-looking wooden fence.

GPT4 Output Examples

**User Instruction-Response Pairs:**
Short Instruction: Could you replace the streetlight with a vintage lantern?
- Response: Replace streetlight with vintage lantern.
Moderate Instruction: Could you add a cozy fireplace to the living room? It should create a warm and inviting atmosphere.
- Response: Add a cozy fireplace to the living room.
Extended Instruction: I'd appreciate it if you could swap the city skyline with a mountain range. The mountains should look majestic and slightly misty, as if it's early morning. I want the scene to feel natural and peaceful, with a sense of solitude in the wilderness.
- Response: Replace city skyline with misty mountains.

Figure 4. One example to illustrate the prompt we use to generate the instruction-response pairs for image editing by GPT-4o.

costume.</image_gen>, which is designed to effectively address the user's request. Thus, user instructions and the responses form input-answer pairs for training. Detailed information on designed routing tokens and corresponding specialist models are explained in the **Appendix**.

**Chain-of-Action.** By introducing domain-specific routing tokens, *Olympus* enables chain-of-action capabilities, allowing to handle multiple tasks within a single instruction. For instance, consider a user prompt combining pose-based image generation and image editing: `"In homage to the pose imagery and the prompt 'a majestic castle', generate an image. In the following step, please refine the image by adding green trees."` The predicted response using the routing tokens is: `<pose_to_image>a majestic castle</pose_to_image><image_edit>adding green trees</image_edit>`. Therefore, *Olympus* can sequentially route user instructions to the appropriate modules for pose-conditioned image generation and image editing, in alignment with the task-oriented routing tokens. Moreover, *Olympus* supports up to five consecutive tasks within a single prompt and is capable of scaling to accommodate an even larger number of tasks, thereby demonstrating its flexibility and scalability.

### 3.3. Training

Since the goal is to generate task-specific response together with its routing tokens conditioning on the user instructions, we can train MLLMs with next-token prediction paradigm using the cross-entropy loss:

$$P(Y_a|\mathcal{F}_v, \mathcal{F}_t) = \prod_{i=1}^{L} P_{\theta}(y_i|\mathcal{F}_v, \mathcal{F}_t, Y_{a,<i}). \quad (1)$$

Here, $L$ represents the sequence length of the response $Y_a$, $\mathcal{F}_v$ denotes the visual embedding which is adopted for those multimodal instructions, and $\theta$ means the trainable parameters of MLLMs. The notation $Y_{a,<i}$ denotes all tokens preceding the current token $y_i$, and $\mathcal{F}_t$ represents the input instruction embeddings.

### 3.4. Inference

As displayed in Figure 3, upon receiving a prompt, *Olympus* generates a response with task-customized routing tokens. These tokens invoke the appropriate AI models to handle various tasks, and their predictions are aggregated into a final response. For tasks solvable by MLLMs alone, responses are generated directly, bypassing routing tokens.

| Additional Covered Tasks |
| --- |
| Image Generation, Image Editing, Controllable Image Generation (Canny, Pose, Segmentation, Depth, Normal, Scribble), Video Generation, Text-to-3D Generation, Image-to-3D Generation, Image Deblurring, Image Super-Resolution, Image Deraining, Image Denoising, Pose Estimation, Normal Estimation, Canny Estimation, Depth Estimation, Visual Grounding, Object Detection, Object Segmentation, Referring Video Object Segmentation, Controllable Video Generation (Canny, Pose, Segmentation, Depth, Normal, Scribble), Video Editing |

(a) 20 covered tasks in Olympus framework.



(b) Number of instructions for different tasks.

| Dataset Statistic | |
| --- | --- |
| # of Training Instructions (Single Task) | 381.5K |
| # of Training Instructions (Chain-of-Action) | 64.8K |
| # of Evaluation Instructions (Single Task) | 49.6K |
| # of Evaluation Instructions (Chain-of-Action) | 7.2K |
| Max Instruction Word Length | 372 |
| Ave Instruction Word Length | 20.2 |
| Ave Response Word Length | 10.7 |
| Ave # of COA Tasks | 3.4 |

(c) Statistic of the collected dataset.



# of 2-task: 13955     # of 3-task: 25565
# of 4-task: 23709     # of 5-task: 8771

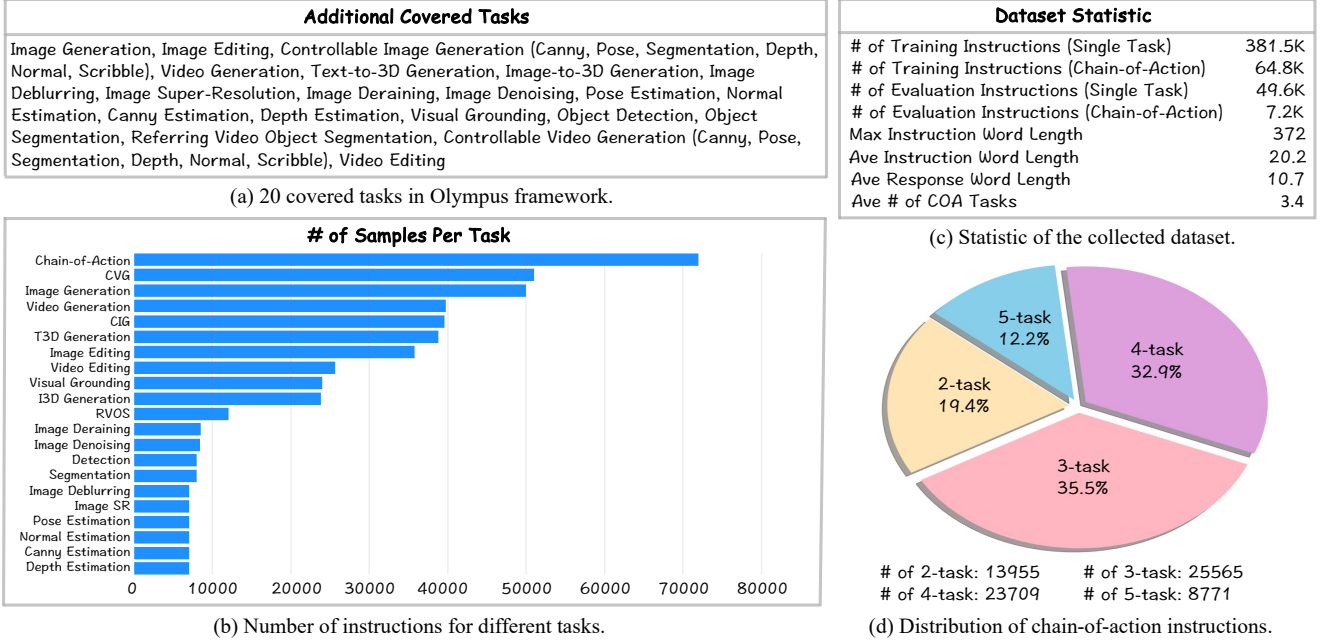(d) Distribution of chain-of-action instructions.

Figure 5. The statistic of the collected dataset. Note that CVG and CIG denote controllable video generation and controllable image generation, RVOS represents referring video object segmentation and image SR means image super-resolution in figure (b).

