JavisGPT: A Unified Multi-modal LLM for Sounding-Video Comprehension and Generation

Kai Liu[1,2*], **Jungang Li**[3*], **Yuchong Sun**[4*], **Shengqiong Wu**[2], **Jianzhang Gao**[4], **Daoan Zhang**[5], **Wei Zhang**[6], **Sheng Jin**[7], **Sicheng Yu**[8], **Geng Zhan**[9], **Jiayi Ji**[2], **Fan Zhou**[1], **Liang Zheng**[10], **Shuicheng Yan**[2], **Hao Fei**[2†], **Tat-Seng Chua**[2]

[1]ZJU, [2]NUS, [3]HKUST(GZ), [4]RUC, [5]UR, [6]HZCU, [7]NTU, [8]SMU, [9]USYD, [10]ANU

Project: https://JavisVerse.github.io/JavisGPT

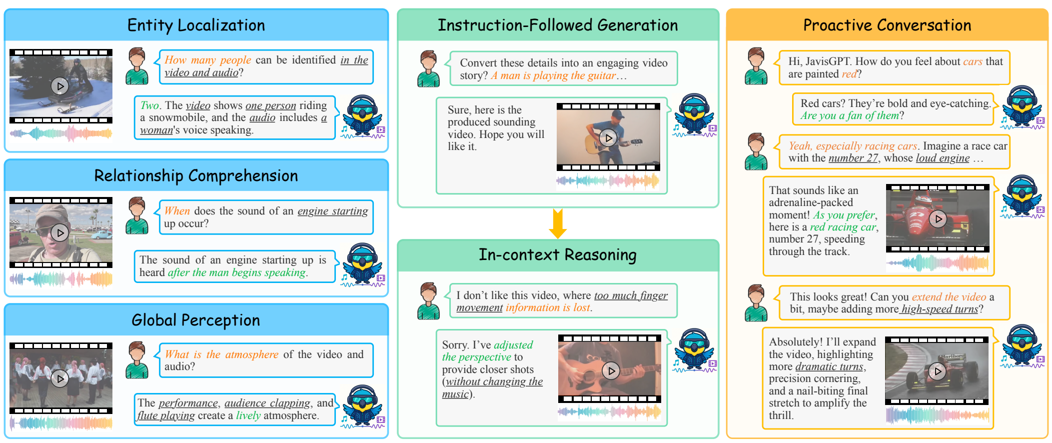


Figure 1: JavisGPT supports multi-level synchronized audio-video (SyncVA) content (i.e., sounding video) comprehension (**left**), and simultaneously complex instruction-based SyncVA generation, interleaved in-context generation (**middle**), and multi-turn proactive conversations (**right**).

Abstract

This paper presents **JavisGPT**, the first unified multimodal large language model (MLLM) for Joint Audio-Video (JAV) comprehension and generation. JavisGPT adopts a concise encoder–LLM–decoder architecture, featuring a SyncFusion module for spatio-temporal audio-video fusion and synchrony-aware learnable queries to bridge a pretrained JAV-DiT generator. This design enables temporally coherent video-audio understanding and generation from multimodal instructions. We design an effective three-stage training pipeline consisting of multimodal pretraining, audio-video fine-tuning, and large-scale instruction-tuning, to progressively build multimodal comprehension and generation from existing vision-language models. To support this, we further construct `JavisInst-Omni`, a high-quality instruction dataset with over 200K GPT-4o-curated audio-video-text dialogues that span diverse and multi-level comprehension and generation scenarios. Extensive experiments on JAV comprehension and generation benchmarks show that JavisGPT outperforms existing MLLMs, particularly in complex and temporally synchronized settings.

*Equal contribution. Work done during Kai Liu's visiting period at NUS. Email: kail@zju.edu.cn
†Corresponding author. Email: haofei7419@gmail.com

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

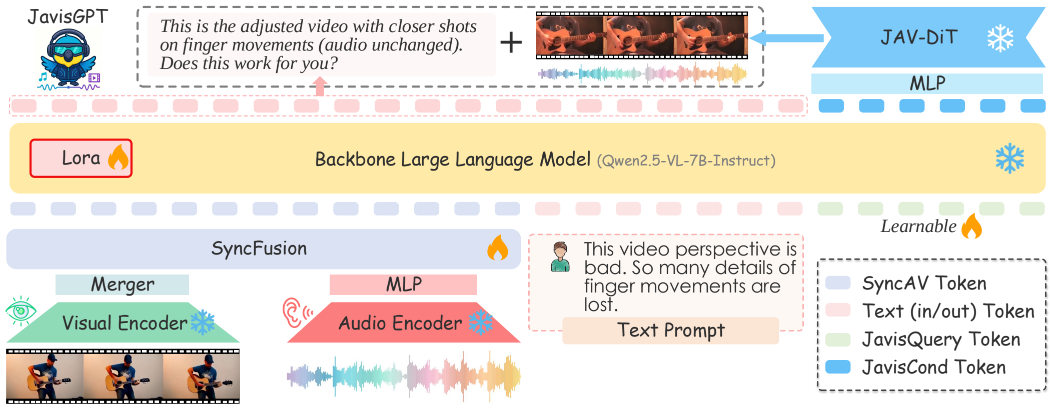


Figure 2: **Overall architecture of JavisGPT**, which can perceive and produce videos and sounds simultaneously. The input video and audio are tokenized and fed into the **SyncFusion** module. The resulting synchronized audio-video representation, together with the text tokens and learnable `JavisQueries` are then passed to the LLM backbone. During decoding, the yielded `JavisCond` embeddings are used to align the LLM intents to the semantic conditional space of downstream JAV-DiT, enabling high-quality and synchronized sounding-video generation.

1 Introduction

Joint understanding and generation of multiple modalities has recently sparked increasing interest in the community [72, 81, 21, 73, 87, 51], owing to its broader applicability compared to unimodal systems. However, current efforts predominantly focus on image-text frameworks [97, 85, 74, 52, 11], or treat video and audio understanding and generation as separate modalities [91, 82, 43]. In contrast, truly unified modeling of synchronized sounding videos—which requires precise temporal alignment between audio and visual streams for both comprehension and generation—remains significantly underexplored. In this paper, we aim to bridge this gap by designing an MLLM capable of jointly understanding and generating videos with synchronized audio (i.e., sounding video). This is a critical yet practical problem with broad real-world applications [87, 56, 32], such as dynamic avatar chatbots, sounding movie curation, and music-driven video analysis.

We introduce **JavisGPT**, a bespoke MLLM for high-quality, unified comprehension and generation in joint audio-video (JAV) tasks. As its core, JavisGPT is a holistic system that accepts as input separate audio and videos, synchronized sounding videos, and user text prompts. After comprehending and reasoning over these input signals, it produces either textual responses and/or synchronized sounding videos. Our architecture supports a wide range of JAV scenarios, including question-answering in comprehending sounding audios, generation of sounding audios following user instructions, upon in-context reasoning, and interleaved text-audio-video generation in multi-turn user dialogues (ref. Fig. 1). As a comprehensive and state-of-the-art (SOTA) system, we highlight the design of JavisGPT below (with detailed discussions with related works provided in Sec. 2).

Model architecture. JavisGPT adopts an encoder-LLM-decoder architecture (ref. Fig. 2), with Qwen2.5 [88] as the LLM backbone. The visual encoder is inherited from Qwen2.5-VL [5], and the audio encoder is based on BEATs [10]. Audio and video features, along with user prompts and learnable JavisQuery tokens, are passed to the LLM. To enable fine-grained spatiotemporal alignment, we propose a dedicated SyncFusion module that fuses audio and video representations into synchronized SyncAV tokens for unified comprehension. At the output stage, the LLM generates textual responses along with JavisCond tokens, which encode contextual semantics and serve as conditioning inputs for a pretrained JAV-DiT generator [40]. We choose JavisDiT for its generation quality and flexibility, and incorporate hierarchical JavisQueries to provide spatiotemporal priors, further enhancing synchronization in audio-video generation.

