# VITRON: A Unified Pixel-level Vision LLM for Understanding, Generating, Segmenting, Editing

**Hao Fei**[1,2] **Shengqiong Wu**[1,2] **Hanwang Zhang**[1,3] **Tat-Seng Chua**[2] **Shuicheng Yan**[1,*]
[1,*]Skywork AI, Singapore [2] National University of Singapore [3] Nanyang Technological University
haofei37@nus.edu.sg swu@u.nus.edu hanwangzhang@ntu.edu.sg
dcscts@nus.edu.sg shuicheng.yan@kunlun-inc.com

**Project Homepage:** https://vitron-llm.github.io/

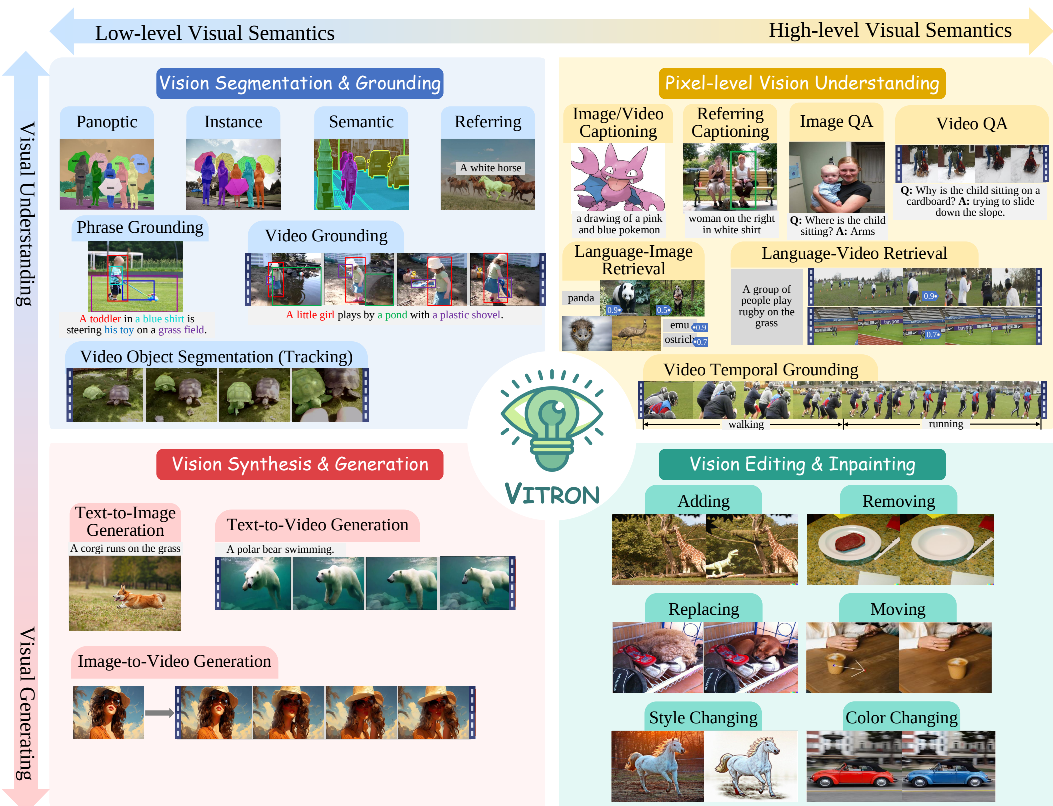



Figure 1: VITRON supports four main task clusters of visions, spanning visual comprehension to visual generation, from low level to high level.

## Abstract

Recent developments of vision large language models (LLMs) have seen remarkable progress, yet still encounter challenges towards multimodal generalists, such as coarse-grained instance-level understanding, lack of unified support for both images and videos, and insufficient coverage across various vision tasks. In this paper,

*Shuicheng Yan is the corresponding author. This work was performed when Hao Fei was an Associate Member, and Shengqiong Wu was an Intern at Skywork AI.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

we present **VITRON**, a universal pixel-level vision LLM designed for comprehensive *understanding*, *generating*, *segmenting*, and *editing* of both static images and dynamic videos. Building on top of an LLM backbone, VITRON incorporates encoders for images, videos, and pixel-level regional visuals within its frontend modules, while employing state-of-the-art visual specialists as its backend, via which VITRON supports a spectrum of vision end tasks, spanning visual comprehension to visual generation, from low level to high level. To ensure an effective and precise message passing from LLM to backend modules for function invocation, we propose a novel hybrid method by simultaneously integrating discrete textual instructions and continuous signal embeddings. Further, we design various pixel-level spatiotemporal vision-language alignment learning for VITRON to reach the best fine-grained visual capability. Finally, a cross-task synergy module is advised to learn to maximize the task-invariant fine-grained visual features, enhancing the synergy between different visual tasks. Demonstrated over 12 visual tasks and evaluated across 22 datasets, VITRON showcases its extensive capabilities in the four main vision task clusters. Overall, this work illuminates the great potential of developing a more unified multimodal generalist.

# 1 Introduction

Recently, the field of multimodal large language models (MLLMs) has witnessed rapid and flourishing development across multiple communities. Extensive research efforts have been directed towards augmenting powerful, purely language-based LLMs with modules capable of visual perception, thereby extending their applicability to MLLMs [1, 49, 63, 127, 77, 111, 27]. MLLMs, such as BLIP-2 [49], LLaVA [63], MiniGPT-4 [138] and GPT-4V [121] etc., demonstrate a robust and exceptional capability in image understanding, paralleling the deep semantic comprehension of language. In the realm of vision, the ability to process and comprehend dynamic videos is equally critical. Concurrently, several MLLMs have emerged with a focus on video understanding, e.g., VideoChat [50] and Video-LLaMA [128], demonstrating significant advancements in video comprehension.

Subsequent studies have sought to further expand the capabilities of MLLMs, with efforts bifurcating into two primary dimensions. On one hand, there's a deepening of MLLMs' understanding of vision, transitioning from coarse, instance-level comprehension towards a pixel-level, fine-fined understanding of images, thereby achieving visual regional grounding capabilities, as seen in GLaMM [84], PixelLM [85], and MiniGPT-v2 [11], etc., alongside the counterparts in pixel-grounding video LLMs [74]. On the other hand, there's an expansion in the breadth of functionalities MLLMs can support within the vision field. A portion of the research has already ventured into enabling MLLMs not just to comprehend input vision signals but also to support the generation and output of vision content, with systems like GILL [43], Emu [96], etc., flexibly generating image content, and GPT4Video [105] and NExT-GPT [114] achieving video generation.

We posit that the future trend of vision LLMs necessarily involves the enhancement of their capabilities towards a high degree of unification, i.e., multimodal generalists. However, our observations reveal that despite the diversity of existing vision LLMs developed by the community, there is still a clear lack of unification. **First**, almost all existing vision LLMs treat images and videos as separate entities, either supporting only images or videos [1, 96, 138, 128]. We argue for a unified vision MLLM framework that concurrently supports both images and videos, acknowledging that vision inherently comprises both static images and dynamic videos - both core components of our world and largely interchangeable in most scenarios. **Second**, the current support for vision functionalities in MLLMs is found wanting, with most models only capable of understanding [63, 138], or at most generating images or videos [20, 105]. We contend that future MLLMs should embrace a broader spectrum of vision tasks and functionalities, enabling unified support for all vision-related tasks and achieving an "*one for all*" capability, which is vital for real-world applications, especially in vision creation that often involves a series of iterative and interactive operations. For example, users typically start by *generating* images from text, transforming an idea into visual content; and then refining this content through further fine-grained *editing* to add more details; following, proceeding to create dynamic content by *generating* videos from the images; and finally, engaging in several rounds of iterative interaction, such as video *editing*, to enhance and finalize their creation. **Last but not the least**, for a generalist integrated with various multimodal functionalities, one key lies in how to ensure that all tasks achieve their best performance as much as possible. This includes both that, 1) the instructions

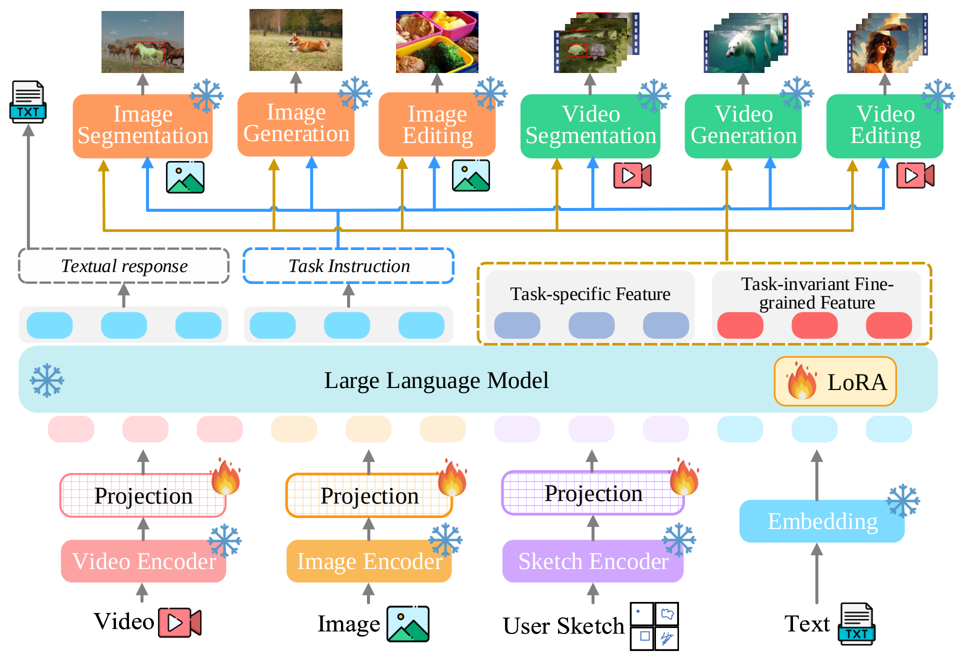



Figure 2: Technical overview of the VITRON framework.

from the LLM are precisely conveyed to the downstream decoders, and 2) different tasks do not undermine each other but rather cooperate.

To address all these gaps, this paper introduces **VITRON**, a pioneering universal pixel-level vision LLM, as shown in Fig. 2. First, VITRON leverages a backbone LLM for comprehending, reasoning, decision-making, and multi-round user interactions. To perceive both image and video modal signals and support fine-grained user visual inputs, VITRON incorporates encoders for images, videos, and regional box/sketch-specified inputs. On the backend, several state-of-the-art (SoTA) image and video modules are integrated for decoding and executing a wide range of vision tasks, spanning from lower to higher levels, such as visual understanding (perceiving and reasoning), generating, segmenting (grounding and tracking), editing (inpainting). To ensure that VITRON precisely conveys the LLM's decisions to various backend decoder modules for function invocation, we propose a novel hybrid method of instruction passing. Specifically, we enable the LLM to output not only discrete textual instructions, but also continuous signal feature embeddings passed to the modules. Finally, to maximize the functionalities of different modules within VITRON, we further devise a synergy module, where we fully maximize the task-persistent fine-grained visual features to be shared among different visual tasks.

The overall training for VITRON aims to equip it with robust and powerful vision understanding and manipulation capabilities. We first imbue VITRON basic MLLM skills by carrying out 1) vision-language alignment learning between the frontend encoders and central LLM, also 2) invocation-oriented instruction tuning, and 3) embedding-oriented alignment tuning between LLM and backend modules. Going beyond this, we further try to strengthen VITRON's capacities. On the one hand, we introduce fine-grained spatiotemporal vision grounding instruction tuning, training LLM on grounding predictions and pixel-aware perception for images and videos, such that VITRON sufficiently gains pixel-level visual perception. On the other hand, we utilize adversarial training [29, 100] to decouple *task-specific features* from *task-invariant fine-grained visual features* in signal feature representations, thereby enhancing the synergy between different tasks.

Extensive experiments covering 12 tasks across 22 datasets are performed. Leveraging its advanced architecture as a multimodal generalist, VITRON demonstrates proficiency in a comprehensive range of vision tasks. Notably, the unified system's performance is on par with or even surpasses singleton state-of-the-art specialists on specific tasks. Further analyses reveal the efficacy of each design of the system. Our overall contributions are summarized as follows.