| Method | LM | Res. | VQAv2 | GQA | VisWiz | SQA$^I$ | VQA$^T$ | MME-P | MME-C | MMB | MM-Vet | POPE | MMMU |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Shikra [8] | V-13B | 224 | 77.4 | - | - | - | - | - | - | 58.8 | - | - | - |
| IDEFICS-9B [33] | L-7B | 224 | 50.9 | 38.4 | 35.5 | - | 25.9 | - | - | 48.2 | - | - | - |
| IDEFICS-80B [33] | L-65B | 224 | 60.0 | 45.2 | 36.0 | - | 30.9 | - | - | 54.5 | - | - | - |
| Qwen-VL-Chat [5] | Q-7B | 448 | 78.2 | 57.5 | 38.9 | 68.2 | 61.5 | 1487.5 | 360.7 | 60.6 | - | - | 32.9 |
| mPLUG-Owl2 [88] | L-7B | 448 | 79.4 | 56.1 | 54.5 | 68.7 | 58.2 | 1450.2 | 313.2 | 64.5 | 36.2 | 85.8 | 32.1 |
| LLaVA-1.5 [45] | V-7B | 336 | 78.5 | 62.0 | 50.0 | 66.8 | 58.2 | 1510.7 | 316.1 | 64.3 | 30.5 | 85.9 | 32.0 |
| MobileVLM-3B [12] | M-2.7B | 336 | - | 59.0 | - | 61.2 | 47.5 | 1288.9 | - | 59.6 | - | 84.9 | - |
| MobileVLM-v2-3B [13] | M-2.7B | 336 | - | 61.1 | - | 70.0 | 57.5 | 1440.5 | - | 63.2 | - | 84.7 | - |
| LLaVA-Phi [100] | P-2.7B | 336 | 71.4 | - | 35.9 | 68.4 | 48.6 | 1335.1 | - | 59.8 | 28.9 | 85.0 | - |
| Imp-v1 [67] | P-2.7B | 384 | 79.5 | 58.6 | - | 70.0 | 59.4 | 1434.0 | - | 66.5 | 33.1 | 88.0 | - |
| MoE-LLaVA-3.6B [39] | P-2.7B | 384 | 79.9 | 62.6 | 43.7 | 70.3 | 57.0 | 1431.3 | - | 68.0 | 35.9 | 85.7 | - |
| TinyLLaVA [95] | P-2.7B | 384 | 79.9 | 62.0 | - | 69.1 | 59.1 | 1464.9 | - | 66.9 | 32.0 | 86.4 | - |
| Bunny-3B [25] | P-2.7B | 384 | 79.8 | 62.5 | - | 70.9 | - | 1488.8 | 289.3 | 68.6 | - | 86.8 | 33.0 |
| Mipha-3B [99] | P-2.7B | 384 | 81.3 | 63.9 | 45.7 | 70.9 | 56.6 | 1488.9 | 295.0 | 69.7 | 32.1 | 86.7 | 32.5 |
| *Olympus* (Ours) | P-2.7B | 384 | 80.5 | 63.9 | 48.2 | 70.7 | 53.4 | 1520.7 | 283.2 | 71.2 | 33.8 | 86.6 | 32.8 |

Table 1. Multimodal evaluation across 11 benchmarks: VQAv2 [22], GQA [28], VisWiz [24], SQA$^I$: ScienceQA-IMG [52], VQA$^T$: TextVQA [65], MME-P: MME Perception [20], MME-C: MME Cognition [20], MMB: MMBench [50], MM-Vet [90], POPE [36] and MMMU [91]. "V", "L", "Q", "M" and "P" represent Vicuna [10], LLaMA [73], Qwen [5], MobileLLaMA [12] and Phi-2 [55]. Res. refers to the image resolution used by the visual backbone.

*Olympus* introduces a training-based paradigm where MLLMs act as controllers, addressing diverse tasks using both internal capabilities and external expert models. This general framework seamlessly integrates MLLMs with domain-specific models to tackle universal tasks.

## 4. Experiment

### 4.1. Experimental Setup

**MLLM Model**. Our model follows the setting of Mipha [99] with its vision and language encoders, i.e., SigLIP-384 [92] and Phi-2 [55]. For the multimodal projector, same as

LLaVA [48] and Mipha [99], we adopt a two-layer MLP.

**Training Setting.** We initialize the weights from Mipha-3B [99] and fine-tune the model on the LLaVA-Mix665K dataset [48] and *OlympusInstruct* for 2 epochs, using a learning rate of 5e-5 and a batch size of 256 on 64 V100 32GB GPUs. The whole training process takes approximately 24.8 hours. All model components, including the vision encoder, language encoder, and MLP, are fully fine-tuned during the training process.

**Evaluation Details.** We compare our method with a bunch of state-of-the-art multimodal large language models (MLLMs) across 11 popular benchmarks, as shown
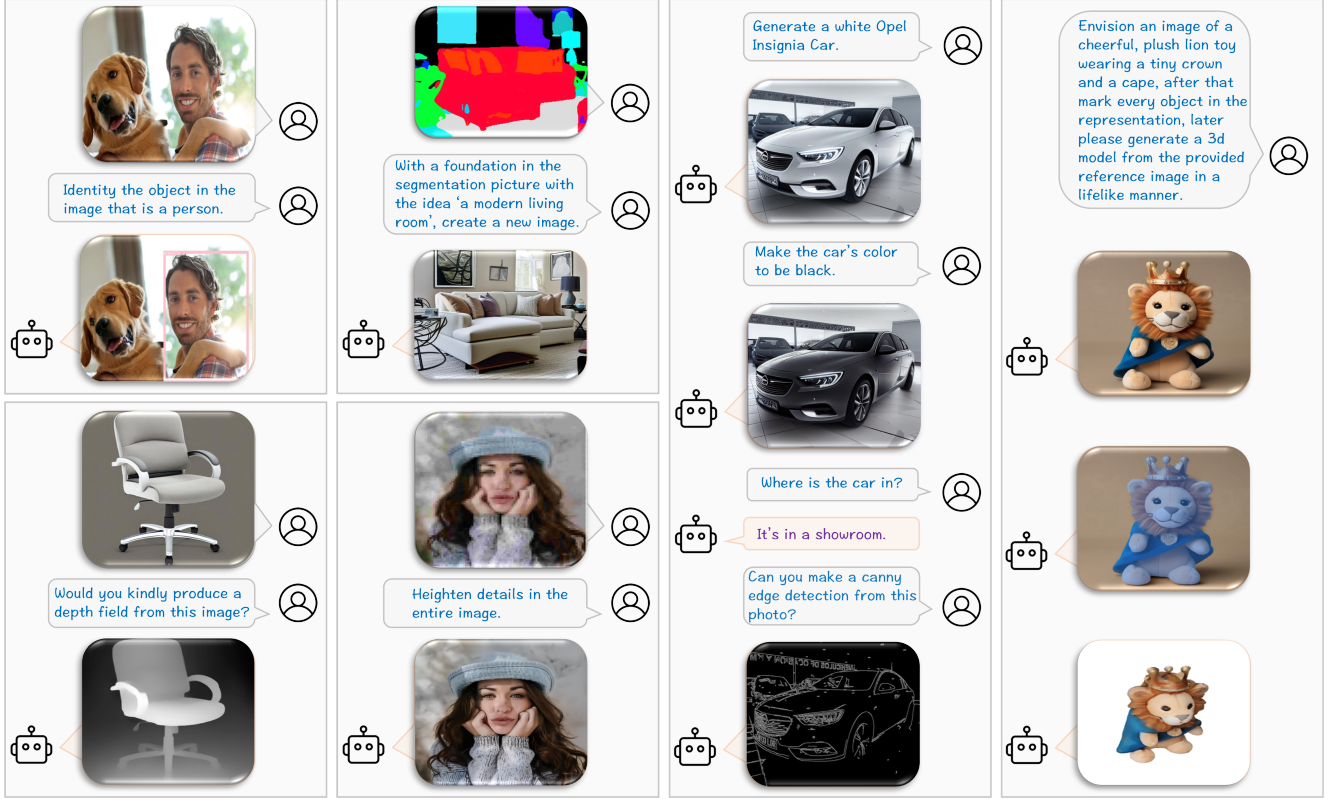
Figure 6. Diverse applications of *Olympus*. The **1st** and **2nd** columns show the scenarios for single task, the **3rd** column displays the results under multi-turn conversations, while **the last (4th)** column shows the chain-of-action capacity of *Olympus*.

| Method | Acc ↑ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| HuggingGPT (GPT-4o mini) | 70.14 | 76.51 | 72.14 | 75.46 |
| HuggingGPT (GPT-4o) | 81.35 | 85.54 | 81.55 | 83.56 |
| **Olympus (Ours)** | **94.75** | **95.80** | **94.75** | **95.77** |

Table 2. Evaluation results on *OlympusBench* under the single-task setting. Metrics include accuracy (%), precision (%), recall (%), and F1-score (%).

| Method | ED ↓ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| HuggingGPT (GPT-4o mini) | 0.45 | 65.14 | 48.51 | 53.14 |
| HuggingGPT (GPT-4o) | 0.35 | 75.03 | 60.23 | 61.25 |
| **Olympus (Ours)** | **0.18** | **91.82** | **92.75** | **91.98** |

Table 3. Evaluation results on *OlympusBench* under the chain-of-action setting. ED represents edit distance.

in Table 1. These benchmarks include VQA-v2 [22], GQA [28], ScienceQA-IMG [52], MME perception and cognition [20], MMBench [50], MM-Vet [90], TextVQA [65], and POPE [36], etc. For task routing performance, we evaluate accuracy, precision, recall, and F1 score, and for chain-of-action tasks, we also report edit distance [53], following HuggingGPT [62].