Training strategies. To effectively adapt the vision-language backbone to JAV tasks, we design a three-stage training pipeline: (1) *MM-PreTrain*: introducing the audio input branch for comprehension and preliminarily aligning the output embeddings of LLM with the condition space of JavisDiT; (2) *AV-FineTune*: enhancing synchronized audio-video comprehension and generation via SyncFusion and hierarchical JavisQueries, respectively; and (3) *MM-InstTune*: enables generalizable instruction following and multimodal reasoning via large-scale instruction tuning across text, video, and audio.

Dataset collection. To facilitate instruction tuning, we construct `JavisInst-Omni`, a large-scale, diverse, and high-quality instruction dataset covering both comprehension and generation. The dataset contains 200K dialogue trajectories involving tightly interleaved text, audio, and video, simulating complex single- and multi-turn interactions. All samples are annotated via GPT-4o and manually verified for quality. The instructions span a wide range of task formats, including QA, captioning, instruction-following generation, and in-context multimodal reasoning.

We compare JavisGPT with existing JAV-like LLMs on various JAV benchmarks, where JavisGPT yields SOTA performance across all scenarios. Moreover, for audio-video joint generation, JavisGPT also significantly outperforms existing methods, producing high-quality and diverse sounding videos. We reiterate our main points below:

- We present JavisGPT, the first unified MLLM for sounding-video comprehension and generation.
- We present its architecture design, including the audio-video synchronization mechanism, its multi-stage tuning strategy, and a large-scale instruction dataset for JAV LLMs training.
- Evaluation indicates the effectiveness of JavisGPT in JAV comprehension and generation.

2 Related Work

Multimodal comprehension. Recently, MLLMs have significantly advanced many multimodal fields, such as the modeling of image, video, and audio contents [5, 12, 18, 61, 22, 20, 80, 56]. Early approaches, such as Flamingo [4] and BLIP [34, 35], focused on combining text with *static* visual inputs. Recent research extended these efforts to *dynamic* vision and audio modalities, such as PandaGPT [68], Video-LLaMA [92, 14], UnifiedMLLM [37], and MiniOmni [86], which incorporate temporal and auditory perception and reasoning. While these MLLMs achieve promising performance, they typically treat modalities (*e.g.*, audio and video) as independent inputs from separate channels, relying on concatenation or simple feature padding for integration. This strategy overlooks fine-grained spatio-temporal interactions between modalities, compromising their effectiveness in complex scenarios where multimodal synchronization is required, such as JAV.

Multimodal generation. Compared to research efforts for multimodal content comprehension, relatively limited emphasis has been placed on generative capabilities for MLLMs, particularly for scenarios involving multiple interdependent modalities, such as JAV [31, 56]. For audio-video generation, existing MLLMs like HuggingGPT [65], NExT-GPT [82], and UnifiedIO-2 [43] adopt a pipeline-based framework, where individual audio and video generation components are sequentially executed. Although such methods achieve reasonable results, they suffer from issues like error propagation and desynchronized outputs due to the lack of joint modeling.

Unified comprehension and generation. Early approaches such as AV2AV [15] and MultiDialog [53] tackled talking-head generation and speech-driven dialogue, but were limited by simple architectures or the absence of textual input/output. While recent MLLMs aim to unify multimodal comprehension and generation, most still focus on single modalities, lacking effective modeling of cross-modal interactions, particularly the audio-video synchronization. For instance, Video-Salmonn [69, 71] and Meerkat [16] use separate encoders for audio and video with little attention to alignment. Similarly, NExT-GPT [82] and AnyGPT [91] support multimodal generation but fail to synchronize audio and video due to separate decoders. To address these limitations, we propose JavisGPT, which integrates a spatio-temporal alignment mechanism and unified feature learning, advancing the state of the art in joint audio-video (JAV) modeling.

3 Architecture

JavisGPT leverages Qwen2.5-VL-7B-Instruct [5] as backbone, incorporating with SyncFusion and JavisQueries to perform joint comprehension and generation. Sec. 3.1 demonstrates how JavisGPT extends to audio perception and utilizes SyncFusion to explicitly capture spatio-temporal synchrony in input sounding videos. Sec. 3.2 introduces how JavisGPT integrates with a pretrained JAV generator [40] and enhance the generation synchrony through hierarchical JavisQueries.

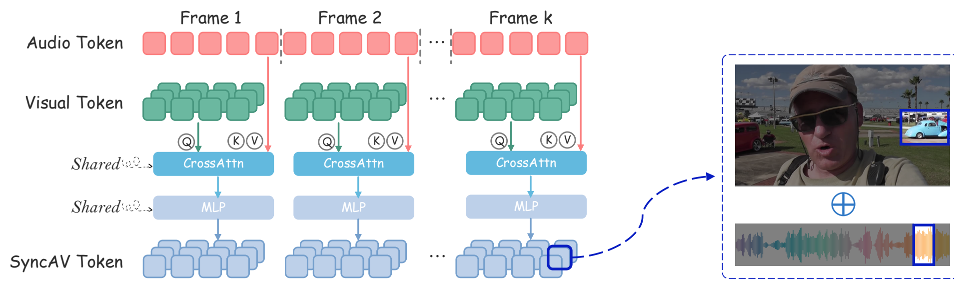


Figure 3: **Mechanism of SyncFusion.** (1) **Left**: We align temporally-segmented audio tokens with corresponding video frames by using cross-attention to merge audio clues into visual patches, so as to capture spatiotemporal synchrony explicitly. (2) **Right**: Each resulting SyncAV token $e^t_{i,j} \in \mathbb{R}^C$ represents a sounding event occurring within the i-th row, j-th column visual patch of t-th frame.

3.1 Synchrony-aware Audio-Video Comprehension

Modality-specific encoding for video and audio. Following recent works [14, 69, 71], we first adopt the vision encoder from Qwen2.5-VL [5] to encode the video inputs to yield dense video representation $\mathbf{V} \in \mathbb{R}^{T_v \times (H \times W) \times C}$, where T_v denotes the number of video frames, (H, W) are the spatial dimensions, and C is the feature dimensionality. Meanwhile, we introduce a parallel audio encoding branch built upon BEATs [10] to extract dense audio representations $\mathbf{A} \in \mathbb{R}^{T_a \times M \times C}$, where T_a is the temporal length of the audio features, M is the number of frequency bins, and C is the feature dimension. Specifically, the raw audio waveform is first converted into a 2D mel-spectrogram, comprising T_a time frames and $M = 128$ frequency bins. Each audio frame corresponds to a 25 ms window with a 10 ms overlap, enabling high-resolution temporal encoding. Finally, a two-layer MLP ϕ^a_θ is applied to project audio features into the LLM token embedding space, serving as a bridge for JavisGPT to understand audio content.

Synchronized audio-video fusion. Unlike previous MLLMs that straightforwardly concatenate video and audio features together [14, 69] or simply interleave audio-video frames [71, 87], JavisGPT introduces a SyncFusion module ψ^{av} to explicitly model the spatio-temporal synchrony in sounding videos. As illustrated in Fig. 3, we first temporally align the audio and video streams by evenly dividing the audio representations to match the number of video frames, transforming the audio features from $\mathbf{A} \in \mathbb{R}^{T_a \times M \times C}$ to $\mathbf{A}' \in \mathbb{R}^{T_v \times (r \times M) \times C}$, where $r = \lceil T_a / T_v \rceil$, with zero-paddings applied if needed. Then, a shared cross-attention module is employed to inject audio cues into the frame-wise video representations, followed by a two-layer MLP to generate *SyncAV* representations. The length of *SyncAV* representations matches that of the vanilla extracted video representations, ensuring compatibility with the MRoPE positional encoding used in the Qwen2.5-VL [5]. Each resulting token $e^{t_v}_{i,j} \in \mathbb{R}^C$ encodes a localized audio-visual event occurring at the i-th row, j-th column visual patch of frame t_v. By integrating audio semantics into visual features, the SyncFusion module enhances representational expressiveness and achieves precise spatio-temporal synchronization—crucial for downstream tasks requiring fine-grained audio-video understanding.