❶ To our knowledge, we for the first time propose a grand unified vision MLLM, VITRON, capable of pixel-level understanding, generating, segmenting, editing of both images and videos. ❷ We

| Model | Vision Supporting | | Pixel/Regional Understanding | Segmenting/ Grounding | Generating | Editing | Cross-task Synergy |
|---|---|---|---|---|---|---|---|
| | Image | Video | | | | | |
| Flamingo [1] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| BLIP-2 [49] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| MiniGPT-4 [138] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| LLaVA [63] | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| GILL [43] | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| Emu [96] | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| MiniGPT-5 [135] | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ |
| DreamLLM [20] | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ |
| GPT4RoI [130] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| NExT-Chat [126] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| MiniGPT-v2 [11] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Shikra [12] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Kosmos-2 [78] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| GLaMM [84] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| Osprey [125] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| PixelLM [85] | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ |
| LLaVA-Plus [64] | ✓ | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| VideoChat [50] | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Video-LLaMA [128] | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Video-LLaVA [59] | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| Video-ChatGPT [67] | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ |
| GPT4Video [105] | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| PG-Video-LLaVA [74] | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ |
| NExT-GPT [114] | ✓ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| VITRON (Ours) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 1: Comparisons of existing (partially, imperfect coverage) representative vision MLLM.

introduce a more effective LLM-to-decode instruction-passing mechanism over both discrete texts and continuous signal embeddings. ❸ We propose carrying out various pixel-level vision-language spatiotemporal alignment learning for MLLMs to reach the best fine-grained visual capability. ❹ We devise a synergy module to maximize the task-persistent fine-grained visual features shareable among all different visual tasks, via which VITRON surpasses existing SoTA specialists' performance.

## 2 Related Work

Achieving a profound understanding and comprehensive operational capabilities in vision, ranging from low-level visual pixel understanding [7, 65, 120, 46, 55, 122, 102, 52–54] to high-level comprehension of overall semantics [19, 45, 70, 23, 31, 38, 39, 48, 115, 26, 56, 24], represents a significant topic. Recent years have seen the development of highly potent large-scale vision models, such as ViT [21] and CLIP [83], which have achieved remarkable vision understanding capabilities; models like SAM [42] and SEEM [139] have solved vision segmentation tasks; and diffusion-based models [34, 82, 73, 28, 112, 86, 25] have reached unprecedented performance in vision generation. Yet these models might lack an LLM as a central decision processor, unable to flexibly interpret user intent or execute tasks interactively [97, 47, 114]. The emergence of LLMs has exhibited unprecedented intelligence capability [76, 16, 99]. Extending the success of language understanding in LLMs, researchers have promptly investigated and developed various MLLMs, enabling LLMs to comprehend vision. By integrating high-performance vision encoders of images or videos into language-based LLMs, these models have been made capable of understanding vision signals [77, 1, 49, 81, 63]. Going beyond vision understanding, further research has aimed to enhance MLLMs, for instance, by endowing them with vision generation capabilities [43, 96] or supporting pixel-level understanding and grounding [130, 125, 85, 132, 110]. In Table 1 we summarize some existing popular vision MLLMs in terms of the vision function support.

However, we observe that current research on vision LLMs lacks depth in two critical aspects. Firstly, current vision LLMs tend to separate images and videos, supporting either one or the other. The construction of a unified MLLM is crucial, as vision inherently encompasses both static images and dynamic videos, both of which are core components of our visual world. Thus, covering both aspects simultaneously is essential for optimally adapting to practical applications. Although models like NExT-GPT [114] have relatively well-supported unification across various modalities, they fall short in supporting pixel-level in-depth vision understanding and comprehensive support for vision operation tasks. The second issue is the incomplete support for vision tasks by existing MLLMs. Most current MLLMs primarily support understanding images or videos [63, 138], with only a few supporting generation [20, 105] or editing/inpainting [113]. Building a generalist that can handle (almost) all vision-related tasks and operations in an end-to-end architecture should be the next major trend for vision MLLMs. Yet simply integrating existing visual specialists into an LLM to form MLLMs is not sufficient enough, as genuine human-level AI should possess universal intelligence with robust cross-task generalizability [72]. Thus, it is necessary to further consider how to enable synergy effects [20] among different task specialists within a generalist, for which goal, we have devised a synergy strategy in this work. Besides, compared to the multimodal comprehension capabilities of MLLM, endowing MLLM with strong multimodal generative abilities is even more challenging. The key lies in how to effectively and unbiasedly convey MLLM's semantic understanding signals to the backbone decoder modules. There are two mainstream approaches to LLM-to-decoder message passing within the MLLM community. One is based on discrete textual instructions [106, 90, 104], and the other on continuous signal embeddings [43, 20, 114]. However, we find that these two methods are complementary. Specifically, the former allows the LLM to efficiently convey task execution commands to the backend modules through simple text, but it struggles to provide modality-specific signals; the latter can conveniently carry the features needed for tasks, but fails to accurately convey execution intention (especially for managing many modules). In this work, we propose a hybrid method by integrating them together.

# 3 Architecture of VITRON

VITRON takes most common '*encoder-LLM-decoder*' architecture paradigm, as in existing popular MLLMs [63, 20, 114]. The overall framework is shown in Fig. 2, where three key blocks are included: 1) frontend vision&language encoders, 2) central LLM for semantics understanding and text generation, and 3) backend decoder modules for user responding and vision manipulation.

## 3.1 Frontend Vision-Language Encoding

For both images and videos, we employ the CLIP ViT-L/14@336px [83] as the encoder, respectively. The video encoder independently processes each frame, further employing average pooling across the temporal dimension to yield overall temporal representation features. Then, we employ a regional pixel-aware visual extractor as the sketch encoder for user interaction, e.g., clicking, drawing boxes or polygons, and making scribbles. We mainly follow [125], and use the object-based representations of mask regions that come from user's inputs, which not only encode the pixel-level visual features but also gather the spatial position information of each region. The region features are pooled with also the binary mask of spatial geometry of the object region encoded, and the resulting embeddings are used. Then, the multimodal feature representations are passed to LLM via linear projection.

## 3.2 Core LLM

In VITRON, an LLM serves as the pivotal agent. Following the most common practice [15, 94, 128], we utilize Vicuna (7B, version 1.5). The LLM processes inputs from both language and visual modalities to perform semantic understanding and reasoning, and then make decisions. For visual comprehension tasks, LLM directly outputs textual responses for users. On the other side, LLM also needs to transmit signals and instructions to backend modules, directing them to invocate more complex tasks that go beyond text generation, such as visual segmentation, generation, and editing. As emphasized earlier, the ability of LLMs to effectively and precisely convey messages is crucial to the performance of complex multimodal tasks. To this end, we propose fully integrating the advantages of the two common message-passing methods: *discrete textual instructions* and *continuous signal embeddings*. The former aids in accurately invoking different backbone modules (thanks to the LLM's proficiency in task dispatching), while the latter supplements with richer modality-preserved visual features that cannot be directly described through discrete text. As depicted in Fig. 2, the LLM outputs 1) text responses for users, 2) text instructions for module invocation, and 3) feature embeddings of special tokens. The feature embeddings are split into the task-specific features and the task-invariant fine-grained visual-language features. Both the text instructions and feature embeddings are passed to backbone modules.

## 3.3 Backend Visual Specialists

To enable our MLLM with various visual task abilities, we integrate an array of singleton vision specialists into LLM. For image generation and editing, we integrate the diffusion-based model GLIGEN [57]. For image and video segmentation, we opt for SEEM [139]. For video generation, ZeroScope [8] and I2VGen-XL [131] are utilized for text-to-video and image-to-video tasks, respectively. Lastly, for video editing functionality, we incorporate StableVideo [9]. The text instructions from LLM first determine which task module to invoke; simultaneously, feature embeddings are fed into the corresponding module's feature encoder to assist with task execution. Specifically, we design a structured invocation template, including 1) Module name, 2) Invocation command, and 3) Region (optional) specifying a fine-grained vision feature needed for certain tasks. The feature embeddings include both *task-specific features* and *task-invariant fine-grained features*. The purpose of this design is to achieve feature decoupling, during which we aim to have the task-invariant fine-grained features shared as widely as possible among all tasks to facilitate synergy between different tasks.

# 4 Pixel-aware Synergistic Vision-Language Understanding Tuning

With the VITRON framework, we now train the model with three stages of targets. First, we try to endow it with basic multimodal capabilities, i.e., comprehension and generation. Then, we engage in fine-grained vision grounding instruction tuning to further enhance the model's pixel-level perception abilities. Finally, we carry out cross-task synergy learning, maximizing the shared fine-grained features among all tasks.

## 4.1 Basic Multimodal Comprehension and Generation Skill Training

In the first stage of training, the primary goal is to equip the MLLM with basic multimodal understanding and generation abilities, including the frontend alignment of encoder-LLM, as well as the backend alignment of LLM-decoder. Appendix §B.1 details all the following three types of training.

**Overall Vision-Language Alignment Learning.** This is to ensure the input vision and language are mapped to a unified feature space. Following prior common practice, we utilize datasets comprising 'image-caption' pairs (CC3M [89]), 'video-caption' pairs (Webvid [4]), and 'region-caption' pairs (RefCOCO [40]) drawn from existing established corpora and benchmarks. When provided with an image, video, or specific visual region, we engage the frozen LLM to generate a text description or caption that aligns with the reference caption.

**Text Invocation Instruction Tuning.** This step of training aims to equip the system with the precise capability to execute commands, allowing the LLM to generate appropriate and correct invocation text instructions. To accomplish this, we collect a total of 55,000+ instruction tuning samples.

**Embedding-oriented Decoder Alignment Tuning.** Besides using explicit textual instruction to invocate downstream modules, the signal feature embedding/representation (from LLM) should also be fed to the modules. Following [114], we align the feature embedding with all the visual modules' input encoders via the decoding-side projection layers, i.e., by minimizing their distances.

## 4.2 Fine-grained Spatiotemporal Vision Grounding Instruction Tuning

A visual generalist should require a strong capability of pixel-aware vision understanding of both images and videos. Thus, we propose a fine-grained spatiotemporal vision grounding instruction tuning for VITRON. The core idea is to enable the LLM to ground the fine-grained spatiality of images and the detailed temporality of videos. Appendix §B.2 extends more detailed descriptions of the following three learning aspects.

**Image Spatial Grounding.** Considering that the LLM alone can only output text, we design it to respond with the corresponding bounding box areas. We focus on two types of tasks: grounded image captioning [133, 137] and referring image segmentation [40].

**Video Spatial-Temporal Grounding.** For videos, the LLM must identify spatial regions and ground them within the temporal context of the video, essentially achieving video tracking. Similarly, we explore tasks such as grounded video captioning [136] and referring video tracking [107].

**Grounding-aware Vision QA.** The grounding tasks mentioned above only touch upon the low-level aspects of vision perception. However, in many scenarios, it's essential for the LLM to possess high-level, in-depth vision reasoning capabilities, building upon the foundational low-level pixel grounding. Thus, we further introduce grounding-aware vision QA, including Image-QA [88, 37] and Video-QA [124], enabling LLM to undertake semantic-level QA tasks based on the grounded results.

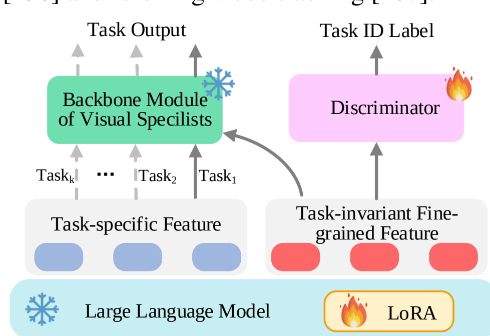



Figure 3: Illustration of the synergy module.

## 4.3 Cross-task Synergy Learning

As a generalist, directly invoking different specialists leads to a critical issue: *how to ensure that the different modules (tasks) work together synergistically?* Otherwise, without such collaboration, integrating them into a single compound system would be meaningless. To achieve this, here we propose decomposing the signal feature embeddings into task-specific features and task-invariant fine-grained features. Intuitively, since all the visual tasks we focus on are fine-grained, the more extensively the task-invariant fine-grained features are shared among different tasks, the more these tasks can benefit from each other, thus gaining greater synergy. Thereafter, we introduce a cross-task synergy learning module, as shown in Fig. 3. We employ adversarial training [3] to decouple task-specific from task-invariant features. We first let different backbone visual specialists make task predictions based on these two features (via concatenation). Meanwhile, we encourage a third-party discriminator (acts as a classifier) to determine which is the current task based solely on the shared feature representation. Ideally, once the discriminator can no longer accurately identify the task, the shared feature can be considered the most purified and broadly applicable across tasks.