### 4.2. Quantitative Evaluation

**Task Routing Performance.** To demonstrate our routing effectiveness, we compare our results with HuggingGPT on *OlympusBench* using GPT-4o mini and GPT-4o models for

| Method | Success Rate ↑ |
|---|---|
| HuggingGPT (GPT-4o mini) | 65.8 |
| HuggingGPT (GPT-4o) | 75.2 |
| **Olympus (Ours)** | **86.5** |

Table 4. Human Evaluation of different methods, we display the results of success rate (%).

their strong predictive capabilities in Table 2 and 3. For a fair comparison, we included prompts covering all task types supported by *Olympus* and excluded prompts for irrelevant tasks. In Table 2, under the single-task setting, our method achieves notable improvements of **13.4%**, **10.26%**, **13.2%**, and **12.21%** in accuracy, precision, recall, and F1 score, even against the strong GPT-4o model. In the chain-of-action setting, *Olympus* demonstrates further gains of **0.17**, **16.79%**, **32.52%**, and **30.73%** for edit distance, precision, recall, and F1 score, respectively.

Additionally, we collected 200 diverse human-generated instructions to evaluate *Olympus*'s real-life performance against HuggingGPT, using success rate as the metric. The success rate reflects whether specialized models generate outputs that fully satisfy user requests, requiring both accurate task planning and effective task-tailored prompt generation. As shown in Table 4, *Olympus* outperforms HuggingGPT with an **11.3%** higher success rate using GPT-4o. These results highlight the significant potentials of *Olympus* and

| # of Tasks | VQAv2 | GQA | VisWiz | SQA$^I$ | VQA$^T$ | MME-P | MME-C | MMB | MM-Vet | POPE | MMMU |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 81.0 | 64.0 | 46.2 | 70.8 | 55.3 | 1498.3 | 293.2 | 70.1 | 32.6 | 86.6 | 32.4 |
| 5 | 80.5 | 64.2 | 45.6 | 70.9 | 53.5 | 1468.3 | 310.4 | 70.5 | 34.9 | 86.5 | 32.5 |
| 10 | 80.4 | 64.1 | 46.1 | 71.2 | 53.0 | 1546.7 | 333.9 | 70.2 | 33.8 | 86.2 | 32.9 |
| **20** | 80.5 | 63.9 | 48.2 | 70.7 | 53.4 | 1520.7 | 283.2 | 71.2 | 33.8 | 86.6 | 32.8 |

Table 5. The ablation study of varying the number of tasks on multimodal benchmarks. "0 task" denotes fine-tuning the model only using LLaVA-Mix665K dataset [45].

| # of tasks | Acc ↑ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| 5 | 96.38 | 96.36 | 96.45 | 97.61 |
| 10 | 96.15 | 95.85 | 96.23 | 97.07 |
| 15 | 95.84 | 95.78 | 95.84 | 96.79 |
| **20** | 94.75 | 95.80 | 94.75 | 95.77 |

Table 6. Ablation study of different numbers of tasks adopted for training under single-task setting.

| # of tasks | ED ↓ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| 5 | 0.12 | 93.23 | 94.32 | 93.35 |
| 10 | 0.14 | 92.23 | 93.45 | 92.28 |
| 15 | 0.17 | 91.97 | 92.89 | 92.01 |
| **20** | 0.18 | 91.82 | 92.75 | 91.98 |

Table 7. Ablation study of different numbers of tasks adopted for training under chain-of-action setting.

the collected *OlympusInstruct* dataset.

**Multimodal Understanding Performance.** Table 1 compares our method with existing approaches on multimodal benchmarks. *Olympus* achieves comparable performance to Mipha-3B on multiple benchmarks and even surpass it on 4 benchmarks such as VizWiz (**+2.5%**), MME-P (**+31.8**), MMB (**+1.5%**), and MM-Vet (**+1.7%**). While it shows a slight drop on TextVQA, *Olympus* uniquely supports model routing for 20 diverse vision tasks.

## 4.3. Ablation Study

The ablation study exploring the impact of varying the number of training tasks is presented in Tables 5, 6, 7, and illustrated in Figure 7. Table 5 demonstrates that the number of tasks has a limited influence on overall performance across multimodal benchmarks. Notably, the 10-task setting achieves the best results on MME-P [20], while the 20-task setting performs optimally on VizWiz [24]. The 20-task configuration is selected for its robustness and generality across a diverse range of tasks. Tables 6 and 7 illustrate a slight performance degradation in both single-task and chain-of-action settings as the number of tasks increases. This degradation is reasonably attributed to the increased prediction complexity associated with handling a larger number of tasks.

Figure 7 highlights the training cost, which increases from 1286.4 GPU hours without utilizing *OlympusInstruct* to 1589.5 GPU hours with it, representing a modest 23.6% increase in time. This relatively low cost increase is attributed to the avoidance of training complex generative models. Further experimental details are provided in the **Appendix**.

## 4.4. Visualization

Figure 6 illustrates the versatility of *Olympus* across various tasks. The first two columns present single-turn examples, including visual grounding, depth estimation, controllable image generation, and image super-resolution. The third column illustrates *Olympus*'s proficiency in executing a variety of tasks, such as image editing, visual question answering
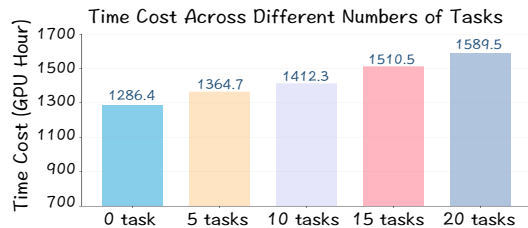


Figure 7. Training time cost for varying numbers of tasks.

(VQA), and canny edge detection, within the context of multi-turn conversations. This is particularly noteworthy given that *OlympusInstruct* does not include any multi-turn conversation data, underscoring our model's impressive capacity for generalization. The final column showcases its chain-of-action capability to conduct text-to-image generation, object segmentation and image-to-3D generation within one instruction. These examples clearly show that *Olympus* can handle diverse prompts for multiple tasks and generate comprehensive responses for users.

## 5. Limitation

Since *Olympus* is trained on the dataset collected through GPT-4o, it still has some limitations, e.g., the quality and diversity of the samples collected directly impact the performance of the generated responses. The inherent biases and inaccuracies in GPT-4o's responses propagate into MLLMs, potentially leading to suboptimal or biased outputs.

## 6. Conclusion

We present *Olympus*, a universal task router designed to address diverse computer vision tasks by integrating MLLM's internal abilities with task-specific routing to expert models. To achieve this, we introduce *OlympusInstruct* and *OlympusBench*, datasets collected from GPT-4o covering 20 distinct tasks. With the presented routing tokens, *Olympus* can handle multiple tasks within a single prompt, highlighting its potential as a robust foundation for unifying a wide range of computer vision tasks.

# References

[1] Emanuele Aiello, LILI YU, Yixin Nie, Armen Aghajanyan, and Barlas Oguz. Jointly training large autoregressive multimodal models. In *ICLR*, 2024. 3

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 2

[3] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 1, 2023. 2

[4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015. 1

[5] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966, 2023. 2, 6

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 14, 15

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, pages 1877–1901. Curran Associates, Inc., 2020. 2

[8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 6

[9] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21401–21412, 2024. 2

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023. 6

[11] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Oliveira Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. Palm: Scaling language modeling with pathways. 2022. 2

[12] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 2023. 3, 6

[13] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024. 6

[14] Marcos V Conde, Ui-Jin Choi, Maxime Burchi, and Radu Timofte. Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In *European Conference on Computer Vision*, pages 669–687. Springer, 2022. 15

[15] Marcos V Conde, Gregor Geigle, and Radu Timofte. High-quality image restoration following human instructions. *arXiv preprint arXiv:2401.16468*, 2024. 15

[16] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2

[17] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, Xiangwen Kong, Xiangyu Zhang, Kaisheng Ma, and Li Yi. DreamLLM: Synergistic multimodal comprehension and creation. In *ICLR*, 2024. 2, 3

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*. OpenReview.net, 2021. 3

[19] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified

flow transformers for high-resolution image synthesis. In *ICML*, 2024. 2

[20] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 6, 7, 8

[21] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024. 2, 3

[22] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 6, 7

[23] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14953–14962, 2023. 3

[24] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 6, 8

[25] Muyang He, Yexin Liu, Boya Wu, Jianhao Yuan, Yueze Wang, Tiejun Huang, and Bo Zhao. Efficient multimodal learning from data-centric perspective. *arXiv preprint arXiv:2402.11530*, 2024. 6