3.2 Spatio-Temporally Synchronized Audio-Video Generation

Equipping backbone MLLM with DiT generator. Drawing inspiration from Pan et al. [52], we introduce N learnable query embeddings Q^c to bridge the hidden state space of the LLM with the semantic condition space of the DiT generator. As demonstrated in Fig. 4, when JavisGPT encounters the special token `<|av_start|>` during the NTP process, the N learnable tokens are immediately inserted into the input sequence. These tokens are processed through the causal attention layers of the LLM, after which a two-layer MLP ϕ^c_θ are applied to project the N hidden states $\mathbf{h} = \mathbb{R}^{N \times C}$ into the condition embedding space of the DiT, yielding $\hat{\mathbf{c}} = \mathbb{R}^{N \times D}$, which serve as semantic conditions for generating synchronized sounding videos. Unlike MetaQueries [52], which completely replaces the original conditions $\mathbf{c}$ with $\hat{\mathbf{c}}$ and require full fine-tuning of the DiT, the proposed JavisGPT retains the original DiT parameters by freezing the generator and instead aligns $\hat{\mathbf{c}}$ with the original T5-XXL [60] text encoder with alignment loss $\mathcal{L}^c_{align} = \| \hat{\mathbf{c}} - \mathbf{c} \|_2$ [82]. This design greatly simplifies training and reduces computational cost while maintaining effective conditioning for generation, and Appendix D provides further discussions with other generative MLLMs.

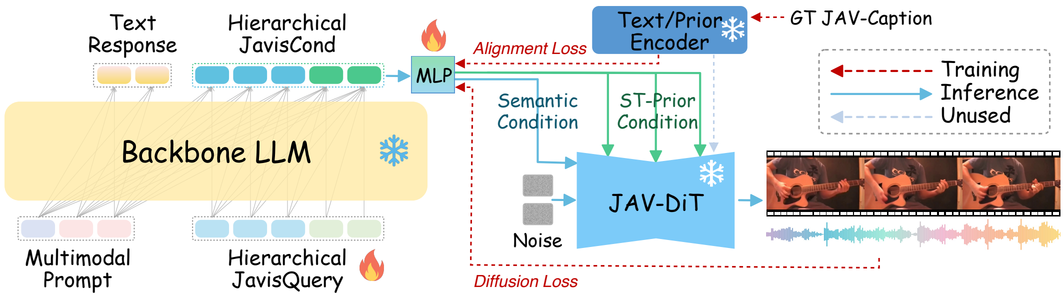


Figure 4: **Instruction-followed and synchronized audio-video generation.** We use learnable queries to gather useful information from all the audio, video, and text inputs, and use a two-layer MLP to map the hidden states from LLM to the conditional space of DiT. The hierarchical semantic and spatiotemporal prior conditions further enhance the synchrony in generated sounding videos. Alignment loss and diffusion loss are integrated to reduce optimization difficulty and training cost.

Synchrony-enhanced audio-video generation. To further strengthen the spatio-temporal synchrony of generated sounding videos, we follow Liu et al. [40] to additionally design N' learnable query tokens Q^s to estimate spatio-temporal prior (ST-prior) embeddings $\hat{\mathbf{s}}$ for auxiliary conditions. As shown in Fig. 4, the semantic condition $\hat{\mathbf{c}}$ a coarse description of the overall audio-visual event (*e.g.*, 'a car starts its engine and leaves the screen'), while the ST-prior condition $\hat{\mathbf{s}}$ captures fine-grained synchrony. Specifically, the spatial prior governs where events occur and disappear (*e.g.*, 'the car is in the top-left corner of screen'), whereas the temporal prior determines when they begin and end (*e.g.*, 'sound starts at 2s, exits at 7s, fades at 9s'). To compute $\hat{\mathbf{s}}$, we employ a separate MLP projector ϕ^s_θ to map the final-layer representations of the prior query tokens into the ST-prior condition space, aligning them with the ST-prior encoder used in JavisDiT [40]. Finally, we combine the alignment loss between the predicted ST-prior embeddings $\hat{\mathbf{s}}$ and the ground-truth ST-priors $\mathbf{s}$ with the denoising objective, jointly optimizing both to strengthen the audio-video synchrony:

$$\mathcal{L}_{comp} = \| \hat{\mathbf{c}} - \mathbf{c} \|_2 + \| \hat{\mathbf{s}} - \mathbf{s} \|_2 + \| \epsilon - \hat{\epsilon}(\mathbf{a}_t, \mathbf{v}_t, \hat{\mathbf{c}}, \hat{\mathbf{s}}, t) \|_2 \triangleq \mathcal{L}^c_{align} + \mathcal{L}^s_{align} + \mathcal{L}^{c,s}_{diff} \,, \quad (1)$$

where $\hat{\epsilon}(*)$ is the DiT model that predicts the noise/velocity added to video $\mathbf{v}$ and audio $\mathbf{a}$ at timestamp t with (estimated) text condition $\hat{\mathbf{c}}$ and st-prior condition $\hat{\mathbf{s}}$, and ϵ is the ground-truth. We follow Liu et al. [40] to use rectified flow [41] as the scheduler.

4 Training and Datasets

This section presents the three-stage training pipeline designed to efficiently adapt the visual-language backbone MLLM [5] into our proposed JavisGPT for joint comprehension and generation of sounding videos. A brief summary of the training process is demonstrated in Tab. 1, with detailed training configurations provided in Appendix B.2.

Table 1: **Summary of the training pipeline**. 'MM-PT' denotes *multimodal pre-training*, 'AV-FT' refers to *audio-video fine-tuning*, and 'MM-InstTune' means *multimodal instruction-tuning*.

Stage	Tasks	Trainable Modules	# Parameters	Training Objective
MM-PT	A→T, T→AV	$\phi^a, Q^{c,s}, \phi^{c,s}$	174.36M	$\mathcal{L}_{ntp} + \mathcal{L}_{align}$
AV-FT	AV→T, T→AV	$\psi^{av}, Q^{c,s}, \phi^{c,s}, \Psi^{\text{LLM}}_{lora}$	589.09M	$\mathcal{L}_{ntp} + \mathcal{L}_{align} + \mathcal{L}_{diff}$
MM-InstTune	A/V/AV+T→T+AV	$\phi^a, \psi^{av}, Q^{c,s}, \phi^{c,s}, \Psi^{\text{LLM}}_{lora}$	651.50M	$\mathcal{L}_{ntp} + \mathcal{L}_{align} + \mathcal{L}_{diff}$

4.1 Pre-training for Basic Video-Audio Comprehension and Generation

The first stage equips JavisGPT with basic AV comprehension and generation skills by extending Qwen2.5-VL to audio and aligning it with the JavisDiT generator.

Audio input alignment learning. As described in Sec. 3.1, we optimize the only parameters of the two-layer MLPs ϕ^a, which are applied to project the audio features (encoded by BEATs [10]) to the

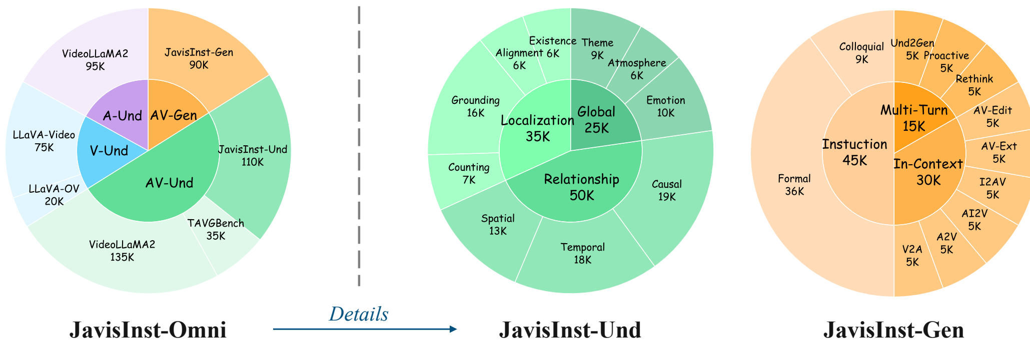


Figure 5: (1) **Left**: We curate a large-scale, diverse, and balanced cross-modal instruction-tuning dataset (`JavisInst-Omni`) from multiple sources. (2) **Right**: Within `JavisInst-Omni`, `JavisInst-Und` and `JavisInst-Gen` are specifically developed for multi-level audio-video comprehension and generation with synchrony-awareness. Details are presented in Appendix C.

input token embedding space of LLM. Specifically, following the setup of VideoLLaMA2 [14], we train ϕ^a on approximately 600K audio-text pairs using audio captioning and audio-based question-answering tasks by minimizing the next-token prediction loss $\mathcal{L}_{\text{ntp}}$ on the generated text responses.