# 5 Experiments

Now we try to quantify the performance of VITRON on the four vision task groups, covering 12 tasks across 22 datasets. All the training of VITRON is conducted on 10× A100 (80G) GPUs. To ensure a fair comparison, all subsequent experiments adopt settings same/similar to those of baseline systems, with evaluations following established practices. See more implementation details in Appendix §C. Due to space limits, more experimental results are presented in Appendix §D.

## 5.1 Results on Vision Segmentation

| Method | RefCOCO [40] | | | RefCOCO+ [123] | | | RefCOCOg [68] | |
|---|---|---|---|---|---|---|---|---|
| | Val | TestA | TestB | Val | TestA | TestB | Val | Test |
| LAVT [120] | 72.7 | 75.8 | 68.8 | 62.1 | 68.4 | 55.1 | 61.2 | 62.1 |
| GRES [61] | 73.8 | 76.5 | 70.2 | 66.0 | 71.0 | 57.7 | 65.0 | 66.0 |
| LISA [46] | 74.1 | 76.5 | 71.1 | 62.4 | 67.4 | 56.5 | 66.4 | 68.5 |
| NExT-Chat [126] | 74.7 | 78.9 | 69.5 | 65.1 | 71.9 | 56.7 | 67.0 | 67.0 |
| VITRON | **75.5** | **79.5** | **72.2** | **66.7** | **72.5** | **58.0** | **67.9** | **68.9** |
| w/o syng. | -2.4 | -2.0 | -1.9 | -1.7 | -2.1 | -1.5 | -1.8 | -1.6 |

Table 2: Results (cIoU) of referring image segmentation. 'w/o syng.': without synergy learning.

**Image Segmentation.** Table 2 presents the results of referring image segmentation on three datasets: RefCOCO [40], RefCOCO+ [123] and RefCOCOg [68]. We compare with several significant models, including state-of-the-art non-MLLM approaches and the MLLM baseline, NExT-Chat. It is evident that our VITRON, while slightly underperforming compared to NExT-Chat on the RefCOCO Val&TestA datasets, achieves superior performance on the remaining sets.

| Method | VidSTG [134] | HC-STVG [98] |
|---|---|---|
| G-DINO [65] | 25.3 | 19.5 |
| Video-LLaMA [128] | 28.6 | 26.1 |
| Video-ChatGPT [67] | 32.8 | 20.8 |
| PG-Video-LLaVA [74] | 34.2 | 28.3 |
| VITRON | **39.5** | **31.4** |
| w/o syng. | -4.3 | -3.7 |

Table 3: Results (mIoU) of video spatial grounding on two datasets.

| Method | $\mathcal{J}\&\mathcal{F}$ | $\mathcal{J}$ | $\mathcal{F}$ |
|---|---|---|---|
| RDE [51] | 77.4 | 73.6 | 81.2 |
| XMem [14] | 81.0 | 77.4 | 84.5 |
| DeAOT [122] | 80.7 | 76.9 | 84.5 |
| ISVOS [102] | 82.8 | 79.3 | 86.2 |
| VITRON | **84.2** | **81.5** | **86.7** |
| w/o syng. | -2.1 | -1.3 | -1.0 |

Table 4: Results of video object segmentation on DAVIS 17 [80] Test-Dev set.

**Video Segmentation.** For video segmentation, we explore two tasks: video spatial grounding (with bounding box) and video object segmentation (aka., video tracking; with mask). Table 3 showcases the comparisons between VITRON and current state-of-the-art (SoTA) video MLLMs in video spatial grounding. It is clear that VITRON significantly outperforms PG-Video-LLaVA. Table 4 presents a comparison of VITRON with some SoTA systems in video tracking, where our system continues to demonstrate superior performance.

## 5.2 Results on Fine-grained Vision Understanding

Next, we evaluate VITRON's capability in achieving fine-grained vision understanding, focusing mainly on region-level tasks for both images and videos.

**Region-level Image Understanding.** We test VITRON on tasks including image referring expression comprehension and image regional captioning. The comparisons and results

| Method | METEOR | CIREr |
|---|---|---|
| GRIT [109] | 15.2 | 71.6 |
| Kosmos-2 [78] | 14.1 | 62.3 |
| NExT-Chat [126] | 12.0 | 79.6 |
| MiniGPT-v2 [11] | 15.0 | 86.4 |
| GLaMM [84] | 16.2 | 106.0 |
| Osprey [125] | 16.6 | 108.3 |
| VITRON | **18.0** | **111.6** |
| w/o syng. | -3.0 | -8.6 |

Table 5: Performance of image regional captioning on RefCOCOg [68].

shown in Tables 5 illustrate that VITRON surpasses the best baseline across various datasets and metrics, proving its strong and accurate fine-grained semantic understanding of images.

The above two tasks focus solely on the model's ability to recognize at the region level. Taking a step further, we delve deeper into assessing the capability for image semantics understanding, particularly through image-based Visual Question Answering (VQA) tasks. These tasks effectively reflect the model's proficiency in comprehending the deeper semantic content of images. Table 6 displays the results across a series of six datasets for image-based VQA. We primarily compare two groups of models: those with and without pixel-wise vision grounding capabilities. The findings indicate that models equipped with fine-grained grounding abilities indeed show stronger task performance, suggesting that fine-grained grounding contributes to a more profound understanding of semantics. Notably, our VITRON achieves the highest performance among the models evaluated.

| Method | Ground? | OKVQA [88] | GQA [37] |
|---|---|---|---|
| Flamingo [1] | ✗ | 44.7 | - |
| BLIP-2 [49] | ✗ | 45.9 | 41.0 |
| InstructBLIP [17] | ✗ | - | 49.5 |
| MiniGPT-4 [138] | ✗ | 37.5 | 30.8 |
| LLaVA [63] | ✗ | 54.4 | 41.3 |
| Shikra [12] | ✓ | 47.2 | - |
| MiniGPT-v2 [11] | ✓ | 57.8 | 60.1 |
| VITRON | ✓ | **59.4** | **62.1** |
| w/o syng. | ✓ | -2.0 | -1.7 |

Table 6: Results (accuracy) on image-based VQA.

| Method | Ground? | ActivityNet-QA [124] | |
|---|---|---|---|
| | | Accuracy | Score |
| VideoChat [50] | ✗ | - | 2.2 |
| LLaMA-Adapter [30] | ✗ | 34.2 | 2.7 |
| Video-LLaMA [128] | ✗ | 12.4 | 1.1 |
| Video-ChatGPT [67] | ✗ | 35.2 | 2.7 |
| Video-LLaVA [59] | ✗ | 45.3 | 3.3 |
| PG-Video-LLaVA [74] | ✓ | 39.9 | 3.3 |
| VITRON | ✓ | **51.0** | **3.7** |
| w/o syng. | ✓ | -4.4 | -0.6 |

Table 7: Results (accuracy and confidence Score) on video QA.

**Region-level Video Understanding.** Similarly, for videos, we evaluate the Region-level Video Understanding capability. Building on observations from images, we now directly engage in video QA tasks. Table 7 presents the results on video QA across four representative datasets. Interestingly, while PG-Video-LLaVA has video grounding capabilities, it does not show better results than Video-LLaVA, which lacks grounding. However, our VITRON achieves superior performance. This indirectly proves that our system possesses more accurate video grounding capabilities (as previously demonstrated in Table 8), aiding in better video semantics understanding.

### 5.3 Results on Vision Generation

| Method | FID (↓) |
|---|---|
| GLIDE [75] | 12.24 |
| SD [86] | 11.21 |
| NExT-GPT [114] | 11.28 |
| Emu [96] | 11.66 |
| GILL [43] | 12.20 |
| DreamLLM [20] | 8.46 |
| VITRON | **7.57** |
| w/o syng. | +4.4 |

Table 8: Text-to-Image generation on COCO-caption data [60].

| Method | FID (↓) | CLIPSIM (↑) |
|---|---|---|
| CogVideo [33] | 23.59 | 0.2631 |
| MakeVideo [92] | 13.17 | 0.3049 |
| Latent-VDM [86] | 14.25 | 0.2756 |
| Latent-Shift [2] | 15.23 | 0.2773 |
| CoDi [97] | — | 0.2890 |
| NExT-GPT [114] | 13.04 | 0.3085 |
| VITRON | **10.11** | **0.3682** |
| w/o syng. | +3.17 | -0.5672 |

Table 9: Text-to-Video generation on MSR-VTT [118].

| Method | FVD (↓) | IS (↑) |
|---|---|---|
| AnimateAny [18] | 642.64 | 63.87 |
| DynamiCrafter [116] | 404.50 | 41.97 |
| SEINE [13] | 306.49 | 54.02 |
| VideoCrafter1 [10] | 297.62 | 50.88 |
| VITRON | **175.46** | **56.89** |
| w/o syng. | +96.24 | -5.03 |

Table 10: Image-to-Video generation on UCF101 [93].

Next, we assess our system's capabilities in vision generation, focusing on three of the most representative types of generation tasks: text-to-image generation, text-to-video generation, and image-to-video generation. These tasks broadly cover the spectrum of image generation requirements. Tables 8, 9, and 10 showcase how our VITRON performs in comparison to other SoTA systems, including both MLLM and non-MLLM synthesizers. The results clearly demonstrate that VITRON outperforms on all three tasks. For instance, in both text-to-image and text-to-video generation tasks, VITRON shows more advanced performance compared to NExT-GPT. Similarly, in the image-to-video generation task, VITRON still outshines the SoTA baseline, VideoCrafter1, showcasing superior results.

### 5.4 Results on Vision Editing

**Image Editing.** We use the MagicBrush dataset [129], which challenges models with an editing query that demands a series of complex edits to an image. These edits include removing, changing, inpainting, and adding elements. Since there are currently no MLLM systems that support image editing, our comparison is limited to non-LLM expert systems. In Table 11, we present the performance of different models across various metrics. VITRON demonstrates stronger performance on all metrics, indicating its stable image editing capabilities.

| Method | CLIP$_{dir}$ (↑) | CLIP$_{img}$ (↑) | CLIP$_{out}$ (↑) | L1 (↓) |
|---|---|---|---|---|
| InstructPix2Pix [6] | 0.115 | 0.837 | 0.245 | 0.093 |
| MagicBrush [129] | 0.123 | 0.883 | 0.261 | 0.058 |
| PnP [101] | 0.025 | 0.568 | 0.101 | 0.280 |
| NT-Inv [71] | 0.121 | 0.752 | 0.263 | 0.077 |
| Emu-Edit [91] | 0.135 | 0.897 | 0.261 | 0.052 |
| VITRON | **0.142** | **0.910** | **0.274** | **0.047** |
| w/o syng. | -0.012 | -0.104 | -0.078 | + 0.036 |

Table 11: Image editing results on MagicBrush [129].

| Method | Target-Editing | NonTarget-Unediting |
|---|---|---|
| Text2LIVE [5] | 4.5 | 1.3 |
| Tune-A-Video [108] | 7.8 | 4.6 |
| VITRON | **8.9** | **8.2** |
| w/o syng. | -2.2 | -1.6 |

Table 12: Human evaluation on video editing.

**Video Editing.** For video editing, the community currently lacks a standardized benchmark and evaluation method akin to those for image editing. Therefore, we opted for a manual evaluation approach. We asked different video editing systems to edit the same video based on the same query, after which five individuals were asked to score the edited videos. The evaluation focused on 1) the success of target content modifications and 2) the faithfulness/fidelity of non-target content. Table 12 presents the manual evaluation results for video editing. It is clear that VITRON outperforms the two baseline systems in both respects, showcasing superior video editing capabilities. Following this, we visualized the process of video editing by VITRON.

## 6 Discussions

Above we demonstrate the overall efficacy of VITRON via extensive quantitative comparison. Now we take one step further, exploring how and why the system advances via in-depth analyses.