[26] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, pages 6840–6851, 2020. 1

[27] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeurIPS*, 2022. 2

[28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 6, 7

[29] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 2, 17

[30] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023. 15

[31] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 14

[32] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 3

[33] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36, 2024. 6

[34] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 2

[35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, pages 19730–19742, 2023. 2, 3

[36] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 6, 7

[37] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 2

[38] Yaobo Liang, Chenfei Wu, Ting Song, Wenshan Wu, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *Intelligent Computing*, 3:0063, 2024. 3

[39] Bin Lin, Zhenyu Tang, Yang Ye, Jiaxi Cui, Bin Zhu, Peng Jin, Junwu Zhang, Munan Ning, and Li Yuan. Moe-llava: Mixture of experts for large vision-language models. *arXiv preprint arXiv:2401.15947*, 2024. 6

[40] Yuanze Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. Revive: Regional visual representation matters in knowledge-based visual question answering. *Advances in Neural Information Processing Systems*, 35:10560–10571, 2022. 3

[41] Yuanze Lin, Chen Wei, Huiyu Wang, Alan Yuille, and Cihang Xie. Smaug: Sparse masked autoencoder for efficient video-language pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2459–2469, 2023. 2

[42] Yuanze Lin, Yi-Wen Chen, Yi-Hsuan Tsai, Lu Jiang, and Ming-Hsuan Yang. Text-driven image editing via learnable regions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7059–7068, 2024. 2

[43] Yuanze Lin, Ronald Clark, and Philip Torr. Dreampolisher: Towards high-quality text-to-3d generation via geometric diffusion. *arXiv preprint arXiv:2403.17237*, 2024. 2

[44] Yuanze Lin, Yunsheng Li, Dongdong Chen, Weijian Xu, Ronald Clark, Philip Torr, and Lu Yuan. Rethinking visual prompting for multimodal large language models with external knowledge. *arXiv preprint arXiv:2407.04681*, 2024. 2

[45] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pages 26296–26306, 2024. 6, 8, 14, 16

[46] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, 2024. 2

[47] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1

[48] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36, 2024. 2, 6, 14

[49] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 15

[50] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023. 6, 7

[51] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9970–9980, 2024. 15

[52] Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022. 6, 7

[53] Andres Marzal and Enrique Vidal. Computation of normalized edit distance and applications. *IEEE transactions on pattern analysis and machine intelligence*, 15(9):926–932, 1993. 7

[54] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024. 2

[55] Microsoft. Phi-2: The surprising power of small language models, 2023. 6

[56] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1

[57] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. *CoRR*, abs/2306.14824, 2023. 2

[58] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 1, 2, 14, 15

[59] Yujia Qin, Shengding Hu, Yankai Lin, Weize Chen, Ning Ding, Ganqu Cui, Zheni Zeng, Yufei Huang, Chaojun Xiao, Chi Han, Yi Ren Fung, Yusheng Su, Huadong Wang, Cheng Qian, Runchu Tian, Kunlun Zhu, Shihao Liang, Xingyu Shen, Bokai Xu, Zhen Zhang, Yining Ye, Bowen Li, Ziwei Tang, Jing Yi, Yuzhang Zhu, Zhenning Dai, Lan Yan, Xin Cong, Yaxi Lu, Weilin Zhao, Yuxiang Huang, Junxi Yan, Xu Han, Xian Sun, Dahai Li, Jason Phang, Cheng Yang, Tongshuang Wu, Heng Ji, Zhiyuan Liu, and Maosong Sun. Tool learning with foundation models, 2023. 3

[60] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems*, 35:36479–36494, 2022. 2

[61] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[62] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 7, 17

[63] Min Shi, Fuxiao Liu, Shihao Wang, Shijia Liao, Subhashree Radhakrishnan, De-An Huang, Hongxu Yin, Karan Sapra, Yaser Yacoob, Humphrey Shi, et al. Eagle: Exploring the design space for multimodal llms with mixture of encoders. *arXiv preprint arXiv:2408.15998*, 2024. 2

[64] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[65] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 6, 7

[66] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Emu: Generative pretraining in multimodality. In *ICLR*, 2023. 3

[67] Yi-Lin Sung, Linjie Li, Kevin Lin, Zhe Gan, Mohit Bansal, and Lijuan Wang. An empirical study of multimodal model merging. *arXiv preprint arXiv:2304.14933*, 2023. 6

[68] Dídac Surís, Sachit Menon, and Carl Vondrick. Vipergpt: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11888–11898, 2023. 3

[69] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*, pages 1–18. Springer, 2025. 14, 15

[70] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *NeurIPS*, 36, 2024. 2, 3

[71] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024. 2, 3

[72] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 2

[73] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023. 2, 3, 6

[74] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024. 2, 3

[75] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 3

[76] Junfeng Wu, Yi Jiang, Qihao Liu, Zehuan Yuan, Xiang Bai, and Song Bai. General object foundation model for images and videos at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3783–3795, 2024. 15

[77] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 2

[78] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023. 2, 3

[79] Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. *arXiv preprint arXiv:2409.11340*, 2024. 2, 3

[80] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 15

[81] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024. 2, 3

[82] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xgen-mm (blip-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024. 2

[83] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 15

[84] Shuai Yang, Yuying Ge, Yang Li, Yukang Chen, Yixiao Ge, Ying Shan, and Yingcong Chen. Seed-story: Multimodal long story generation with large language model. *arXiv preprint arXiv:2407.08683*, 2024. 3

[85] Zhendong Yang, Ailing Zeng, Chun Yuan, and Yu Li. Effective whole-body pose estimation with two-stages distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4210–4220, 2023. 15

[86] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 15

[87] Hanrong Ye, De-An Huang, Yao Lu, Zhiding Yu, Wei Ping, Andrew Tao, Jan Kautz, Song Han, Dan Xu, Pavlo Molchanov, et al. X-vila: Cross-modality alignment for large language model. *arXiv preprint arXiv:2405.19335*, 2024. 2

[88] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. *arXiv preprint arXiv:2311.04257*, 2023. 6

[89] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023. 3

[90] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023. 6, 7

[91] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024. 6

[92] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986, 2023. 6

[93] Jun Zhan, Junqi Dai, Jiasheng Ye, Yunhua Zhou, Dong Zhang, Zhigeng Liu, Xin Zhang, Ruibin Yuan, Ge Zhang, Linyang Li, et al. Anygpt: Unified multimodal llm with discrete sequence modeling. *arXiv preprint arXiv:2402.12226*, 2024. 3

[94] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 15

[95] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. 6

[96] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe

Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024. 2, 3

[97] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *CoRR*, abs/2304.10592, 2023. 2

[98] Jinguo Zhu, Xiaohan Ding, Yixiao Ge, Yuying Ge, Sijie Zhao, Hengshuang Zhao, Xiaohua Wang, and Ying Shan. VL-GPT: A generative pre-trained transformer for vision and language understanding and generation. *CoRR*, abs/2312.09251, 2023. 3

[99] Minjie Zhu, Yichen Zhu, Xin Liu, Ning Liu, Zhiyuan Xu, Chaomin Shen, Yaxin Peng, Zhicai Ou, Feifei Feng, and Jian Tang. Mipha: A comprehensive overhaul of multimodal assistant with small language models. *CoRR*, 2024. 3, 6

[100] Yichen Zhu, Minjie Zhu, Ning Liu, Zhicai Ou, Xiaofeng Mou, and Jian Tang. Llava-phi: Efficient multi-modal assistant with small language model. *arXiv preprint arXiv:2401.02330*, 2024. 2, 6

[101] Zhuofan Zong, Guanglu Song, and Yu Liu. Detrs with collaborative hybrid assignments training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6748–6758, 2023. 15

## A. Appendix

In the supplementary materials, we provide the following sections:

## B. Training Details

We train our models using $64 \times$ V100 GPUs, each equipped with 32GB of memory. The Adam optimizer [31] is employed, combined with a cosine learning rate scheduler, aligning with the configuration utilized in LLaVA [48]. For fine-tuning, we set a learning rate of 5e-5, which is optimized for stability and convergence, and adopt a batch size of 256 to accommodate the large-scale data and distributed training setup. The training process spans two epochs over the combined fine-tuning datasets of LLaVA-Instruct-158K [45] and *OlympusInstruct*, ensuring that the models are effectively exposed to both general-purpose and task-specific instructions.