Preliminary audio-video generation alignment learning. In this part, we learn N learnable query embeddings Q^c along with the projection network ϕ^c to map the LLM's hidden states into the condition space of the DiT generator, which is originally modeled by a T5-XXL encoder [60]. Specifically, The learnable query embeddings Q^c aggregated contextual information from the LLM is Specifically, we use 1.5M sounding video captions from TAVGBench [47] as input prompts, each concatenated with the N queries, which serve to aggregate contextual information from the LLM. Then, the resulting hidden states Q^c corresponding to these queries are projected by ϕ^c to obtain the estimated condition embeddings $\hat{\mathbf{c}}$. To accelerate and stabilize the semantic alignment between the LLM and DiT, we minimize an alignment loss $\mathcal{L}_{\text{align}}$ between the predicted conditions $\hat{\mathbf{c}}$ and the reference conditions $\mathbf{c}$ encoded by the frozen T5-XXL from the same captions. This pre-alignment phase is crucial for providing semantically meaningful initialization, which in turn facilitates subsequent training of fine-grained audio-video synchronization.

4.2 Fine-tuning for Synchronized Audio-Video Comprehension and Generation

Further, the model is fine-tuned to enhance its ability to capture fine-grained audio-video synchrony. LoRA [27] ($r = 128, \alpha = 256$) is integrated with the LLM backbone to strengthen the adaptation.

Synchrony-aware audio-video comprehension. This part focuses on optimizing the SyncFusion module ψ^{av} to strengthen spatio-temporal alignment of the audio-video representations. The module is trained on the sounding-video caption task, where sounding videos are provided as input, and the LLM is required to generate visual and acoustic descriptions. We use around 360K audio-video-text triplets from TAVGBench [47], with supervision provided by the NTP loss.

Synchrony-aware audio-video generation. In this stage, we further fine-tune the semantic and st-prior queries Q^c, Q^s along with the output projectors ϕ^c, ϕ^s, aiming to improve the synchrony of generated sounding videos. Training is conducted by minimizing the loss defined in Equation (1), allowing the model to better align semantic intent with fine-grained spatio-temporal dynamics during generation. We reuse the collected 360K audio-video-text triplets from TAVGBench for this stage.

4.3 Instruction-tuning for Interleaved Comprehension and Generation

As an interactive chatbot, JavisGPT is expected to not only be equipped with instruction-following capabilities but also improve its ability to reason over user intent for more generalized and user-friendly interaction. To this end, we first construct a large-scale, diverse instruction tuning dataset, named `JavisInst-Omni`, to enable flexible and interleaved audio-video comprehension and generation. The overall dataset statistics is depicted in Fig. 5.

JavisInst-Omni dataset. First, to maintain model capacity of comprehending single video and audio modalities, we sample diverse QA instances from existing resources: 95K audio QA samples

Table 2: **Comparison on audio, video, and audio-video comprehension**. We use the ActivityNet [6], Perception [54], and MVbench [36] test sets for video comprehension, the ClothoAQA [39] and TUT2017 [50] test sets for audio comprehension, and the AVQA [90], MU-AVQA [42], and AVSD [2] test sets for synchronized audio-video understanding. JavisGPT exhibits competitive performance across all benchmarks. We also report the data (#Samples) used to adapt a backbone VLM for joint audio-video comprehension. **Best** and secondary results are marked **bold** and underline respectively.

Model (7-8B)	**#Samples**	**Video**			**Audio**		**Audio-Video**		
		ActivityNet	Perception	MVbench	ClothoAQA	TUT2017	AVQA	MU-AVQA	AVSD
Video-LLM									
Video-LLaVA [38]	-	56.5	67.9	58.6	-	-	-	-	-
LLaVA-NeXT [94]	-	53.5	48.8	46.5	-	-	-	-	-
LLaVA-OV [33]	-	55.6	57.1	56.7	-	-	-	-	-
Qwen2-VL [77]	-	57.4	62.3	67.0	-	-	-	-	-
Qwen2.5-VL [5]	-	-	**70.5**	<u>69.6</u>	-	-	-	-	-
InternVL2.5 [12]	-	**58.9**	68.9	**72.0**	-	-	-	-	-
Audio-LLM									
Qwen-Audio [17]	-	-	-	-	57.9	64.9	-	-	-
Qwen2-Audio [18]	-	-	-	-	60.9	73.7	-	-	-
Audio-Video-LLM									
NExT-GPT [82]	1.9M	21.5	33.7	27.9	30.9	6.4	25.3	19.3	30.8
UnifiedIO-2 [43]	9.2B	23.2	34.7	30.4	31.4	9.1	61.2	34.1	16.5
Macaw-LLM [45]	-	-	-	-	-	-	78.7	31.8	34.3
AV-LLM [66]	-	47.2	-	-	-	-	78.7	45.2	52.6
VideoLLaMA [92]	-	12.4	-	-	-	-	80.9	36.6	36.7
VideoLLaMA2 [14]	1.9M	50.2	51.4	54.6	65.1	78.4	-	79.2	57.2
VideoLLaMA2.1 [14]	1.9M	53.0	54.9	57.3	66.3	77.3	-	80.9	57.2
Qwen2.5-Omni [87]	-	57.2	<u>70.4</u>	66.9	**68.0**	<u>78.3</u>	91.5	79.9	**62.8**
JavisGPT (Ours)	1.5M	<u>58.1</u>	70.2	68.4	<u>67.3</u>	**82.1**	**93.8**	**82.1**	<u>62.2</u>

from VideoLLaMA2 [14], 75K video QA samples from LLaVA-Video-178K [96], and 25K image QA instances from LLaVA-OneVision [33]. Then, to enhance joint audio-video comprehension, except for 135K audio-video QA instances from VideoLLaMA2 [14] and 35K audio-video caption samples from TAVGBench [47], we additionally construct `JavisInst-Und` for multi-level synchrony-aware comprehension. Finally, to strengthen synchronized sounding video generation, we curate `JavisInst-Gen` to handle various conversation scenarios. We use GPT-4o to re-annotate samples from TAVGBench [47] to obtain the two specialized subsets. Appendix C provides more details.

Joint instruction-tuning across modalities. To enable JavisGPT supports diverse tasks shown in Fig. 1, we jointly fine-tune all newly introduced modules, including input/output projectors $\phi^a, \psi^{av}, \phi^c, \phi^s$, learnable queries Q^c, Q^s, and the LoRA adapter Ψ^{LLM}_{lora}, on diversified cross-modal instruction-tuning instances from our proposed `JavisInst-Omni` dataset.

5 Experiments

This section presents the experimental results to evaluate the effectiveness of our proposed JavisGPT. For a fair comparison, we train JavisGPT following the three-stage pipeline described in Sec. 4, with each stage running for one epoch. We bring out zero-shot evaluations on diverse downstream tasks covering audio-video comprehension and generation using their official evaluation protocols. Additional implementation and training details can be found in Appendix B.

5.1 Comparison with the State of the Art

Multimodal comprehension. As shown in Tab. 2, our method achieves performance on par with SOTA models on *video/audio understanding*, demonstrating strong unimodal capabilities. It maintains visual comprehension strength of Qwen2.5-VL [5] backbone (70.2 *vs.* 70.5 on Perception [54]) and even slightly outperforms Qwen2.5-Omni [87] on MVBench [36] (video) and TUT2017 [50] (audio). For *audio-video synchronized understanding*, JavisGPT surpasses both Qwen2.5-Omni and VideoLLaMA2.1 [14] on AVQA [90] and MU-AVQA [42], achieving top results on both benchmarks. We attribute this to the spatio-temporal alignment capability of the proposed SyncFusion and our synchrony-aware training strategy. Notably, those results are obtained with fewer training samples (1.5M in total), highlighting its strong data efficiency.

Table 3: **Comparison on audio-video generation.** Results are reported on JavisBench-mini [40]. We achieve stronger generation performance than standalone DiT models and unified MLLMs.