► **Discrete Textual Instruction or Continuous Signal Embedding, Which Better?** Firstly, we explore different message-passing mechanisms to determine whether discrete textual instruction is more beneficial, or whether continuous signal embedding is better for building a multi-modal generalist. Also, we validate the pros and cons of the proposed hybrid method of message passing. We conduct tests on 6 tasks, where we compare the task performance of VITRON using the hybrid method (default setting), without signal embedding and without text instruction, as well as the successful execution rate of the backend task module. Fig. 4 presents the results. As can be observed, overall, the performance under scenarios utilizing both methods is consistently better, which confirms the effectiveness of our hybrid mode. Meanwhile, we find that the method of text instruction is more conducive to the successful execution of backend modules, but soft feature embedding seems to be more useful in terms of specific task performances.

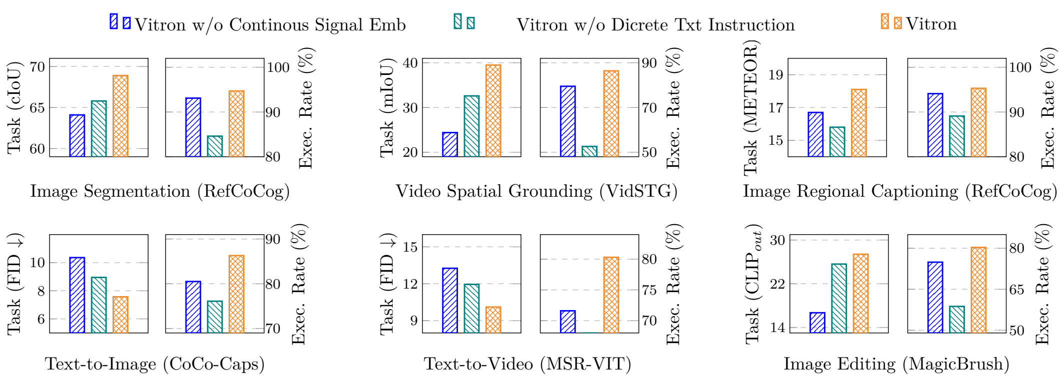



Figure 4: The influences of using different strategies for message passing.

► **How Much Does Each Fine-grained Visual Grounding Learning Contribute?** Next, we validate the specific contribution of the various fine-grained visual grounding learning strategies proposed in §4.2. Fig. 5 (the top 4 relate to image tasks, and the bottom 4 to video tasks) shows the impact on performance when a particular learning strategy is removed. Generally, all these 3 types of fine-grained visual grounding learning strategies are vital for different downstream tasks. For instance, grounding and referring segmentation tasks directly influence fine-grained visual recognition tasks, whereas tuning for grounding-aware visual QA considerably boosts cognition level QA tasks. This verifies the efficacy of our proposed fine-grained visual grounding tuning strategies.

►**Does VITRON Really Achieve Cross-task Synergy?** Finally, we investigate if our system could adequately support cross-task synergy. Based on the results of the ablation item for the 'synergy

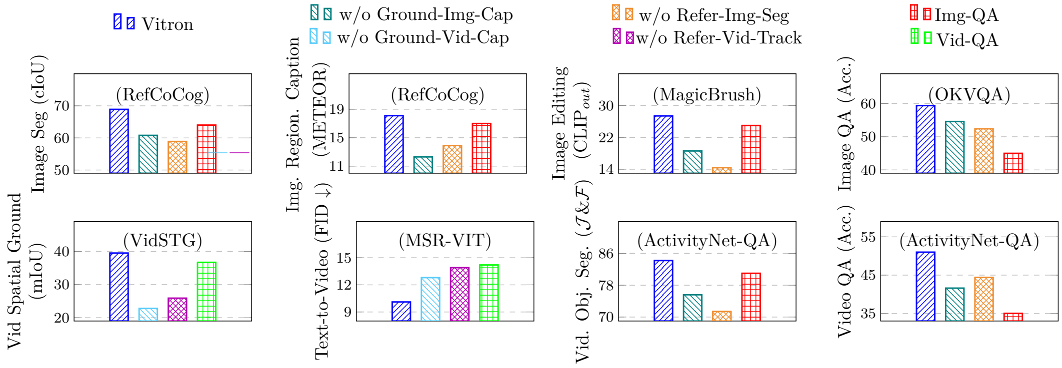



Figure 5: The impact of various fine-grained visual grounding learning strategies.

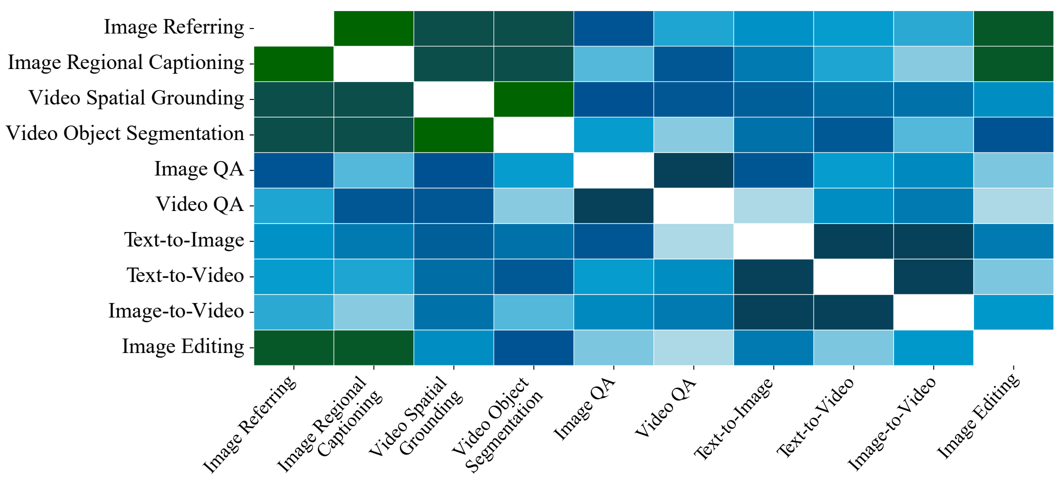



Figure 6: The synergy correlation between each pair of visual tasks. The deeper the color of the cell, the more synergistic they are in between.

module' in Table 2 to Table 12, we can observe that the synergy learning mechanism indeed positively influences overall performance. In Fig. 6 we further study whether there is synergy between different tasks and their collaborative relations. For ease of study, we considered a one-to-one mapping relationship, studying the cooperation between pairs of tasks one at a time. It is evident that the cooperative effects vary between different tasks. Tasks or backbone modules that rely more heavily on fine-grained visual features gained more significant improvements. This also demonstrates that our synergy learning module can successfully facilitate cross-task synergy.

# 7 Conclusion

In this work, we present VITRON, a grand unified pixel-level vision LLM for seamlessly understanding (perceiving and reasoning), generating, segmenting (grounding and tracking), and editing (inpainting) both images and videos. We further introduce a novel hybrid method of message passing that combines discrete textual instructions with continuous signal embeddings to ensure precise function invocation. Furthermore, VITRON employs pixel-level spatiotemporal vision-language alignment to enhance its fine-grained visual capabilities. A cross-task synergy module is also developed to optimize the use of task-invariant fine-grained visual features, boosting synergy across various visual tasks. On 12 visual tasks across 22 datasets, VITRON exhibits extensive capabilities in visual segmentation, fine-grained vision understanding, generation, and editing. Overall, this research showcases the great potential to build a vision-language generalist that can advance toward a more unified AI.


# Acknowledgements

This research is supported by Skywork AI, NExT++ Research Center, and CCF-Kuaishou Large Model Explorer Fund, Project of Future High-tech Video Intelligent Technology Innovation Center.

## References


[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Proceedings of the NeurIPS*, 2022.

[2] Jie An, Songyang Zhang, Harry Yang, Sonal Gupta, Jia-Bin Huang, Jiebo Luo, and Xi Yin. Latent-shift: Latent diffusion with temporal shift for efficient text-to-video generation. *CoRR*, abs/2304.08477, 2023.

[3] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. In *Proceedings of the IJCAI*, pages 4312–4321, 2021.

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proceedings of the ICCV*, pages 1708–1718, 2021.

[5] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Proceedings of the ECCV*, pages 707–723. Springer, 2022.

[6] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the CVPR*, pages 18392–18402, 2023.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the ECCV*, 2020.

[8] Cerspense. Zeroscope: Diffusion-based text-to-video synthesis. 2023. URL `https://huggingface.co/cerspense`.

[9] Wenhao Chai, Xun Guo, Gaoang Wang, and Yan Lu. Stablevideo: Text-driven consistency-aware diffusion video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23040–23050, 2023.

[10] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023.

[11] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[12] 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.

[13] Xinyuan Chen, Yaohui Wang, Lingjun Zhang, Shaobin Zhuang, Xin Ma, Jiashuo Yu, Yali Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Seine: Short-to-long video diffusion model for generative transition and prediction. In *Proceedings of the ICLR*, 2023.

[14] Ho Kei Cheng and Alexander G Schwing. Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In *Proceedings of the ECCV*, pages 640–658. Springer, 2022.

[15] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90 2023.

[16] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022.

[17] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *CoRR*, abs/2305.06500, 2023.

[18] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Fine-grained open domain image animation with motion guidance. *arXiv preprint arXiv:2311.12886*, 2023.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the CVPR*, pages 248–255. Ieee, 2009.

[20] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023.

[21] 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 *Proceedings of the ICLR*, 2021.

[22] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the CVPR*, pages 5374–5383, 2019.

[23] Hao Fei, Qian Liu, Meishan Zhang, Min Zhang, and Tat-Seng Chua. Scene graph as pivoting: Inference-time image-free unsupervised multimodal machine translation with visual scene hallucination. In *Proceedings of the ACL*, pages 5980–5994, 2023.

[24] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, and Tat-Seng Chua. Dysen-vdm: Empowering dynamics-aware text-to-video diffusion with llms. In *Proceedings of the CVPR*, pages 7641–7653, 2024.

[25] Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Proceedings of the ICML*, 2024.

[26] Hao Fei, Shengqiong Wu, Meishan Zhang, Min Zhang, Tat-Seng Chua, and Shuicheng Yan. Enhancing video-language representations with structural spatio-temporal alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[27] Hao Fei, Yuan Yao, Zhuosheng Zhang, Fuxiao Liu, Ao Zhang, and Tat-Seng Chua. From multimodal llm to human-level ai: Modality, instruction, reasoning, efficiency and beyond. In *Proceedings of the COLING: Tutorial Summaries*, pages 1–8, 2024.

[28] Weixi Feng, Xuehai He, Tsu-Jui Fu, Varun Jampani, Arjun R. Akula, Pradyumna Narayana, Sugato Basu, Xin Eric Wang, and William Yang Wang. Training-free structured diffusion guidance for compositional text-to-image synthesis. *CoRR*, abs/2212.05032, 2022.

[29] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[30] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

[31] Xin Gu, Guang Chen, Yufei Wang, Libo Zhang, Tiejian Luo, and Longyin Wen. Text with knowledge graph augmented transformer for video captioning. In *Proceedings of the CVPR*, pages 18941–18951, 2023.

[32] 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.

[33] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. Cogvideo: Large-scale pretraining for text-to-video generation via transformers. *CoRR*, abs/2205.15868, 2022.

[34] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Towards non-autoregressive language models. *CoRR*, 2021.

[35] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the ICLR*, 2022.

[36] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE transactions on pattern analysis and machine intelligence*, 43(5): 1562–1577, 2019.

[37] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the CVPR*, pages 6700–6709, 2019.

[38] Jiayi Ji, Yunpeng Luo, Xiaoshuai Sun, Fuhai Chen, Gen Luo, Yongjian Wu, Yue Gao, and Rongrong Ji. Improving image captioning by leveraging intra-and inter-layer global representation in transformer network. In *Proceedings of the AAAI*, pages 1655–1663, 2021.

[39] Jiayi Ji, Yiwei Ma, Xiaoshuai Sun, Yiyi Zhou, Yongjian Wu, and Rongrong Ji. Knowing what to learn: a metric-oriented focal mechanism for image captioning. *IEEE Transactions on Image Processing*, 31: 4321–4335, 2022.

[40] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the EMNLP*, pages 787–798, 2014.

[41] Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. 33: 2611–2624, 2020.

[42] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[43] Jing Yu Koh, Daniel Fried, and Russ R Salakhutdinov. Generating images with multimodal language models. In *Proceedings of the NeurIPS*, 2023.