Additional training configurations include a warmup ratio of 0.03, which helps in stabilizing the initial training phase, and gradient accumulation steps set to 4 to balance memory efficiency with gradient updates. To handle varying image sizes in the datasets, we employ a padding-based image aspect ratio strategy. Moreover, numerical precision is set to float16, enabling faster computation and reduced memory usage while maintaining sufficient numerical accuracy. These hyperparameters, summarized in Table 8, were meticulously selected to optimize training performance and ensure scalability across diverse tasks.

| Configuration | MLLMs Traning |
|---|---|
| Optimizer | Adam |
| Learning rate | 5e-5 |
| Learning rate schedule | cosine |
| Total training epochs | 2 |
| Weight Decay | 0 |
| Warmup ratio | 0.03 |
| Batch size | 256 |
| Gradient Accumulation Steps | 4 |
| Imgae Aspect Ratio | Pad |
| Numerical precision | `float16` |

Table 8. Summary of training hyperparameters of *Olympus*.

## C. Task-specific Routing Tokens

As illustrated in Figure 8, we present the task-specific routing tokens for 20 distinct computer vision tasks, spanning image, video, and 3D domains. These routing tokens play a crucial role during the training of MLLMs on *OlympusInstruct*, acting as explicit indicators to guide task-specific responses. For instance, when handling a text-to-3D generation task, a sample instruction such as: `I'd appreciate it if you could design a 3D representation of an ancient library, which is a repository of books and scrolls from ancient times.`", paired with the response: `ancient library, a repository of books and scrolls from ancient times.`", can be augmented with the routing tokens corresponding to the text-to-3D generation task. This updated response would then appear as: "`<3D_gen_text>ancient library, a repository of books and scrolls from ancient times.</3D_gen_text>`". Such augmentation ensures that the model learns to associate specific tasks with their respective routing tokens.

By incorporating these routing tokens into the training process, the MLLMs are endowed with the ability to predict and append the appropriate tokens based on diverse user instructions during inference. This mechanism enables the invocation of the most relevant specialist models for a given task, such as `<image_edit>` for image editing or `<video_ref_seg>` for referring video object segmentation. The framework's modularity not only enhances task alignment but also ensures adaptability across evolving domains, facilitating the seamless integration of new tasks and specialist models in the future. This mechanism underscores the scalability and versatility of the *Olympus* framework in handling complex AI tasks.

## D. Adopted Specialist Models

In Table 9, we present the specialist models selected for 20 distinct computer vision and multimodal tasks, illustrating the flexibility and adaptability of our *Olympus* framework. Notably, for tasks like canny estimation, we utilize the highly efficient and widely recognized Canny operator from the OpenCV library. Similarly, for other tasks such as image generation, image editing, and text-to-3D generation, we incorporate state-of-the-art models like Stable Diffusion XL [58], InstructPix2Pix [6], and LGM [69], respectively. By leveraging these specialized, task-specific models, *Olympus* circumvents the need for training excessively large and cumbersome multimodal all-in-one models, instead opting for a modular and scalable approach.

A key strength of the *Olympus* framework lies in its ability to seamlessly integrate superior models as they become available. For instance, advanced models like Ground-

| Task-Specific Routing Tokens (20 tasks) |
|---|

Image Generation: <image_gen></image_gen>  Video Generation:
Image Editing:  Text-to-3D Generation: <3D_gen_text></3D_gen_text>
Video Editing:  Image-to-3D Generation: <3D_gen_image></3D_gen_image>
Image Deblurring:  RVOS:
Object Detection:  Image Super-Resolution:
Normal Estimation:  Pose Estimation:
Canny Estimation:  Depth Estimation:
Visual Grounding:  Image Deraining:
Image Denosing: <image_denosie></image_denoise>  Object Segmentation:

**Controllable Image Generation:**
Pose Condition:  Normal Condition:
Canny Condition:  Segmentation Condition:
Depth Condition:  Scribble Condition:

**Controllable Video Generation:**
Pose Condition:  Normal Condition:
Canny Condition:  Segmentation Condition:
Depth Condition:  Scribble Condition:

Figure 8. Task-specific routing tokens for 20 diverse tasks, covering image, video and 3D domains. Note that "RVOS" denotes referring video object segmentation.

| Specialists for 20 individual computer tasks | |
|---|---|
| **Task / Model** | **Task / Model** |
| Image Generation: Stable Diffusion XL [58] | Video Generation: CogVideoX [86] |
| Image Editing: InstructPix2Pix [6] | Text-to-3D Generation: LGM [69] |
| Video Editing: Text2Video-Zero [30] | Image-to-3D Generation: Wonder3D [51] |
| Image Deblurring: InstructIR [15] | Referring Video Object Segmentation: GLEE [76] |
| Object Detection: Co-DETR [101] | Image Super-Resolution: Swin2SR [14] |
| Normal Estimation: Sapiens [76] | Pose Estimation: DWPose [85] |
| Canny Estimation: OpenCV Canny Operator | Depth Estimation: Depth Anything V2 [83] |
| Visual Grounding: GroundingDINO [49] | Image Deraining: InstructIR [15] |
| Image Denoising: InstructIR [15] | Object Segmentation: SegFormer [80] |
| Controllable Image Generation: ControlNet [94] | Controllable Video Generation: Text2Video-Zero [30] |

Table 9. Specified models to solve 20 different tasks.

ingDINO [49] for visual grounding or Wonder3D [51] for image-to-3D generation can be directly adopted to replace existing specialists, ensuring up-to-date performance across tasks. This modularity enables *Olympus* to remain efficient and user-oriented, adapting to specific requirements without the overhead of comprehensive retraining. By selectively incorporating these external models, *Olympus* provides a practical and scalable solution to address diverse and evolving user needs in various AI applications.

## E. Dataset Statistic

We provide a comprehensive breakdown of the statistics for the *OlympusInstruct* and *OlympusBench* datasets in Ta-

ble 10. Specifically, the number of instruction-response pair samples is listed for each task. For example, we collected 45,000 and 5,000 samples for text-guided image generation on *OlympusInstruct* and *OlympusBench*, respectively. Similarly, video generation tasks include 35,786 and 3,976 samples, reflecting the diversity and scale of these datasets. Across various image and video processing tasks, such as editing, segmentation, and deblurring, we maintained a balanced distribution to ensure comprehensive task coverage.

For controllable image generation (CIG) and controllable video generation (CVG), we collected balanced samples across six specific conditions: pose, canny, normal, scribble, segmentation, and depth. Each condition is represented with

| # of OlympusInstruct/OlympusBench across different tasks | | | |
|---|---|---|---|
| **Task** | **# of samples** | **Task** | **# of samples** |
| Image Generation | 45000 / 5000 | Video Generation | 35786 / 3976 |
| Image Editing | 34927 / 3880 | Text-to-3D Generation | 29250 / 3250 |
| Video Editing | 32227 / 3580 | Image-to-3D Generation | 10800 / 1200 |
| Image Deblurring | 6300 / 700 | Referring Video Segmentation | 21600 / 2400 |
| Object Detection | 7200 / 800 | Image Super-Resolution | 6300 / 700 |
| Normal Estimation | 6300 / 700 | Pose Estimation | 6300 / 700 |
| Canny Estimation | 6300 / 700 | Depth Estimation | 6300 / 700 |
| Visual Grounding | 23040 / 2560 | Image Denoising | 7574 / 841 |
| Image Deraining | 7650 / 850 | Object Segmentation | 7200 / 800 |
| Pose-to-Image (CIG) | 5946 / 658 | Canny-to-Image (CIG) | 5950 / 658 |
| Normal-to-Image (CIG) | 5896 / 658 | Scribble-to-Image (CIG) | 5949 / 658 |
| Segmentation-to-Image (CIG) | 5926 / 658 | Depth-to-Image (CIG) | 5923 / 658 |
| Pose-to-Video (CVG) | 7650 / 850 | Canny-to-Video (CVG) | 7650 / 850 |
| Normal-to-Video (CVG) | 7650 / 850 | Scribble-to-Video (CVG) | 7650 / 850 |
| Segmentation-to-Video (CVG) | 7650 / 850 | Depth-to-Video (CVG) | 7650 / 850 |
| Chain-of-Action | 64800 / 7200 | Total | 446344 / 49585 |