Method	AV-Quality			Text-Consistency				AV-Consistency			AV-Synchrony
	FVD↓	KVD↓	FAD↓	TV-IB↑	TA-IB↑	CLIP↑	CLAP↑	AV-IB↑	CAVP↑	AVHScore↑	JavisScore↑
DiT											
MM-Diff [61]	2311.9	12.2	27.5	0.080	0.032	0.173	0.048	0.119	0.783	0.109	0.070
JavisDiT [40]	327.8	1.9	7.6	0.141	**0.184**	0.322	**0.314**	**0.203**	**0.799**	0.181	0.153
MLLM											
NExT-GPT [82]	1463.2	6.1	7.0	0.123	0.153	0.207	0.306	0.067	0.732	0.061	0.038
UnifiedIO-2 [43]	1597.4	6.4	**6.7**	0.137	0.138	0.313	0.294	0.106	0.747	0.094	0.053
JavisGPT (Ours)	**317.5**	**1.8**	7.6	**0.145**	0.180	**0.324**	0.308	0.202	0.797	**0.185**	**0.157**

Multi-modal generation. Tab. 3 compares the text-to-audio-video generation performance of JavisGPT with existing methods. JavisGPT significantly outperforms NExT-GPT [82] and UnifiedIO-2 [43] in AV quality, semantic consistency, and AV synchrony. These gains largely stem from the explicit synchrony modeling through the spatio-temporal prior projector, coupled with a carefully designed training strategy. In addition, JavisGPT slightly outperforms the base generator of JavisDiT [40], which we attribute to the stronger semantic understanding and encoding capacity of the LLM backbone [5], leading to more precise condition representations, ultimately improving the quality of generated sounding videos.

Interleaved comprehension and generation. As no existing benchmark directly measures joint understanding and generation of sounding videos, we fill this gap by carefully curating a new evaluation set comprising 100 multi-turn QA dialogues covering four types of interleaved understanding-generation tasks. We employ three unbiased volunteers to conduct human evaluation over five dimensions, covering both textual and audio-video outputs. Full setup and detailed analysis are provided in Appendix E.1 The average results are reported in Fig. 6, where our method consistently outperforms the two competing methods with joint comprehension-generation capabilities by a large margin. Notably, we observe (1) NExT-GPT [82] frequently refuses to respond or generates noisy audio-video outputs, severely impacting *instruction-following* and *generation quality*; (2) UnifiedIO-2 [43] is inferior to *context reasoning* and *proactive thinking*, failing to connect user history (*e.g.*, preferences) with current instructions.

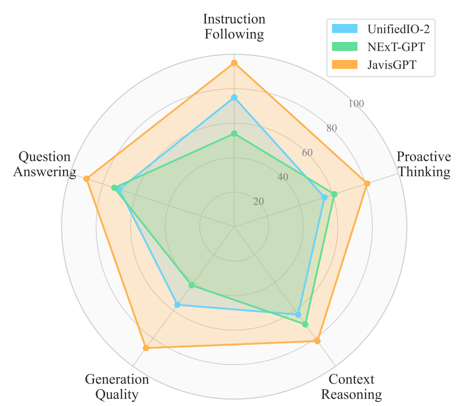


Figure 6: **Human evaluation on interleaved conversation.** JavisGPT significantly outperforms UnifiedIO-2 and NExT-GPT.

5.2 Further Analysis

Effectiveness of SyncFusion on audio-video comprehension. To assess the impact of synchronization strategies, we train models solely on the AV understanding subset of JavisInst-Omni and evaluate their performance across three AV comprehension benchmarks. In Tab. 4, we report the accuracy, the number of encoded AV tokens and per-sample inference latency. Results show that the interleaving strategy used in Video-Salmonn2 [70] and Qwen2.5-Omni [87] fails to outperform simple concatenation [38] and introduces additional latency (over 2× slowdown) due to memory-discontinuous tensor operations. The Q-former [35, 63, 82], despite offering faster inference, suffers from optimization instability [12, 5] and yields substantially lower performance. In contrast, our proposed SyncFusion achieves strong AV understanding performance while reducing both token count and inference latency, showing a more balanced trade-off between performance and efficiency.

Impact of the three training stages on high-quality AV generation. Tab. 5 analyzes the contribution of each training stages to AV generation performance. We first evaluate the model after MM-Pretrain (Sec. 4.1, without AV-FineTune) and find that merely aligning with DiT's original text encoder [60] alone is insufficient to effectively bridge the LLM and DiT, resulting in poor generation quality (*e.g.*, a low JavisScore [40] of 0.069). Next, we skip pretraining and only apply diffusion loss for AV-Finetune (Sec. 4.2, without MM-Pretrain). While this significantly improves generation quality,

Table 4: **Comparing SyncFusion with other options for AV-synchronization**. The SyncFusion module can effectively and efficiently capture the audio-video synchrony.

AVSync	AVQA	MU-AVQA	AVSD	#Tokens↓	Latency↓
Concat	93.3	80.7	61.3	3.5K	246ms
Interleave	93.3	80.6	61.6	3.5K	555ms
BiCrossAttn	93.2	80.6	61.1	3.5K	263ms
Q-Former	71.4	54.7	56.3	*0.2K*	182ms
SyncFusion	**93.4**	**81.4**	**62.0**	**2.0K**	**224ms**

Table 5: **Ablation on training stages**. We compare training without audio-video fine-tuning (w/o AV-FT), without multimodal pretraining (w/o MM-PT), and with all stages (w/ All).

Stage	Quality		Consistency			Synchrony
	FVD↓	FAD↓	TV-IB↑	TA-IB↑	AV-IB↑	JavisScore↑
w/o AV-FT	537.4	9.6	0.125	0.112	0.152	0.069
w/o MM-PT	349.6	8.4	0.137	0.171	0.178	0.135
w/ All	**317.5**	**7.6**	**0.145**	**0.180**	**0.202**	**0.157**

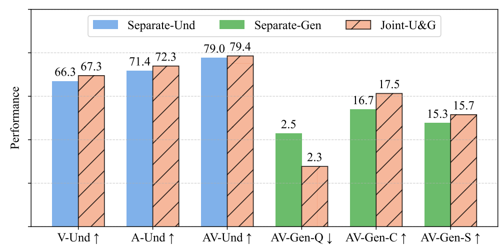


Figure 7: **Comparison on joint and separate training of understanding and generation**. Joint training generally leads to better performance, especially on the generation side.

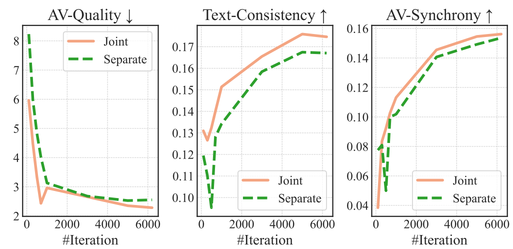


Figure 8: **Generation performance vs. training iterations**. We observe that joint training consistently improves generation quality, instruction following (text consistency) and synchronization.

consistency, and synchrony, training exhibits instability in early stages. To ensure convergence, we had to reduce the learning rate from 1e-4 to 1e-5, which ultimately compromises final performance. The best results are obtained by combining MM-Pretrain and AV-Finetune in our progressive training paradigm, confirming the necessity of staged adaptation for effective audio-video generation.

Joint training improves both understanding and generation. Fig. 7 compares the performance of comprehension and generation tasks under separate *vs.* joint training, with each score averaged over the corresponding test sets (e.g., 'V-Und' is the average accuracy over ActivityNet [6], Perception [54], and MVBench [36]). Results show that joint training consistently outperforms separate training in both understanding and generation tasks, indicating a mutual enhancement effect. The improvement is more substantial on the *generation side*.

To better understand this effect, Fig. 8 presents the evolution of generation metrics across three dimensions. As the number of generation training samples increases, the inclusion of comprehension data leads to steady gains in both AV-Quality and Text-Consistency. It suggests that semantic understanding of AV content enhances the quality of condition embeddings used for generation—a trend aligned with findings in MetaQueries [52] for image tasks. However, AV synchrony shows only marginal improvement (+0.04), possibly indicating the need for a more unified mechanism bridging comprehension and generation. We further discuss this in Appendix A.