[44] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017.

[45] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[46] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023.

[47] Juncheng Li, Kaihang Pan, Zhiqi Ge, Minghe Gao, Hanwang Zhang, Wei Ji, Wenqiao Zhang, Tat-Seng Chua, Siliang Tang, and Yueting Zhuang. Fine-tuning multimodal llms to follow zero-shot demonstrative instructions. In *Proceedings of the ICLR*, 2023.

[48] Juncheng Li, Siliang Tang, Linchao Zhu, Wenqiao Zhang, Yi Yang, Tat-Seng Chua, and Fei Wu. Variational cross-graph reasoning and adaptive structured semantics learning for compositional temporal grounding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[49] 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 *Proceedings of the ICML*, pages 19730–19742, 2023.

[50] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *CoRR*, abs/2305.06355, 2023.

[51] Mingxing Li, Li Hu, Zhiwei Xiong, Bang Zhang, Pan Pan, and Dong Liu. Recurrent dynamic embedding for video object segmentation. In *Proceedings of the CVPR*, pages 1332–1341, 2022.

[52] Xiangtai Li, Wenwei Zhang, Jiangmiao Pang, Kai Chen, Guangliang Cheng, Yunhai Tong, and Chen Change Loy. Video k-net: A simple, strong, and unified baseline for video segmentation. In *Proceedings of the CVPR*, 2022.

[53] Xiangtai Li, Haobo Yuan, Wenwei Zhang, Guangliang Cheng, Jiangmiao Pang, and Chen Change Loy. Tube-link: A flexible cross tube baseline for universal video segmentation. In *Proceedings of the ICCV*, 2023.

[54] Xiangtai Li, Henghui Ding, Wenwei Zhang, Haobo Yuan, Guangliang Cheng, Pang Jiangmiao, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[55] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *Proceedings of the CVPR*, 2024.

[56] Yicong Li, Xiang Wang, Junbin Xiao, Wei Ji, and Tat-Seng Chua. Invariant grounding for video question answering. In *Proceedings of the CVPR*, pages 2918–2927, 2022.

[57] Yuheng Li, Haotian Liu, Qingyang Wu, Fangzhou Mu, Jianwei Yang, Jianfeng Gao, Chunyuan Li, and Yong Jae Lee. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the CVPR*, pages 22511–22521, 2023.

[58] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *Proceedings of the CVPR*, pages 4641–4650, 2016.

[59] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[60] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Proceedings of the ECCV*, pages 740–755, 2014.

[61] Chang Liu, Henghui Ding, and Xudong Jiang. Gres: Generalized referring expression segmentation. In *Proceedings of the CVPR*, pages 23592–23601, 2023.

[62] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

[63] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *CoRR*, abs/2304.08485, 2023.

[64] Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, et al. Llava-plus: Learning to use tools for creating multimodal agents. *arXiv preprint arXiv:2311.05437*, 2023.

[65] 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.

[66] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

[67] Muhammad Maaz, Hanoona Abdul Rasheed, Salman H. Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *CoRR*, abs/2306.05424, 2023.

[68] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the CVPR*, pages 11–20, 2016.

[69] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831*, 2016.

[70] Victor Siemen Janusz Milewski, Marie-Francine Moens, and Iacer Calixto. Are scene graphs good enough to improve image captioning? In *Proceedings of the AACL*, pages 504–515, 2020.

[71] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the CVPR*, pages 6038–6047, 2023.

[72] Meredith Ringel Morris, Jascha Sohl-dickstein, Noah Fiedel, Tris Warkentin, Allan Dafoe, Aleksandra Faust, Clement Farabet, and Shane Legg. Levels of agi: Operationalizing progress on the path to agi. *arXiv preprint arXiv:2311.02462*, 2023.

[73] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[74] Shehan Munasinghe, Rusiru Thushara, Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, Mubarak Shah, and Fahad Khan. Pg-video-llava: Pixel grounding large video-language models. *arXiv preprint arXiv:2311.13435*, 2023.

[75] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the ICML*, pages 16784–16804, 2022.

[76] OpenAI. Introducing chatgpt. 2022.

[77] OpenAI. Gpt-4 technical report. 2022.

[78] 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. *arXiv preprint arXiv:2306.14824*, 2023.

[79] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the ICCV*, pages 2641–2649, 2015.

[80] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

[81] Long Qian, Juncheng Li, Yu Wu, Yaobo Ye, Hao Fei, Tat-Seng Chua, Yueting Zhuang, and Siliang Tang. Momentor: Advancing video large language model with fine-grained temporal reasoning. In *Proceedings of the ICML*, 2024.

[82] Leigang Qu, Shengqiong Wu, Hao Fei, Liqiang Nie, and Tat-Seng Chua. Layoutllm-t2i: Eliciting layout guidance from LLM for text-to-image generation. In *Proceedings of the ACM MM*, pages 643–654, 2023.

[83] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the ICML*, pages 8748–8763, 2021.

[84] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Erix Xing, Ming-Hsuan Yang, and Fahad S Khan. Glamm: Pixel grounding large multimodal model. *arXiv preprint arXiv:2311.03356*, 2023.

[85] Zhongwei Ren, Zhicheng Huang, Yunchao Wei, Yao Zhao, Dongmei Fu, Jiashi Feng, and Xiaojie Jin. Pixellm: Pixel reasoning with large multimodal model. *arXiv preprint arXiv:2312.02228*, 2023.

[86] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the CVPR*, pages 10674–10685, 2022.

[87] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021.

[88] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *Proceedings of the ECCV*, pages 146–162, 2022.

[89] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the ACL*, pages 2556–2565, 2018.

[90] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving AI tasks with chatgpt and its friends in huggingface. *CoRR*, abs/2303.17580, 2023.

[91] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. *arXiv preprint arXiv:2311.10089*, 2023.

[92] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *CoRR*, abs/2209.14792, 2022.

[93] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[94] Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. Pandagpt: One model to instruction-follow them all. *CoRR*, abs/2305.16355, 2023.

[95] Peize Sun, Jinkun Cao, Yi Jiang, Zehuan Yuan, Song Bai, Kris Kitani, and Ping Luo. Dancetrack: Multi-object tracking in uniform appearance and diverse motion. In *Proceedings of the CVPR*, pages 20993–21002, 2022.

[96] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[97] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *CoRR*, abs/2305.11846, 2023.

[98] Zongheng Tang, Yue Liao, Si Liu, Guanbin Li, Xiaojie Jin, Hongxu Jiang, Qian Yu, and Dong Xu. Human-centric spatio-temporal video grounding with visual transformers. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(12):8238–8249, 2021.

[99] 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.

[100] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[101] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the CVPR*, pages 1921–1930, 2023.

[102] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Chuanxin Tang, Xiyang Dai, Yucheng Zhao, Yujia Xie, Lu Yuan, and Yu-Gang Jiang. Look before you match: Instance understanding matters in video object segmentation. In *Proceedings of the CVPR*, pages 2268–2278, 2023.

[103] Peng Wang, An Yang, Rui Men, Junyang Lin, Shuai Bai, Zhikang Li, Jianxin Ma, Chang Zhou, Jingren Zhou, and Hongxia Yang. Ofa: Unifying architectures, tasks, and modalities through a simple sequence-to-sequence learning framework. In *Proceedings of the ICML*, pages 23318–23340. PMLR, 2022.

[104] Xinyu Wang, Bohan Zhuang, and Qi Wu. Modaverse: Efficiently transforming modalities with llms. *arXiv preprint arXiv:2401.06395*, 2024.

[105] Zhanyu Wang, Longyue Wang, Zhen Zhao, Minghao Wu, Chenyang Lyu, Huayang Li, Deng Cai, Luping Zhou, Shuming Shi, and Zhaopeng Tu. Gpt4video: A unified multimodal large language model for lnstruction-followed understanding and safety-aware generation. *arXiv preprint arXiv:2311.16511*, 2023.

[106] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *CoRR*, abs/2303.04671, 2023.

[107] Dongming Wu, Wencheng Han, Tiancai Wang, Xingping Dong, Xiangyu Zhang, and Jianbing Shen. Referring multi-object tracking. In *Proceedings of the CVPR*, pages 14633–14642, 2023.

[108] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, 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. *CoRR*, abs/2212.11565, 2022.

[109] Jialian Wu, Jianfeng Wang, Zhengyuan Yang, Zhe Gan, Zicheng Liu, Junsong Yuan, and Lijuan Wang. Grit: A generative region-to-text transformer for object understanding. *arXiv preprint arXiv:2212.00280*, 2022.

[110] Jianzong Wu, Xiangtai Li, Chenyang Si, Shangchen Zhou, Jingkang Yang, Jiangning Zhang, Yining Li, Kai Chen, Yunhai Tong, Ziwei Liu, et al. Towards language-driven video inpainting via multimodal large language models. *arXiv preprint arXiv:2401.10226*, 2024.

[111] Mingrui Wu, Xinyue Cai, Jiayi Ji, Jiale Li, Oucheng Huang, Gen Luo, Hao Fei, Xiaoshuai Sun, and Rongrong Ji. Controlmllm: Training-free visual prompt learning for multimodal large language models. *arXiv preprint arXiv:2407.21534*, 2024.

[112] Shengqiong Wu, Hao Fei, Hanwang Zhang, and Tat-Seng Chua. Imagine that! abstract-to-intricate text-to-image synthesis with scene graph hallucination diffusion. In *Proceedings of the NeurIPS*, pages 79240–79259, 2023.

[113] Shengqiong Wu, Hao Fei, Xiangtai Li, Jiayi Ji, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Towards semantic equivalence of tokenization in multimodal llm. *arXiv preprint arXiv:2406.05127*, 2024.

[114] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal llm. In *Proceedings of the ICML*, 2024.

[115] Junbin Xiao, Angela Yao, Zhiyuan Liu, Yicong Li, Wei Ji, and Tat-Seng Chua. Video as conditional graph hierarchy for multi-granular question answering. In *Proceedings of the AAAI*, pages 2804–2812, 2022.

[116] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023.

[117] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the ACM MM*, pages 1645–1653, 2017.

[118] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *Proceedings of the CVPR*, pages 5288–5296, 2016.

[119] Ning Xu, Linjie Yang, Yuchen Fan, Dingcheng Yue, Yuchen Liang, Jianchao Yang, and Thomas Huang. Youtube-vos: A large-scale video object segmentation benchmark. *arXiv preprint arXiv:1809.03327*, 2018.

[120] Zhao Yang, Jiaqi Wang, Yansong Tang, Kai Chen, Hengshuang Zhao, and Philip HS Torr. Lavt: Language-aware vision transformer for referring image segmentation. In *Proceedings of the CVPR*, pages 18155–18165, 2022.

[121] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1): 1, 2023.

[122] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. 35:36324–36336, 2022.

[123] Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *Proceedings of the ECCV*, pages 69–85, 2016.

[124] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019.

[125] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. *arXiv preprint arXiv:2312.10032*, 2023.

[126] Ao Zhang, Liming Zhao, Chen-Wei Xie, Yun Zheng, Wei Ji, and Tat-Seng Chua. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*, 2023.

[127] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. Vpgtrans: Transfer visual prompt generator across llms. 36, 2024.

[128] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *CoRR*, abs/2306.02858, 2023.

[129] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. 36, 2024.

[130] Shilong Zhang, Peize Sun, Shoufa Chen, Min Xiao, Wenqi Shao, Wenwei Zhang, Kai Chen, and Ping Luo. Gpt4roi: Instruction tuning large language model on region-of-interest. *arXiv preprint arXiv:2307.03601*, 2023.

[131] Shiwei Zhang, Jiayu Wang, Yingya Zhang, Kang Zhao, Hangjie Yuan, Zhiwu Qin, Xiang Wang, Deli Zhao, and Jingren Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models. *arXiv preprint arXiv:2311.04145*, 2023.

[132] Tao Zhang, Xiangtai Li, Hao Fei, Haobo Yuan, Shengqiong Wu, Shunping Ji, Chen Change Loy, and Shuicheng Yan. Omg-llava: Bridging image-level, object-level, pixel-level reasoning and understanding. *arXiv preprint arXiv:2406.19389*, 2024.