Table 10. The number of collected samples for each task on *OlympusInstruct*/*OlympusBench*. "CIG" and "CVG" denotes controllable image generation and controllable video generation, respectively.

| MLLM | VQAv2 | GQA | VisWiz | SQA$^I$ | VQA$^T$ | MME-P | MME-C | MMB | MM-Vet | POPE | MMMU |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Mipha-3B | 81.3 | 63.9 | 45.7 | 70.9 | 56.6 | 1488.9 | 295.0 | 69.7 | 32.1 | 86.7 | 32.5 |
| **Olympus-3B** | 80.5 | 63.9 | 48.2 | 70.7 | 53.4 | 1520.7 | 283.2 | 71.2 | 33.8 | 86.6 | 32.8 |
| LLava-7B | 78.5 | 62.0 | 50.0 | 66.8 | 58.2 | 1510.7 | 316.1 | 64.3 | 30.5 | 85.9 | 32.0 |
| **Olympus-7B** | 78.3 | 61.9 | 52.3 | 67.5 | 57.9 | 1460.2 | 328.9 | 65.6 | 32.0 | 85.4 | 32.1 |
| LLava-13B | 80.0 | 63.3 | 53.6 | 71.6 | 61.3 | 1531.3 | 295.4 | 67.7 | 36.1 | 85.9 | 33.6 |
| **Olympus-13B** | 79.8 | 63.1 | 56.1 | 71.9 | 61.0 | 1502.3 | 335.2 | 68.9 | 36.8 | 85.6 | 33.7 |

Table 11. The ablation study of varying the number of tasks on multimodal benchmarks. "0 task" denotes fine-tuning the model only using LLaVA-Mix665K dataset [45].

approximately equal proportions to ensure robust model performance under diverse constraints. For example, pose-to-image generation includes 5,946 samples in *OlympusInstruct* and 658 in *OlympusBench*, while pose-to-video generation contains 7,650 and 850 samples, respectively.

In total, we collected 446,344 examples for *OlympusInstruct* to serve as a comprehensive training dataset and 49,585 examples for *OlympusBench* to support rigorous evaluation.

**Chain-of-Action Samples.** To simulate complex, sequential tasks, we curated instruction-response pairs for chain-of-action scenarios. Here, $N$ random tasks (ranging from 2 to 5) were selected, and one sample from each task was combined to construct multi-step instruction-response pairs. This approach resulted in 64,800 and 7,200 samples for chain-of-action tasks in *OlympusInstruct* and *OlympusBench*, respectively. These chain-of-action samples are designed to assess the model's ability to handle multi-step and sequential

| Method | Acc ↑ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| HuggingGPT (GPT-4o mini) | 70.14 | 76.51 | 72.14 | 75.46 |
| HuggingGPT (GPT-4o) | 81.35 | 85.54 | 81.55 | 83.56 |
| **Olympus-3B** | 94.75 | 95.80 | 94.75 | 95.77 |
| **Olympus-7B** | 95.63 | 96.71 | 95.63 | 96.62 |
| **Olympus-13B** | 96.71 | 97.65 | 96.71 | 96.74 |

Table 12. Evaluation results on *OlympusBench* under the single-task setting. Metrics include accuracy (%), precision (%), recall (%), and F1-score (%).

tasks within one user instruction effectively, showcasing the versatility and adaptability of the *Olympus* framework.

# F. Ablation Study

In this section, we conduct more ablation study experiments to provide deeper insight into the effect of varying MLLMs and training epochs for *Olympus*.

| Method | ED ↓ | Pre ↑ | Recall ↑ | F1 ↑ |
|---|---|---|---|---|
| HuggingGPT (GPT-4o mini) | 0.45 | 65.14 | 48.51 | 53.14 |
| HuggingGPT (GPT-4o) | 0.35 | 75.03 | 60.23 | 61.25 |
| **Olympus-3B** | 0.18 | 91.82 | 92.75 | 91.98 |
| **Olympus-7B** | 0.14 | 93.11 | 93.89 | 93.15 |
| **Olympus-13B** | 0.12 | 94.54 | 95.02 | 94.58 |

Table 13. Evaluation results on *OlympusBench* under the chain-of-action setting. ED represents edit distance.

| Method | Success Rate ↑ |
|---|---|
| HuggingGPT (GPT-4o mini) | 65.8 |
| HuggingGPT (GPT-4o) | 75.2 |
| **Olympus* (Ours)** | 80.1 |
| **Olympus (Ours)** | 86.5 |

Table 14. Human evaluation of different methods, we display the results of success rate (%). Olympus-3B is selected for evaluation, and Olympus* denotes training the model with coarse responses.

| Example Pairs | Prefixes | Phrases | Complexities | Success Rate ↑ |
|---|---|---|---|---|
| ✗ | ✗ | ✗ | ✗ | 44.3 |
| ✓ | ✗ | ✗ | ✗ | 61.8 |
| ✓ | ✓ | ✗ | ✗ | 68.2 |
| ✓ | ✓ | ✓ | ✗ | 76.4 |
| ✓ | ✓ | ✓ | ✓ | 86.5 |

Table 15. Ablation study of different prompt components. "Complexities" and "Example Pairs" denote defined user instruction complexities and given user instruction-response pairs.

**The Impact of Adopting Different MLLMs.** To demonstrate the generality and robustness of the proposed Olympus framework, we utilize the LLava-7B and LLava-13B models as baselines, alongside Mipha-3B. These models are fine-tuned under our framework to produce Olympus-3B, Olympus-7B, and Olympus-13B, respectively. The results, summarized in Table 11, indicate that our Olympus models achieve performance on par with or superior to their original baselines across several benchmarks. For instance, Olympus-7B demonstrates significant improvements when compared to LLava-7B, achieving higher performance on VisWiz (+2.3%), MME-C (+12.8), MM-Vet (+1.5%) and MMB (+1.3%), with only a slight drop in MME-P and minimal reductions across other benchmarks. These results highlight the effectiveness of our framework in adapting existing models across multimodal tasks. Interestingly, Olympus-3B outperforms its larger counterparts, Olympus-7B and Olympus-13B, on specific benchmarks. This can be attributed to the higher input image resolution and a more robust visual encoder inherited from Mipha-3B, emphasizing the role of architectural design and input quality in determining performance.

Furthermore, as detailed in Tables 12 and 13, Olympus-3B, Olympus-7B, and Olympus-13B achieve superior routing performance on *OlympusBench* compared to Hugging-

GPT [62]. In the single-task setting, Olympus models consistently exhibit higher accuracy, precision, recall, and F1 scores, with Olympus-13B achieving the best overall performance (e.g., 96.71% accuracy and 96.74% F1 score). Similarly, in the chain-of-action setting, Olympus models significantly reduce edit distance (ED) while maintaining high precision, recall, and F1 scores, with Olympus-13B again leading (e.g., 0.12 ED and 94.58% F1 score). These results underscore the versatility of Olympus across diverse models and parameter scales, cementing its potential as a powerful framework for task routing.

**The Effect of Different Response Designs.** By default, we prompt GPT-4o [29] to generate concise and practical responses for each user instruction. To investigate the influence of different response designs, we also evaluate the performance of coarse responses, where the responses directly replicate the user instructions without simplification. The trained model using coarse responses is named Olympus*. For comparison, we report results on 200 collected human-generated instructions, as shown in Table 14. The evaluation metric used is the success rate, which measures whether the specialized models generate outputs that fully satisfy user requests. As shown in Table 14, Olympus*, which adopts coarse responses, achieves a success rate of 80.1%, while Olympus, utilizing concise and practical responses, achieves a significantly higher success rate of 86.5%. This improvement highlights the advantage of practical responses, which are concise, direct, and focused on effectively addressing user requirements. In contrast, coarse responses, although diverse in styles, tones, and complexities, often introduce ambiguities or redundancies that can hinder the model's ability to align with user intent.

In addition, HuggingGPT (GPT-4o mini and GPT-4o) exhibits lower success rates, further underscoring the advantage of our approach for improved task routing performance.