Case study: instruction-followed and synchrony-aware comprehension and generation. Fig. 9 presents a qualitative comparison with UnifiedIO-2 [43] and NExT-GPT [82] in a two-round dialogue setting. In the first round, the models are asked to identify the source of a gunfire sound. UnifiedIO-2 gives a general response ('a semi-automatic weapon'), while NExT-GPT hallucinates an explosion of a flower. In contrast, our model accurately grounds the sound in a video game scene, showing better spatio-temporal grounding and scene understanding. In the second round, the models are asked to generate a gunfire-related video. UnifiedIO-2 produces irrelevant key frames, NExT-GPT fails to generate, while our JavisGPT successfully synthesizes coherent and synchronized audio-visual content. More examples are presented in Appendix F.

6 Conclusion and Discussion

In this work, we propose **JavisGPT**, a unified MLLM for synchronized AV comprehension and generation. Built upon an encoder-LLM-decoder architecture, JavisGPT introduces a SyncFusion

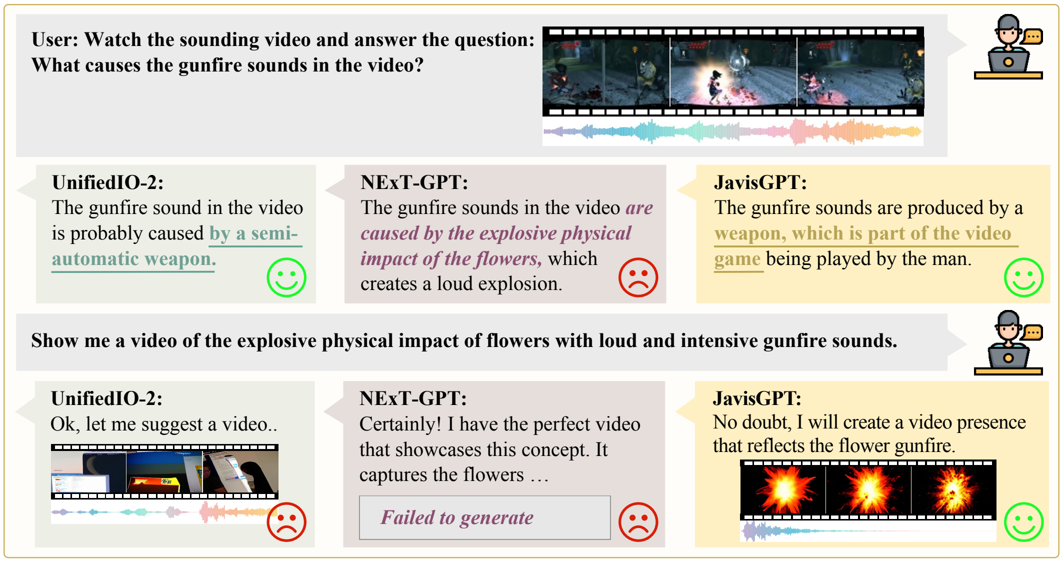


Figure 9: **Case study on joint audio-video understanding and generation.** (1) *Round-I* evaluates the instruction-based AV understanding; (2) *Round-II* validates on context-based AV generation. JavisGPT can faithfully understand and generate high-quality synchronous audio-video content.

module for spatio-temporal alignment and integrates a pretrained JavisDiT generator via hierarchical condition embedding. Moreover, we establish a three-stage training pipeline tailored for synchronized generation and curate a large-scale `JavisInst-Omni` dataset for complex audio-video-text interactions. Extensive experiments demonstrate state-of-the-art performance on both unimodal and synchronized JAV tasks, with superior generation quality, consistency, and synchrony. We believe JavisGPT offers a strong foundation for future research in synchronized media generation.

Discussion 1: the extended capabilities for sounding video comprehension and generation. In the future, JavisGPT can be further extended to achieve higher fidelity in several directions: (1) Support for speech input and output [87], which can be enabled by replacing the current BEATs [10] tokenizer with more advanced alternatives such as Whisper [58] or WavTokenizer [28]; (2) Fine-grained controllable audio-video generation and editing, which requires the downstream JAV-DiT to be conditioned on additional control signals for precise generation [29]; (3) Complex instruction-based generation, which involves enabling the backbone LLM to perform reasoning over sophisticated prompts [79], and providing dense contextual embeddings to JAV-DiT that also requires fine-tuning to adapt to the shifted input distribution.

Discussion 2: the ultimate framework for joint audio-video comprehension and generation. The encoder–LLM–decoder architecture adopted in this work is a well-established and widely validated paradigm [91, 82, 43], characterized by continuous token representations on the input side and a diffusion-based decoder/generator on the output side. In parallel, a growing number of unified models have emerged that perform end-to-end auto-regression (AR) over multimodal tokens [78, 83, 74, 76]. Both discrete [83] and continuous [76] AR-based approaches have demonstrated significant mutual enhancement between understanding and generation tasks. Building upon these advances, extending from text–image to joint text–audio–video understanding and generation holds great potential for achieving even more remarkable capabilities.

Acknowledgement

This research/project is supported inpart by the National Research Foundation, Singapore, under its National Large Language Models Funding Initiative (AISG Award No: AISG-NMLP-2024-002), in part by Fundamental Research Funds for the Central Universities, and in part by Zhejiang Provincial Natural Science Foundation of China under Grant No. LDT23F01013F01. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Huda AlAmri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K. Marks, Chiori Hori, Peter Anderson, Stefan Lee, and Devi Parikh. Audio visual scene-aware dialog. In *Proceedings of the CVPR*, pages 7558–7567, 2019.

[3] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, et al. Audio visual scene-aware dialog. In *Proceedings of the CVPR*, pages 7558–7567, 2019.

[4] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Proceedings of the NeurIPS*, 35:23716–23736, 2022.

[5] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[6] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the CVPR*, pages 961–970, 2015.

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[8] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020.

[9] Jiuhai Chen, Zhiyang Xu, Xichen Pan, Yushi Hu, Can Qin, Tom Goldstein, Lifu Huang, Tianyi Zhou, Saining Xie, Silvio Savarese, et al. Blip3-o: A family of fully open unified multimodal models-architecture, training and dataset. *arXiv preprint arXiv:2505.09568*, 2025.

[10] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058*, 2022.

[11] Xiaokang Chen, Zhiyu Wu, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, and Chong Ruan. Janus-pro: Unified multimodal understanding and generation with data and model scaling. *arXiv preprint arXiv:2501.17811*, 2025.

[12] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[13] Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *arXiv preprint arXiv:2311.00213*, 2023.

[14] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024.

[15] Jeongsoo Choi, Se Jin Park, Minsu Kim, and Yong Man Ro. Av2av: Direct audio-visual speech to audio-visual speech translation with unified audio-visual speech representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27325–27337, 2024.

[16] Sanjoy Chowdhury, Sayan Nag, Subhrajyoti Dasgupta, Jun Chen, Mohamed Elhoseiny, Ruohan Gao, and Dinesh Manocha. Meerkat: Audio-visual large language model for grounding in space and time. In *Proceedings of the ECCV*, pages 52–70. Springer, 2024.

[17] Yunfei Chu, Jin Xu, Xiaohuan Zhou, Qian Yang, Shiliang Zhang, Zhijie Yan, Chang Zhou, and Jingren Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models. *arXiv preprint arXiv:2311.07919*, 2023.

[18] Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, et al. Qwen2-audio technical report. *arXiv preprint arXiv:2407.10759*, 2024.

[19] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE, 2020.

[20] 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 International Conference on Machine Learning*, 2024.

[21] Hao Fei, Shengqiong Wu, Hanwang Zhang, Tat-Seng Chua, and Shuicheng Yan. Vitron: A unified pixel-level vision llm for understanding, generating, segmenting, editing. In *Proceedings of the Advances in neural information processing systems*, 2024.

[22] 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.

[23] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[24] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the CVPR*, pages 15180–15190, 2023.

[25] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[26] Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. Audioclip: Extending clip to image, text and audio. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 976–980. IEEE, 2022.

[27] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Proceedings of the ICLR*, 1(2):3, 2022.