[133] Wenqiao Zhang, Haochen Shi, Siliang Tang, Jun Xiao, Qiang Yu, and Yueting Zhuang. Consensus graph representation learning for better grounded image captioning. In *Proceedings of the AAAI*, pages 3394–3402, 2021.

[134] Zhu Zhang, Zhou Zhao, Yang Zhao, Qi Wang, Huasheng Liu, and Lianli Gao. Where does it exist: Spatio-temporal video grounding for multi-form sentences. In *Proceedings of the CVPR*, pages 10668–10677, 2020.

[135] Kaizhi Zheng, Xuehai He, and Xin Eric Wang. Minigpt-5: Interleaved vision-and-language generation via generative vokens. *arXiv preprint arXiv:2310.02239*, 2023.

[136] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. Grounded video description. In *Proceedings of the CVPR*, pages 6578–6587, 2019.

[137] Yuanen Zhou, Meng Wang, Daqing Liu, Zhenzhen Hu, and Hanwang Zhang. More grounded image captioning by distilling image-text matching model. In *Proceedings of the CVPR*, pages 4776–4785, 2020.

[138] 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.

[139] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Wang, Lijuan Wang, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. In *Proceedings of the NeurIPS*, 2024.

# A Details of Backbone Visual Modules/Specialists

To address the inability of text-based LLMs in handling various vision tasks, we consider integrating off-the-shelf external modules. Once the LLM generates invocation details through understanding the input and recognizing the user's intent, the corresponding modules are activated to produce non-textual outputs. Technically, we employ a variety of current SoTA expert models for vision processing. For image generation and editing, we integrate the diffusion-based model GLIGEN [57]. For image and video segmentation, we opt for SEEM [139]. For video generation, ZeroScope [8] and I2VGen-XL [131] are utilized for text-to-video and image-to-video tasks, respectively. Lastly, for video editing functionality, we incorporate StableVideo [9]. In Table 13, we summarize the functionality of each backend module, along with a specification of the inputs and outputs.

| No. | Function | Model | Input | Output |
|---|---|---|---|---|
| 1 | Text Generation | - | - | - |
| 2 | Image Generation | GLIGEN [57] | Text | Image |
| 3 | Image Segmentation | SEEM [139] | Text, Image | Image, Mask ǀ BBox |
| 4 | Image Editing | GLIGEN [57] | Text, Image [BBox ǀ Mask] | Image |
| 5 | Video Generation | ZeroScope [8] | Text | Video |
| 6 | | I2VGen-XL [131] | Image | Video |
| 7 | Video Segmentation | SEEM [139] | Text, Video [BBox ǀ Mask] | Video, Mask ǀ BBox |
| 8 | Video Editing | StableVideo [9] | Text, Video | Video |

Table 13: Summary of backend modules in VITRON.

# B Extensions of Pixel-aware Synergistic Vision-Language Understanding Learning

This section extends more details of the §4 in the main article.

## B.1 Baisc MLLM Skill Training

**Overall Vision-Language Alignment Learning.** In line with the methodologies in current MLLMs, our approach involves This step aims at mapping the input vision language features to a unified feature space. This space creates representations that the central LLM can interpret, thereby enabling it to process incoming vision signals effectively. We utilize datasets of 'image-caption' pairs (CC3M [89]), 'video-caption' pairs (Webvid [4]), and 'region-caption' pairs (RefCOCO [40]) from existing corpora. When provided with an image, video, or specific visual region, we engage the frozen LLM to generate a text description or caption that aligns with the reference caption.

**Invocation-oriented Instruction Tuning.** The above phase of training endows both the LLM and the frontend encoders with the ability to understand vision. This step, instruction tuning for invocation, aims to equip the system with the precise capability to execute commands, allowing the LLM to generate appropriate and correct invocation text. This text is then used to trigger various backend task execution modules. Different terminal vision tasks might require distinct invocation commands. To unify this, we try to standardize the LLM's response output into a structured text format, which includes: 1) User response output, which directly replies to the user's input. 2) Module name, indicating which function or task is to be executed. 3) Invocation command, a meta-instruction for triggering the task module. 4) Region (optional), specifying a fine-grained vision feature needed for certain tasks, such as in video tracking or vision editing, where backend modules require this information. For the region, based on LLM's pixel-level understanding, a bounding box described by the coordinates ($X_l$, $Y_t$, $X_r$, $Y_b$) will be output. Following, we exemplify a video tracking example for the structured LLM response for module invocation.

► **Structured LLM response for module invocation (with video tracking example):**

**Input:**  

Can you help me track the clock I circled in the video?

**A: [LLM response]**

Sure! You provide a video where there is a female cop talking on the walkie-talkie, responding to emergency calls, and crime prevention. Also in the scene, the clock is green in color. Following I will outline the clock in the video.

**B: [Task Instruction]**
**<Module>** Video Segmentation **</Module>**
**<Instruction>** segmentation: clock **</Instruction>**
**<Region>** (0.23, 0.35, 0.11, 0.26) **</Region>**

| **Function** | **VITRON Input** | **Module Input** | **Data Source** | **Amount** |
|---|---|---|---|---|
| Image Generation | | Image-Caption | CC3M [89] | 4,000 |
| Image Segmentation | | | RefCOCO [40] | 4,000 |
| | | | RefCOCO [40] | 5,000 |
| | | , Object-Name | gRefCOCO [61] | 2,028 |
| Image Editing | | | COCO2017 [60] | 4,000 |
| | | , Bounding-Box | MagicBrush [129] | 5,000 |
| Video Generation | | Video-Caption | WebVid [4] | 7,000 |
| | | | LAION-400M [87] | 4,000 |
| Video Segmentation | | | WebVid [4], VG [44] | 5,000 |
| | | , Bounding-Box | WebVid [4] | 5,000 |
| Video Editing | | , Editing-Query | WebVid [4] | 5,000 |
| | | , Editing-Query | WebVid [4] | 5,000 |

Table 14: Feature summary of the constructed data for text invocation instruction tuning. in image segmentation means the reference image provided by users. in video segmentation means the intermediate referred video keyframe interpreted within the system.

To teach the LLM to produce the correct invocation responses, we need to create data for instruction tuning. A key is ensuring that the data covers all possible scenarios. We must take into account different ways users might interact with the system for each functionality mentioned in Table 13 (except for text generation). For example, when requesting video creation, a user might describe what they want purely in text, or provide a reference image as the basis for the desired video. Similarly, for editing images or videos, users could express their editing requests either through text, or by using sketches, scribbles and other interactions. Thus, the LLM needs to be versatile in accepting various types of user inputs and generating an accurate invocation response that matches the requirements of the backend modules. Technically, we make use of the existing annotated datasets for various vision tasks included in this work. For each task under specific different user input scenarios, with the corresponding data, we design various template dialogue-format examples. Based on these examples we then prompt the GPT-4 to generate more samples under various topics and enriched scenarios. Finally, we collect a total of 55,000+ invocation-oriented instruction tuning samples. In Table 14 we provide a summary of these datasets.

**Embedding-oriented Decoder Alignment Tuning.** Besides using the explicit textual instruction to invocate downstream modules, also the signal feature embedding/representation (from LLM) should also be fed to the modules. Denote the *task-specific features* as $\boldsymbol{v}^p$ and *task-invariant fine-grained features* as $\boldsymbol{v}^s$. We will concatenate them as one unified feature embedding $\boldsymbol{v} = [\boldsymbol{v}^p; \boldsymbol{v}^s]$ and then send $\boldsymbol{v}$ to the downstream module.

Following [114], we align the feature embedding with all the visual module's input encoders via the decoding-side projection layers. We do this feature alignment learning by minimizing the distance between the projected feature embedding and the module's input encoder. For example for diffusion-based image or video generation, we may directly use the textual condition encoder, while keeping all the other modules fixed. Technically, to endow the model to produce other modalities beyond text, we add the signal tokens to the vocabulary of the LLM. In the alignment training phase, we mainly take the captions from CC3M, WebVid, and AudioCaps as inputs and concatenate them with the special signal tokens as outputs. The loss function comprises three key components: 1) the negative log-likelihood of producing signal tokens, and 2) the caption alignment loss: $l_2$-distance between the hidden states of signal tokens produced by the LLM and the conditional text representations derived from the text encoder within diffusion models, and 3) conditional latent denoising loss [86].

### B.2 Fine-grained Spatiotemporal Vision Grounding Instruction Tuning

We propose a fine-grained spatiotemporal vision grounding instruction tuning for VITRON. The core idea is to enable the LLM to ground the fine-grained spatiality of images and the detailed temporality of videos. Technically, we leverage LoRA [35] to enable a small subset of parameters within the LLM to be updated during the tuning.

**Image Spatial Grounding.** Considering that the LLM alone can only output text, we design it to respond with the corresponding bounding box areas. We focus on two types of tasks: grounded image captioning and referring image segmentation. For grounded image captioning, we input an image and identify a specific object within it, prompting the LLM to describe the identified object. Conversely, for referring image segmentation (where we consider outputting a bounding box, akin to phrase grounding), the task involves inputting a complete image along with a related phrase or sentence description, and the LLM is expected to output the object's spatial bounding box, represented by coordinate numbers ($X_l$, $Y_t$, $X_r$, $Y_b$). The X and Y coordinates are normalized real values within the range [0, 1], where $<X_l>$ and $<Y_t>$ indicate the top-left corner, and $<X_r>$ and $<Y_b>$ represent the bottom-right corner of the bounding box. We consider datasets such as Flickr30K Entities [79], RefCOCO [40], RefCOCO+ [123] and RefCOCOg [68]. Utilizing GPT-4, we preprocess and expand the original data into a multi-turn QA format, thereby generating the most suitable instruction-tuning data.

► **Grounded Image Captioning:**
**Input:**
Please generate a detailed caption for the given image, and clearly link each part of the caption to specific objects or areas in the image which you can denote with a bounding box with "object: ($X_l$, $Y_t$, $X_r$, $Y_b$)" format.
**Output:**
*A girl wearing a pink dress is sitting on a long bench in the park reading a book.*
*girl:* ($X_l$, $Y_t$, $X_r$, $Y_b$),
*pink dress:* ($X_l$, $Y_t$, $X_r$, $Y_b$), ...

► **Referring Image Segmentation:**
**Input:**
Please identify the target object from the given images based on the following text query: "A man in a black shirt holding a water bottle".
Please output the bounding box ($X_l$, $Y_t$, $X_r$, $Y_b$) of the target object.
**Output:**
($X_l$, $Y_t$, $X_r$, $Y_b$)

**Video Spatial-Temporal Grounding.** For videos, the LLM must not only identify spatial regions but also ground these within the temporal context of the video, essentially achieving video tracking. Similarly, we explore tasks such as grounded video captioning and referring video tracking. While the approach is akin to that used for images, it requires not just the output of object bounding box coordinates in each frame but also the specification of the duration of the frames ($F_s$, $F_e$) in which these coordinates are to be outputted. Here $<F_s>$ denotes the starting frame number, and $<F_e>$ means the ending frame number. We mainly construct the required instruction tuning data from several key video tracking datasets: LaSOT [22], GOT10K [36], DanceTrack [95], and MOT16 [69].

► **Grounded Video Captioning:**
**Input:**
Please generate a caption for the given video, and link each part of the caption to specific objects in the video with its temporal presence duration. You should denote it with a bounding box with the starting and ending frame number in format as "object: ($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$)".
**Output:**
*A dog is running after a ball on the grass.*
*dog:* ($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$),
*ball:* ($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$), ...

► **Referring Video Tracking:**
**Input:**
Please track the specified object throughout the video based on the following given description, and mark its starting and ending position in each frame: "A man is skiing on a thick-snow-covered slope."
Please output in the format of "($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$)".
**Output:**
($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$)

**Grounding-aware Vision QA.** The grounding tasks mentioned above only touch upon the low-level aspects of vision perception. However, in many scenarios, it's essential for the LLM to possess high-level, in-depth vision reasoning capabilities, building upon the foundational low-level pixel grounding. Therefore, we further introduce grounding-aware vision QA, enabling the LLM to undertake semantic-level QA tasks based on the grounded results. Specifically, we consider both image-grounded QA and video-grounded QA. The data sources for these tasks remain the same as mentioned above for the grounding tasks. And we again rely on GPT-4[2] to use its imaginative capabilities to design reasonable questions and answers based on the given grounding content of the above grounding/segmentation datasets, formatting these into an instruction-tuning format.