**The Influence of Individual Prompt Components.** To better understand the contribution of various components in task-specific prompts used for generating instruction-response pairs with GPT-4o, we conducted a detailed ablation study. The results, presented in Table 15, highlight the incremental impact of including key elements such as example pairs, prefixes, phrases, varying complexity levels. Starting with a baseline configuration that excludes all these components, we observe a success rate of 44.3%. Incorporating each individual component progressively improves performance, with the inclusion of user instruction-response example pairs alone yielding an 17.5% gain, reaching 61.8%. Adding prefixes further elevates the success rate to 68.2%, while introducing phrases improves it to 76.4%. Finally, the inclusion of complexities, alongside all the aforementioned components, results in a significant improvement, achieving

| # of prefixes | Success Rate ↑ |
|---|---|
| 0 | 80.9 |
| 3 | 84.0 |
| 7 | 86.5 |
| 10 | 86.1 |

(a) different numbers of prefixes.

| # of phrases | Success Rate ↑ |
|---|---|
| 0 | 78.7 |
| 5 | 83.2 |
| 12 | 86.5 |
| 16 | 86.4 |

(b) different numbers of phrases

| # of example pairs | Success Rate ↑ |
|---|---|
| 0 | 69.6 |
| 3 | 80.3 |
| 9 | 86.5 |
| 12 | 86.3 |

(c) different numbers of example pairs

Table 16. Ablation study of varying numbers of prefixes, phrases and user instruction-response example pairs in task-specific prompts.

the highest success rate of 86.5%. These findings underscore the synergistic role of these components in enhancing the quality and effectiveness of task-specific prompt designs.

**Varying Numbers of Prefixes, Phrases and Example Pairs.** In Table 16, we present an ablation study exploring the impact of varying the numbers of prefixes and phrases in task-specific prompts by using the success rate as the evaluation metric. For prefixes, as shown in Table 16 (a), increasing the number of prefixes from 0 to 7 results in a steady improvement in success rate, with the best performance achieved at 7 prefixes (86.5%). However, adding more prefixes beyond this point (e.g., 10) slightly reduces performance (86.1%), indicating a diminishing return or possible overfitting when too many prefixes are used.

Table 16 (b) analyzes the effect of using different numbers of phrases. The success rate increases significantly from 0 to 12 phrases, achieving a peak performance of 86.5%, comparable to the optimal prefix setting. However, increasing the number of phrases to 16 results in a marginal decrease (86.4%), again suggesting that excessive phrases may not necessarily improve performance further.

Similarly, Table 16 (c) examines the role of different numbers of instruction-response example pairs. Note that for different settings, the numbers of short, moderate, and extended examples remain consistent. For instance, if the number of sample pairs equals 3, there is 1 short, 1 moderate, and 1 extended example pair. The results reveal that introducing example pairs substantially boosts the success rate, from 69.6% with no pairs to 86.5% with 9 pairs. Interestingly, increasing the number of pairs to 12 leads to a minor performance drop (86.3%), which may indicate a saturation effect where additional examples no longer provide significant benefit. The final prompt with 9 example pairs to generate user instruction-response pairs for the image editing task has been displayed in Figure 9.

These findings highlight the need for carefully calibrated choices of prefixes, phrases, and example pairs to achieve optimal task-specific prompt design, balancing informativeness and efficiency without overcomplicating the inputs.

**The Complexities of User Instructions.** As shown in Table 17, we analyze the impact of varying instruction complexities in task-specific prompts on the model's success

| Complexities | Success Rate ↑ |
|---|---|
| ✗ | 76.8 |
| S | 77.1 |
| S+M | 83.1 |
| S+M+E | 86.5 |

Table 17. The ablation study of different complexity levels defined in the task-specific prompts. "S", "M" and "E" represent short, moderate and extended complexities respectively. Note that the term ✗ in the "Complexities" column indicates the absence of any complexity definitions in the task-specific prompts.

rate. By default, without any specific complexity definitions, the baseline achieves a success rate of 76.8%. Introducing the definition of short (S) instruction complexity yields a negligible improvement, raising the success rate to 77.1%. When moderate (M) complexities are added alongside short ones (S+M), the success rate rises significantly to 83.1%, demonstrating the benefits of incorporating greater linguistic diversity and intermediate complexity. Finally, further defining "extended" (E) instruction complexity leads to the model achieving its highest performance, with short, moderate, and extended instructions (S+M+E) resulting in a success rate of 86.5%. This progression underscores the critical role of instruction complexity in enhancing the model's ability to effectively interpret and execute user requests.

# G. More Results

We present additional application results in Figure 10, showcasing the versatility of our approach across a wide range of tasks. The first and second columns demonstrate controllable image generation conditioned on diverse inputs such as depth maps, poses, scribbles, and surface normals, illustrating the model's flexibility to adapt to varying conditions. The third column highlights advanced applications like object detection and image deblurring, emphasizing the framework's utility in real-world image processing. Lastly, the fourth column explores cutting-edge tasks, including text-guided image and video generation and object segmentation, underscoring the model's potential for multimodal and domain-specific applications.

In Figure 11, the first column illustrates image and video editing tasks, such as adding flowers beside a cat or brightening a sky. The second column highlights image-to-3D

generation and image deraining, including transforming a 2D car image into a 3D model and removing rain effects to improve clarity. The third column focuses on image deblurring, such as removing motion blur from a car photo, and depth estimation, demonstrated by generating depth maps from images. The fourth column features diverse applications, including text-to-3D generation (e.g., creating intricate 3D shapes from a detailed prompt for "a butterfly"), editing environmental lighting (e.g., converting night to day), and visual question answering (VQA), where image content questions, such as identifying unique rock formation features, are answered with precision.

These results demonstrate the strong generality and scalability of *Olympus*, highlighting its adaptability and effectiveness across a broad and growing set of tasks.

**Task-Specific Prompt**

Generate 50 unique user instructions with paired responses for image editing task focused on modifying objects and backgrounds. These instructions should vary significantly in language style, tone, and complexity to capture authentic interactions. Each instruction should range from short instruction into moderate and even extended instructions with multi-sentence descriptions. Occasionally use prefixes to diversify each instruction's phrasing. Responses must be extremely short, direct and practical, effectively addressing each instruction. Refer to the samples below as guidance, and come up with a wide range of prefixes and phrases to develop short, moderate and extended user instructions.

<u>Prefixes:</u>
1. Is there a way to. 2. Is it possible to. 3. I would appreciate it if you could. 4. Would you. 5. Could you. 6. Let's. 7. Please.

<u>Phrases:</u>
1. replace [object] with [new object]. 2. turn [object] into [new object]. 3. swap [object] with [new object]. 4. erase [object].
5. let [object] [action]. 6. change [object] to [new object]. 7. remove [object]. 8. add [new object] to [object].
9. place [new object] beside [object]. 10. make [object] [action]. 11. have [object] [action]. 12. leave [object] on [new object].

<u>User Instruction Complexities:</u>
1. Short Instruction: One-sentence requests using concreate objects and actions. 2. Moderate Instruction: Two sentences, adding context for clarity. 3. Extended Instruction: Over three sentences, describing emotions, effects, or scenes for richer context.

<u>User Instruction-Response Pairs:</u>
Short Instruction: Is there a way to replace the mountain with a beach?
- Response: Replace the mountain with a beach.
Short Instruction: Could you swap the sky with a starry night?
- Response: Swap the sky with a starry night.
Short Instruction: Is it possible to add a rainbow over the waterfall?
- Response: Add a rainbow over the waterfall.
Moderate Instruction: Help me change dragonfly to spaceship. I believe this change will give a more inviting feel.
- Response: Change dragonfly to spaceship.
Moderate Instruction: Let's turn the field into a flower meadow; it would add more color.
- Response: Turn the field into a flower meadow.
Moderate Instruction: I would appreciate it if you could add a flock of sheep grazing in the meadow. It can bring life to the scene.
- Response: Add a flock of sheep to the meadow.
Extended Instruction: Please replace the old fence in the background with a modern wooden one. The style should be clean and natural-looking, fitting in with a suburban backyard. I'm envisioning a style that's clean and simple, with natural tones that blend seamlessly into a cozy suburban backyard setting. The new fence should add a warm, inviting feel to the space, creating a sense of privacy without feeling closed off.
- Response: Replace the old fence with a modern, clean and natural-looking wooden fence.
Extended Instruction: Would you transform the barren tree into a blossoming cherry blossom tree? This alteration would add vibrant colors and a sense of renewal to the scene, making it more inviting and picturesque. The pink blossoms should contrast beautifully against the green foliage, enhancing the overall aesthetic.
- Response: Transform the barren tree into a blossoming cherry blossom tree.
Extended Instruction: It would be great if you can erase the power lines from the image to preserve the natural beauty of the countryside. Removing these distractions would make the scene more tranquil and undisturbed, allowing the focus to remain on the scenic elements like trees and hills. This cleanup would enhance the overall aesthetic and make the landscape appear more pristine.
- Response: Erase the power lines from the image.