[28] Shengpeng Ji, Ziyue Jiang, Wen Wang, Yifu Chen, Minghui Fang, Jialong Zuo, Qian Yang, Xize Cheng, Zehan Wang, Ruiqi Li, et al. Wavtokenizer: an efficient acoustic discrete codec tokenizer for audio language modeling. In *The Thirteenth International Conference on Learning Representations*, 2024.

[29] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. *arXiv preprint arXiv:2503.07598*, 2025.

[30] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, 2019.

[31] Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: a large language model for zero-shot video generation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 25105–25124, 2024.

[32] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

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

[34] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the ICML*, pages 12888–12900, 2022.

[35] Junnan Li, Dongxu Li, Silvio Savarese, and Steven 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.

[36] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the CVPR*, pages 22195–22206, 2024.

[37] Zhaowei Li, Wei Wang, YiQing Cai, Xu Qi, Pengyu Wang, Dong Zhang, Hang Song, Botian Jiang, Zhida Huang, and Tao Wang. Unifiedmllm: Enabling unified representation for multi-modal multi-tasks with large language model. *arXiv preprint arXiv:2408.02503*, 2024.

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

[39] Samuel Lipping, Parthasaarathy Sudarsanam, Konstantinos Drossos, and Tuomas Virtanen. Clotho-aqa: A crowdsourced dataset for audio question answering. In *Proceedings of the EUSIPCO*, pages 1140–1144, 2022.

[40] Kai Liu, Wei Li, Lai Chen, Shengqiong Wu, Yanhao Zheng, Jiayi Ji, Fan Zhou, Rongxin Jiang, Jiebo Luo, Hao Fei, et al. Javisdit: Joint audio-video diffusion transformer with hierarchical spatio-temporal prior synchronization. *arXiv preprint arXiv:2503.23377*, 2025.

[41] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *Proceedings of the ICLR*, 2023.

[42] Xiulong Liu, Zhikang Dong, and Peng Zhang. Tackling data bias in music-avqa: Crafting a balanced dataset for unbiased question-answering. In *Proceedings of the CVPR*, pages 4478–4487, 2024.

[43] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the CVPR*, pages 26439–26455, 2024.

[44] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. *Advances in Neural Information Processing Systems*, 36:48855–48876, 2023.

[45] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*, 2023.

[46] Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Liang Zhao, et al. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *arXiv preprint arXiv:2411.07975*, 2024.

[47] Yuxin Mao, Xuyang Shen, Jing Zhang, Zhen Qin, Jinxing Zhou, Mochu Xiang, Yiran Zhong, and Yuchao Dai. Tavgbench: Benchmarking text to audible-video generation. In *Proceedings of the ACM MM*, pages 6607–6616, 2024.

[48] Irene Martín-Morató and Annamaria Mesaros. What is the ground truth? reliability of multi-annotator data for audio tagging. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pages 76–80. IEEE, 2021.

[49] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[50] Annamaria Mesaros, Toni Heittola, Aleksandr Diment, Benjamin Elizalde, Ankit Shah, Emmanuel Vincent, Bhiksha Raj, and Tuomas Virtanen. Dcase 2017 challenge setup: Tasks, datasets and baseline system. In *DCASE 2017-workshop on detection and classification of acoustic scenes and events*, 2017.

[51] Microsoft, :, Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, Dong Chen, Dongdong Chen, Junkun Chen, Weizhu Chen, Yen-Chun Chen, Yi ling Chen, Qi Dai, Xiyang Dai, Ruchao Fan, Mei Gao, Min Gao, Amit Garg, Abhishek Goswami, Junheng Hao, Amr Hendy, Yuxuan Hu, Xin Jin, Mahmoud Khademi, Dongwoo Kim, Young Jin Kim, Gina Lee, Jinyu Li, Yunsheng Li, Chen Liang, Xihui Lin, Zeqi Lin, Mengchen Liu, Yang Liu, Gilsinia Lopez, Chong Luo, Piyush Madan, Vadim Mazalov, Arindam

Mitra, Ali Mousavi, Anh Nguyen, Jing Pan, Daniel Perez-Becker, Jacob Platin, Thomas Portet, Kai Qiu, Bo Ren, Liliang Ren, Sambuddha Roy, Ning Shang, Yelong Shen, Saksham Singhal, Subhojit Som, Xia Song, Tetyana Sych, Praneetha Vaddamanu, Shuohang Wang, Yiming Wang, Zhenghao Wang, Haibin Wu, Haoran Xu, Weijian Xu, Yifan Yang, Ziyi Yang, Donghan Yu, Ishmam Zabir, Jianwen Zhang, Li Lyna Zhang, Yunan Zhang, and Xiren Zhou. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras, 2025.

[52] Xichen Pan, Satya Narayan Shukla, Aashu Singh, Zhuokai Zhao, Shlok Kumar Mishra, Jialiang Wang, Zhiyang Xu, Jiuhai Chen, Kunpeng Li, Felix Juefei-Xu, et al. Transfer between modalities with metaqueries. *arXiv preprint arXiv:2504.06256*, 2025.

[53] Se Jin Park, Chae Won Kim, Hyeongseop Rha, Minsu Kim, Joanna Hong, Jeong Hun Yeo, and Yong Man Ro. Let's go real talk: Spoken dialogue model for face-to-face conversation. *arXiv preprint arXiv:2406.07867*, 2024.

[54] Viorica Patraucean, Lucas Smaira, Ankush Gupta, Adria Recasens, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Mateusz Malinowski, Yi Yang, Carl Doersch, et al. Perception test: A diagnostic benchmark for multimodal video models. *Proceedings of the NeurIPS*, 36:42748–42761, 2023.

[55] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015.

[56] Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

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

[58] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR, 2023.

[59] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[60] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[61] Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Proceedings of the CVPR*, pages 10219–10228, 2023.

[62] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 1041–1044, 2014.

[63] Paul Hongsuck Seo, Arsha Nagrani, and Cordelia Schmid. Avformer: Injecting vision into frozen speech models for zero-shot av-asr. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22922–22931, 2023.

[64] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[65] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Proceedings of the NeurIPS*, 36:38154–38180, 2023.

[66] Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding. *arXiv preprint arXiv:2312.06720*, 2023.

[67] Bob L Sturm. The gtzan dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv preprint arXiv:1306.1461*, 2013.

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

[69] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. video-salmonn: Speech-enhanced audio-visual large language models. *arXiv preprint arXiv:2406.15704*, 2024.

[70] Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*, 2023.

[71] Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zujun Ma, and Chao Zhang. Enhancing multimodal llm for detailed and accurate video captioning using multi-round preference optimization. *arXiv preprint arXiv:2410.06682*, 2024.

[72] Zineng Tang, Ziyi Yang, Chenguang Zhu, Michael Zeng, and Mohit Bansal. Any-to-any generation via composable diffusion. *Proceedings of the NeurIPS*, 36:16083–16099, 2023.

[73] Zineng Tang, Ziyi Yang, Mahmoud Khademi, Yang Liu, Chenguang Zhu, and Mohit Bansal. Codi-2: In-context interleaved and interactive any-to-any generation. In *Proceedings of the CVPR*, pages 27425–27434, 2024.

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

[75] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[76] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

[77] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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

[79] Cong Wei, Quande Liu, Zixuan Ye, Qiulin Wang, Xintao Wang, Pengfei Wan, Kun Gai, and Wenhu Chen. Univideo: Unified understanding, generation, and editing for videos. *arXiv preprint arXiv:2510.08377*, 2025.

[80] 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 37th International Conference on Neural Information Processing Systems*, pages 79240–79259, 2023.

[81] 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.

[82] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. NExT-GPT: Any-to-any multimodal LLM. In *Proceedings of the ICML*, pages 53366–53397, 2024.

[83] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[84] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

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

[86] Zhifei Xie and Changqiao Wu. Mini-omni: Language models can hear, talk while thinking in streaming. *arXiv preprint arXiv:2408.16725*, 2024.

[87] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, et al. Qwen2. 5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

[88] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[89] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[90] Pinci Yang, Xin Wang, Xuguang Duan, Hong Chen, Runze Hou, Cong Jin, and Wenwu Zhu. Avqa: A dataset for audio-visual question answering on videos. In *Proceedings of the ACM MM*, pages 3480–3491, 2022.