► **Grounded Image QA:**

**Input:**

Based on the given image, please select the correct answer among all the candidates: Q: Where is the child sitting? A: 1) arms, 2) ground, 3) chair, 4) bed.

Please first identify and ground the target object (in coordinates) mentioned in the question, and then proceed to answer the question.

**Output:**

The target object mentioned in the question is "child," with the position given by *($X_l$, $Y_t$, $X_r$, $Y_b$)*. From this, it can be determined that the child is being held by a woman, which means the child is sitting in the woman's arms. Therefore, the answer is 1) arms.

► **Grounded Video QA:**

**Input:**

Based on the provided video, answer the following question by choosing the most appropriate answer from the options given. Q: If the rope in the video breaks, what would happen to the person in the video? A: 1) running, 2) jumping up, 3) flying away, 4) fracture.

First, analyze the spatial position and temporality of the target object mentioned in the question within the video, and based on this analysis, determine the answer to the question.

**Output:**

The objects involved in the question are the rope *($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$)* and the person *($X_l$, $Y_t$, $X_r$, $Y_b$ | $F_s$, $F_e$)*, with their relationship being that the person is climbing with the assistance of the rope. Based on common sense, if the rope were to break, the person would fall to the ground from a height. The most likely outcome is that the person would break a bone.

### B.3 Cross-task Synergy Learning

Let's denote the *task-specific features* as $\boldsymbol{v}^p$ and *task-invariant fine-grained features* as $\boldsymbol{v}^s$. We will concatenate them as one unified feature embedding $\boldsymbol{v} = [\boldsymbol{v}^p; \boldsymbol{v}^s]$ and then send $\boldsymbol{v}$ to the downstream module $M_k$ for the $k$-th task prediction:

$$y_k \leftarrow \mathrm{M}(\boldsymbol{v}). \tag{1}$$

Although we separate the two feature spaces into the shared one $\boldsymbol{v}^s$ and private one $\boldsymbol{v}^p$, there are still chances that the learned shared and the private features are closely entangled, weakening the refining of the shared task-invariant fine-grained feature. Therefore, we employ a third-party task discriminator with adversarial training to refine the features. The discriminator $D$ is a classifier for predicting what the current task is, based merely on the task-invariant fine-grained feature representation $\boldsymbol{v}^s$. Ideally, once the discriminator cannot accurately identify the task ID $y_k^d$, the task-invariant fine-grained feature representation $\boldsymbol{v}^s$ can be understood as the most purified one. Specifically, the discriminator is a 4-layer 768-d Transformer (Trm) network, where we use a feedforward layer (FFN) with Softmax for the task prediction:

$$\boldsymbol{v}^{'} = \mathrm{Trm}(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n), \tag{2}$$

$$\bar{y}_k^d = \mathrm{Softmax}(\mathrm{FFN}(\boldsymbol{v}^{'})), \tag{3}$$

where $\bar{y}_k^d$ is the predicted task ID.

The target of this adversarial training is to urge the shared features such that the discriminator cannot reliably predict the task ID:

$$\mathcal{L}^{syn} = \min_\theta(\max_D(\sum_k \bar{y}_k^d \log(y_k^d))). \tag{4}$$

[2] https://openai.com/index/gpt-4/

### B.4 Overall Training Remarks

All our framework is trained through three main stages, in a specific ordering of sub-steps:

- **Step-1:** Basic Multimodal Comprehension and Generation Skill Training, cf §4.1.
  - **Step-1.1:** Aligning the encoder-LLM for overall vision-language alignment learning.
  - **Step-1.2:** Doing text invocation instruction tuning such that the MLLM learns to output text instructions in the correct format.
  - **Step-1.3:** When the above step is converging, training the LLM with continuous soft embedding-oriented LLM-decoder alignment, such that the LLM overall can convey the signal to the downstream modules.
- **Step-2:** Fine-grained Spatiotemporal Vision Grounding Instruction Tuning, cf §4.2.
  - **Step-2.1:** Starting by doing the Image Spatial Grounding training, on the grounded image captioning task and referring image segmentation task.
  - **Step-2.2:** When MLLM has the ability for fine-grained spatial understanding, doing the Video Spatial-Temporal Grounding training, on the grounded video captioning task and referring video tracking task.
  - **Step-2.3:** When the MLLM has learned to have the competent ability of both image and video spatiotemporal understanding at the perception level, doing the Grounding-aware Vision QA task at the cognition level.
- **Step-3:** As the final step, when the overall system has learned to have a competitive ability in various visual tasks, dining the cross-task synergy learning, cf §4.3. This should be done by combining both the adversarial training ($\mathcal{L}^{syn}$) with the end-task prediction ($\mathcal{L}_k$). So the total loss of the step-3 is: $\mathcal{L}^{syn} + \sum_k \mathcal{L}_k$.

## C Extended Experimental Settings

We quantify the performance of VITRON on a variety of standard benchmarks for downstream vision tasks and compare it against some of the currently strong-performing systems. Given the countless vision tasks within the community, our experiments focus only on 1-2 of the most representative tasks from each task category for validation. To ensure a fair comparison, all subsequent experiments adopt settings same or similar to those of baseline systems, with evaluations following established practices. Before experiments, we perform targeted pre-training on all of VITRON's backend modules (such as GLIGEN and SEEM) on their respective datasets. This ensures our system is optimized for the best possible performance during testing. Our approach centers on training the linear projection layers of all encoders and efficiently fine-tuning the language model using LoRA.

Our backbone LLM is Vicuna[3], 7B, version 1.5. The CLIP-ViT encoders for both images and videos are with a patch size of 14, and convert all images and video frames into 336px resolutions. The task discriminator in our synergy module is with a Transformer architecture, with 4 layers and each in 768-d representation. To train our model, we employ the AdamW optimizer along with a learning rate scheduler. The pre-training of VITRON unfolds in three phases, all conducted on 10$\sim$16 $\times$ A100 (80G) GPUs. Initially, we train the model using a global batch size of 128 and a maximum learning rate of 3e-4, a process that takes approximately 40 hours. In the second tuning phase, we adjust the model with a maximum learning rate of 1e-5, utilizing a global batch size of 90. This stage of training lasts about 35 hours. The third phase of training employs a global batch size of 128 and maintains the maximum learning rate of 1e-5, completing in roughly 10 hours.

## D More Experiment Results

### D.1 Vision Segmentation

**Video Segmentation.** Table 15 presents the comprehensive comparison of VITRON with some SoTA systems in video tracking on DAVIS 17 [80] Test-Dev and Youtube-VOS 2019 [119] Val sets.

[3] https://huggingface.co/lmsys/vicuna-7b-v1.5

| Method | DAVIS 17 [80] Test-Dev | | | Youtube-VOS 2019 [119] Val | | | |
|---|---|---|---|---|---|---|---|
| | $\mathcal{J}\&\mathcal{F}$ | $\mathcal{J}$ | $\mathcal{F}$ | $\mathcal{J}_s$ | $\mathcal{F}_s$ | $\mathcal{J}_u$ | $\mathcal{F}_u$ |
| RDE [51] | 77.4 | 73.6 | 81.2 | 81.1 | 85.5 | 76.2 | 84.8 |
| XMem [14] | 81.0 | 77.4 | 84.5 | 84.3 | 89.6 | 80.3 | 88.6 |
| DeAOT [122] | 80.7 | 76.9 | 84.5 | 84.6 | 89.4 | 80.8 | 88.9 |
| ISVOS [102] | 82.8 | 79.3 | 86.2 | 85.2 | 89.7 | 80.7 | 88.9 |
| VITRON | **84.2** | **81.5** | **86.7** | **86.5** | **90.4** | **81.9** | **90.2** |

Table 15: Results of video object segmentation.

### D.2 Fine-grained Vision Understanding

**Region-level Image Understanding.** The comparisons of image-referring expression comprehension on three datasets are shown in Tables 16.

| Method | RefCOCO [40] | | | RefCOCO+ [123] | | | RefCOCOg [68] | |
|---|---|---|---|---|---|---|---|---|
| | Val | TestA | TestB | Val | TestA | TestB | Val | Test |
| OFA [103] | 80.0 | 83.7 | 76.4 | 68.3 | 76.0 | 61.8 | 67.6 | 67.6 |
| Shikra [12] | 87.0 | 90.6 | 80.2 | 81.6 | 87.4 | 72.1 | 82.3 | 82.2 |
| MiniGPT-v2 [11] | 88.7 | 91.6 | 85.3 | 79.9 | 85.1 | 74.4 | 84.4 | 84.6 |
| VITRON | **90.9** | **93.2** | **89.3** | **83.7** | **89.1** | **76.9** | **86.4** | **87.0** |

Table 16: Results (accuracy) of image referring expression comprehension.

Table 17 displays the results across 6 datasets for image-based VQA.

| Method | Grounding | OKVQA [88] | GQA [37] | VSR [62] | IconVQA [66] | VizWiz [32] | HM [41] |
|---|---|---|---|---|---|---|---|
| Flamingo [1] | ✗ | 44.7 | - | 31.8 | - | 28.8 | 57.0 |
| BLIP-2 [49] | ✗ | 45.9 | 41.0 | 50.9 | 40.6 | 19.6 | 53.7 |
| InstructBLIP [17] | ✗ | - | 49.5 | 52.1 | 44.8 | 33.4 | 57.5 |
| MiniGPT-4 [138] | ✗ | 37.5 | 30.8 | 41.6 | 37.6 | - | - |
| LLaVA [63] | ✗ | 54.4 | 41.3 | 51.2 | 43.0 | - | - |
| Shikra [12] | ✓ | 47.2 | - | - | - | - | - |
| MiniGPT-v2 [11] | ✓ | 57.8 | 60.1 | 62.9 | 51.5 | 53.6 | 58.8 |
| VITRON | ✓ | **59.4** | **62.1** | **63.9** | **52.2** | **54.7** | **60.2** |

Table 17: Results (accuracy) on image-based VQA.

**Region-level Video Understanding.** Table 18 presents the results of video QA across 4 representative datasets. Interestingly, while PG-Video-LLaVA has video grounding capabilities, it does not show better results than Video-LLaVA, which lacks grounding. However, our VITRON achieves superior performance.

| Method | Grounding | MSVD-QA [117] | | MSRVTT-QA [118] | | TGIF-QA [58] | | ActivityNet-QA [124] | |
|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy | Score | Accuracy | Score | Accuracy | Score | Accuracy | Score |
| VideoChat [50] | ✗ | 56.3 | 2.8 | 45.0 | 2.5 | 34.4 | 2.3 | - | 2.2 |
| LLaMA-Adapter [30] | ✗ | 54.9 | 3.1 | 43.8 | 2.7 | 54.3 | 3.3 | 34.2 | 2.7 |
| Video-LLaMA [128] | ✗ | 51.6 | 2.5 | 29.6 | 1.8 | 51.4 | 3.4 | 12.4 | 1.1 |
| Video-ChatGPT [67] | ✗ | 64.9 | 3.3 | 49.3 | 2.8 | 51.4 | 3.0 | 35.2 | 2.7 |
| Video-LLaVA [59] | ✗ | 70.7 | 3.9 | 59.2 | 3.5 | 70.0 | 4.0 | 45.3 | 3.3 |
| PG-Video-LLaVA [74] | ✓ | 64.1 | 3.7 | 51.6 | 3.3 | 66.8 | 3.8 | 39.9 | 3.3 |
| VITRON | ✓ | **74.9** | **4.0** | **62.0** | **3.8** | **72.4** | **4.1** | **51.0** | **3.7** |

Table 18: Results (accuracy and confidence Score) on video QA.

## E Qualitative Studies of Case Visualizations

### E.1 Vision Segmentation

Fig. 7 further demonstrates an example of how our VITRON processes image segmentation tasks in an interactive manner with the user. When users sketch or doodle outlines on specific areas of an image, VITRON is capable of accurately identifying the corresponding objects within the image. Following this, it precisely generates the bounding box and mask area for the identified objects.

Fig. 8 also demonstrates the process of video segmentation. VITRON successfully identifies, localizes, and tracks the target in the video based on the provided queries. Our system demonstrates a remarkable ability to accurately and flexibly capture the user's intent, even when the desired object to be segmented is described in a highly implicit manner.