Figure 9. The final prompt used to generate user instruction-response pairs for image editing in our experiments.
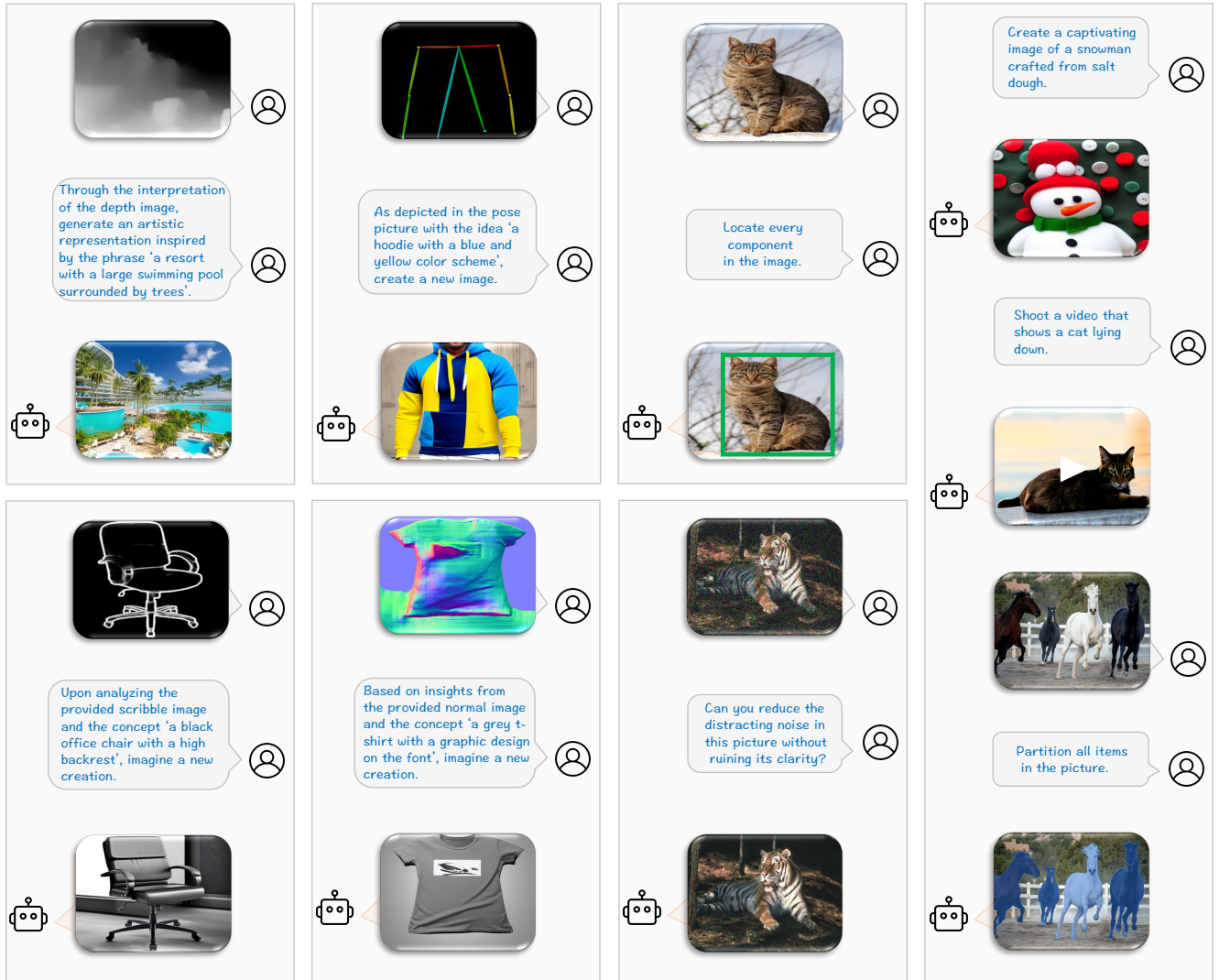
Figure 10. Diverse applications of *Olympus*. The first and second columns denote controllable image generation conditioning on depth, pose, scribble and normal. The third column represent the applications of object detection and image deblurring. The last column represent the applications of text-guided image and video generation, and object segmention.
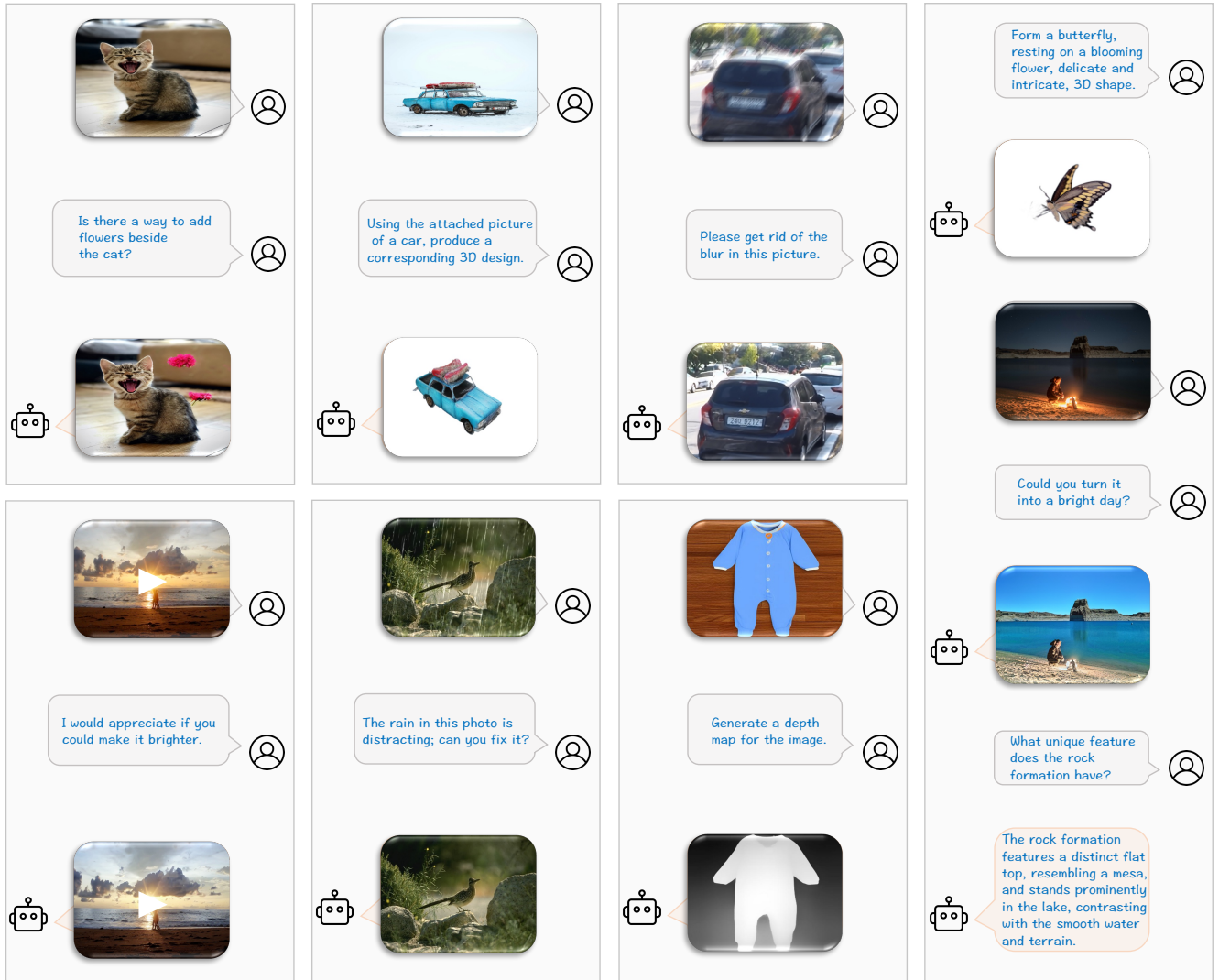
Figure 11. Diverse applications of *Olympus*. The first column represents image and video editing, the second column contains the examples of image-to-3D generation and image deraining, the third column denotes image deblurring and depth estimation, and the final column displays the applications of text-to-3D generation, image editing and visual question answering (VQA).