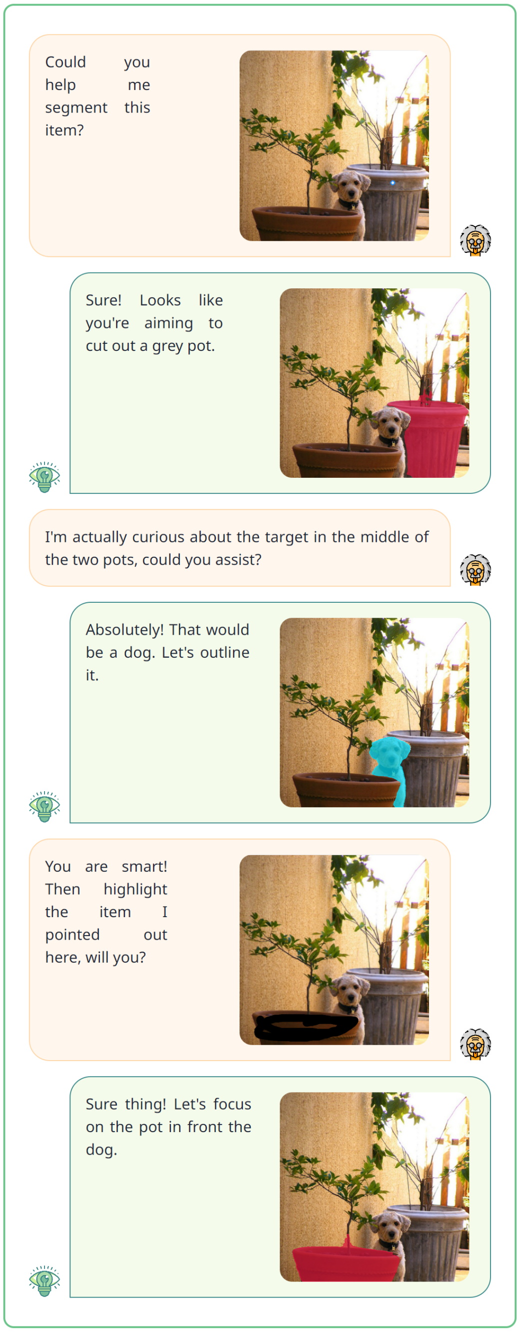



Figure 7: Demonstration of image segmentation.

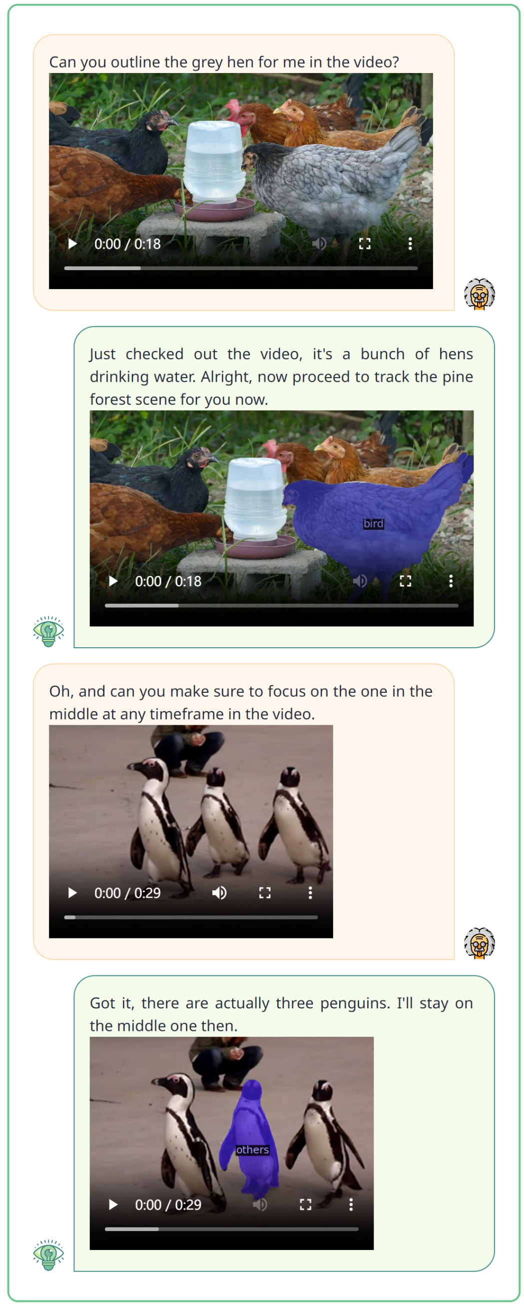



Figure 8: Example of video object segmentation.

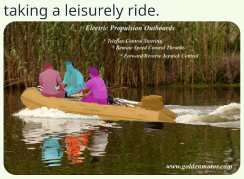
## E.2 Fine-grained Vision Understanding

Fig. 9 highlights VITRON's fine-grained visual understanding capabilities through regional image captioning, to accurately understand and appropriately describe image regions.

Further, we showcase a visualization process illustrating how VITRON accomplishes fine-grained visual understanding through video grounding tasks in Fig. 10. VITRON's precise perception of video details ensures a more accurate understanding of content semantics, leading to stronger video understanding capabilities.

Could you tell me what's happening in this photo?

The picture features a group of people riding in a small electric propulsion boat on a body of water. There are three people visible in the boat, with one person sitting in the front, another in the middle, and the third person in the back. The boat appears to be a motorboat with an electric motor, providing a more environmentally friendly mode of transportation. The people seem to be enjoying their time on the water, possibly engaging in recreational activities or simply taking a leisurely ride.



So what is the possible relationship between the two persons according to the picture?

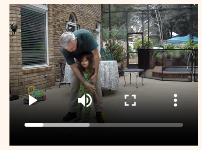
In the image, there is a man holding a tennis racket and a chair umpire on a tennis court. The relationship between the two persons is that the chair umpire is responsible for overseeing the match and making decisions related to the game, such as calling balls and strikes, while the man holding the tennis racket is a tennis player participating in the match.

Then, why the person in black siting on a such high chair?

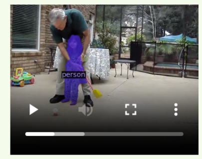
The person in black is sitting on a high chair, also known as a chair umpire's chair, to have a better vantage point to observe the tennis match and make accurate judgments. The chair's elevated position allows the umpire to have a clear view of the entire court, including the lines and the players' movements. This helps the umpire to make fair and impartial decisions during the match, such as calling balls and strikes, and ensuring that the game is played according to the rules.

Figure 9: Demonstration of fine-grained visual understanding via image grounding.

Can you tell me what's happening with the girl in the video?



Looks like the girl is getting a mini golf lesson there. She's gripping a golf stick, and there's a man bend down, to teach the girl how to swing.



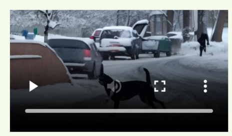
And how'd she manage to hit the ball on the floor?

She hits the ball on the floor with a golf club, which is a piece of equipment used in the sport of golf.

Figure 10: Demonstration of fine-grained visual understanding via video grounding.

I would like to see a video where a dogs run past a car on the street in the snow.

Absolutely, I've got just the video for you.



Hmm, it's cool but not quite what I had in mind. Let me paint a picture for you: how about starting with adding a white dog next to the car?

Got it, I've placed a white dog right next to the car, on its right side, to be precise.

Awesome, now could you convert that image into a dynamic video where the dog is seen running past the car?

Sure thing, here's your video showcasing the dog making a swift move past the car.

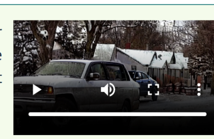


Figure 11: Demonstration of vision generation across different modalities.

## E.3 Vision Generation

Fig. 11 illustrates the process of vision generation across different modalities, including text, image, and video. Initially, users start with a basic text command, and VITRON is capable of transforming a simple idea into a detailed video. However, if users are not satisfied with the video generated directly from text, they can first generate an image from text, then fine-tune or edit this image, and finally create a satisfying video based on the adjusted image. Our VITRON, thanks to its robust interactive capability via multi-turn dialogue, enables users to perform a series of consecutive operations, ultimately facilitating smooth content creation. This fully helps meet the demands of real-world application scenarios.

## E.4 Vision Editing

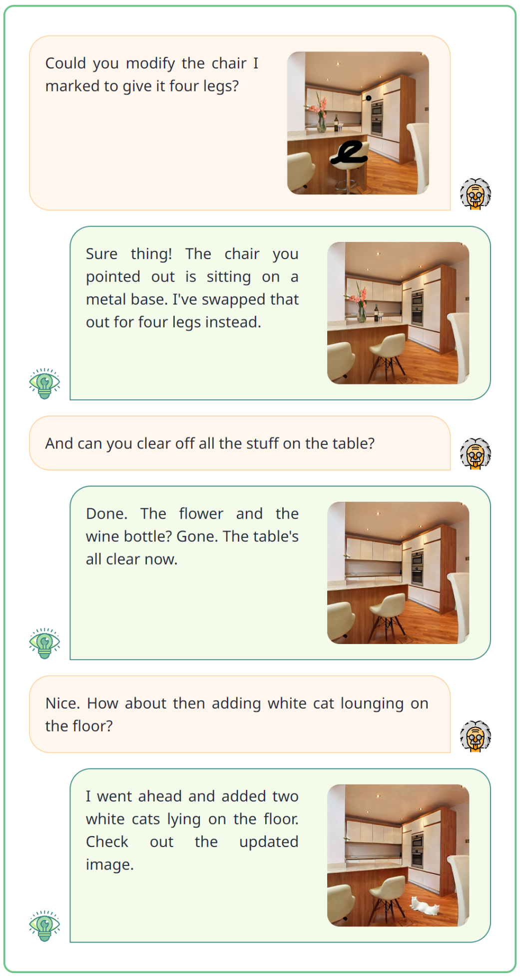



Figure 12: Demonstration of image editing.

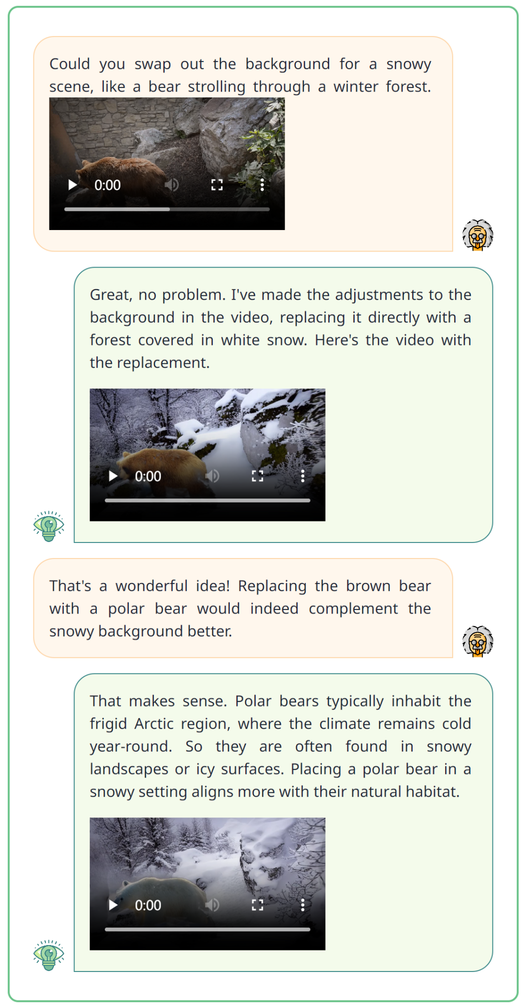



Figure 13: Demonstration of video editing.

We showcase the specific process of this image editing, as illustrated in Fig. 12. VITRON is capable of accepting different forms of user inputs (textual instruction or sketch) for precise image edits. It maintains contextual consistency throughout a series of sequential editing operations, ultimately achieving satisfactory results that meet the user's expectations.

Fig. 13 illustrates this process. VITRON competently handles video editing tasks, including modifications to the content's subject, and changes to the video's style, etc.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: in the Section 1 & 2

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: in the Experiment part.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: in the Section 3 & 4

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: in the Appendix A and Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: All the data utilized in our experiments are publicly available. We will release the code upon the acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: in the Appendix C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: in the Section 5 and Appendix D, and All results are reported after the statistical significance tests.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: in the Section C

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: the research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: [NA]

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: the paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: the paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.