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

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

[93] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

[94] Y Zhang, B Li, H Liu, Y Lee, L Gui, D Fu, J Feng, Z Liu, and C Li. Llava-next: A strong zero-shot video understanding model. *arXiv preprint arXiv:2311.10122*, 2024.

[95] Yiming Zhang, Yicheng Gu, Yanhong Zeng, Zhening Xing, Yuancheng Wang, Zhizheng Wu, and Kai Chen. Foleycrafter: Bring silent videos to life with lifelike and synchronized sounds. *arXiv preprint arXiv:2407.01494*, 2024.

[96] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[97] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.

In the **appendix**, we present discussion in limitation and societal impact (Appendix A,) details in model implementation and evaluation (Appendix B), construction details of JavisInst-Omni (Appendix C), discussion on architecture designs (Appendix D), supplementary investigations on JavisGPT (Appendix E), more quantitative case studies on JavisGPT (Appendix F), and overall discussion with related works (Sec. 2).

A Potential Limitation and Societal Impact

A.1 Contribution Statement

This work is the result of a large and comprehensive project that aims to provide the community with an open-source tool and platform for joint understanding and generation of sounding video, which is the first of its kind. While most of the individual network components are known in literature, it takes significant trials and errors to find a feasible plan, and connect and align them to form a state-of-the-art and yet concise system. In particular, the proposed three-stage training pipeline has proven useful for gradually improving model performance. In addition, the proposed `JavisInst-Omni` dataset represents a very rich mixture of single- and multi-modality data, with the majority being single-turn and the rest multi-turn. We believe the architecture, training methods, and datasets offer a very competitive baseline and have significant value for future data-centric and model-centric studies.

A.2 Limitation and Future Work

While our JavisGPT represents a pioneering effort in synchronized audio-video understanding and generation, the field itself is still in its early stages. In this section, we discuss the current limitations of our method within the context of its time and outline several directions for future work.

Architecture. JavisGPT adopts a straightforward encoder–LLM–decoder architecture. While effective, as briefly mentioned in Sec. 6, this design presents two core inconsistencies: (1) *Misaligned training objectives*. The comprehension tasks rely on next-token prediction (NTP) loss [5] for textual outputs, whereas the generation tasks use diffusion loss [40] (with alignment loss) for audio-visual outputs. These distinct objectives lead to inconsistent optimization signals for the shared LLM backbone. (2) *Asymmetric input–output modeling*. For multimodal content (in this case, audio and video), the input relies on VAE-based continuous embeddings [5], while the output is conditioned through a query-based interface [52]. The connection between the two is indirect: the generation queries can "see" the embeddings from the comprehension stage, allowing understanding to enhance generation; however, the reverse is not true, as the generation queries are not part of the comprehension input. To achieve a truly unified MLLM for audio-video understanding and generation, it may be beneficial to draw inspiration from recent advances in unified modeling for image-based multimodal tasks [78, 83, 76] and adapt them to the audio-video domain. Appendix D further elaborates on possible architectural directions for unifying comprehension and generation, as well as improving spatiotemporal synchronization, which we hope can provide valuable insights for future research in the community.

Scalability. For the sake of rapid development and verification, our current implementation is built upon a 7B-scale backbone LLM [5] and trained on a limited collection of publicly available datasets. As such, the scalability of our approach has not been fully explored. With access to larger computational resources, scaling the backbone LLM to 70B-size or beyond, along with training on trillions of multimodal tokens, may lead to significant performance improvements and unlock further potential in joint audio-video modeling.

Alignment. In the current post-training stage, JavisGPT is only instruction-tuned to acquire basic instruction-following and simple reasoning capabilities. To further enhance the model's generalization ability, reinforcement learning (RL) presents a promising direction for post-training. Specifically: (1) On the comprehension side, RL can transform the base JavisGPT into a thinking model [25, 75, 89] capable of handling complex reasoning tasks in real-world scenarios. (2) On the generation side, RL can substantially improve the quality, textual consistency, and audio-video synchrony of the generated content [59, 64]. We leave this direction as an important avenue for future exploration.

A.3 Societal Impact

JavisGPT enables synchronized audio-video comprehension and generation with language, supporting applications in education, accessibility, and interactive media. However, the same capabilities also raise important societal concerns. The model could be misused to synthesize misleading or harmful audio-visual content, contributing to the spread of misinformation or impersonation (e.g., deepfakes). If trained on sensitive or unfiltered data, it may also reproduce biased or private content, leading to fairness and privacy risks. Moreover, as generation models become more accessible, the barrier to misuse may decrease.

To mitigate these risks, responsible deployment practices are essential, including controlled access (e.g., through APIs), content watermarking, dataset auditing, and development of detection tools. While JavisGPT is released as a research prototype, we highlight the need for continuous evaluation of its broader impact as the field progresses.

B JavisGPT Model Details

B.1 Model Architecture

We propose JavisGPT, a general framework that transforms a vision-language MLLM backbone into a unified model capable of joint comprehension and generation of sounding videos. The backbone LLM is flexible and model-agnostic, allowing for various architectures and scales. In the manuscript, we adopt Qwen2.5-VL-7B-Instruct [5] as our primary backbone, and this section details the architecture configuration. Investigation on alternative base LLMs is also provided in Appendix E.2.

Backbone MLLM. Qwen2.5-VL-7B follows a decoder-only architecture, consisting of 28 Transformer blocks with a hidden size of 3584. During training, the original model parameters are kept frozen. We inject trainable LoRA [27] modules into each linear layer, with hyperparameters set to $r = 128$ and $\alpha = 256$. After training, the LoRA weights can be merged back into the LLM via reparameterization, incurring no additional overhead during inference.

Visual Encoder and Projector. We adopt the ViT and Merger modules from Qwen2.5-VL [5] as the visual encoder and projector, respectively, and keep their parameters frozen during training. The ViT has a hidden size of 1280 and a patch size of 14. After feature extraction, the Merger applies a 2×2 patch grouping and projects it into the LLM token embedding space with a dimension of 3584.

Audio Encoder and Projector. We adopt BEATs [10] as the audio encoder and keep it frozen during training. A learnable 2-layer MLP is used as the projector to map the 768-dimensional features produced by BEATs to the 3584-dimensional token embedding space of the LLM. For all newly introduced MLPs in this work, the intermediate size is set as $4\times$ of the output embedding size, and we omit this detail in the subsequent descriptions.

Audio-Video Synchronizer (SyncFusion). It consists of an MHSA layer with a hidden size of 3584 and 8 attention heads, with the FFN also implemented as a simple 2-layer MLP.

Audio-Video Generator and Projector. We use JavisDiT-v0.1 [40] as the DiT backbone, which consists of 28 layers of dual-stream DiT blocks with a hidden size of 1152. The text encoder is T5-XXL [60], with an embedding size of 4096 and a context length (*e.g.*, number of learnable semantic queries) of 300. The spatiotemporal prior encoder is ImageBind [24], with an embedding size of 1024 and a context length (*e.g.*, number of learnable spatiotemporal prior queries) of 77. Both the semantic projector and the spatiotemporal prior projector are 2-layer MLPs, mapping the 3584-dimensional LLM hidden states to 4096- and 1024-dimensional condition embeddings, respectively.

B.2 Training Configuration

Building upon the strong visual understanding and basic instruction-following capabilities of the Qwen2.5-VL [5] backbone, we adopt the three-stage training strategy described in Sec. 4 to develop JavisGPT for joint understanding and generation of synchronized sounding videos. Table A1 summarizes the key configurations, and the training and data details are provided below. All models are trained on 8 NVIDIA A100-80GB GPUs.

Table A1: **Detailed settings for progressive audio-video-synchronized training.**

Setting	Stage-I	Stage-II	Stage-III
Purpose	MM-PreTrain	AV-FineTune	MM-InstTune
Tasks	A-Und + AV-Gen-Pre	AV-Und + AV-Gen	A/V/AV-Und + AV-Gen
Trainable Module	$\phi^a + Q^{c,s}, \phi^{c,s}$	$\phi^a, \psi^{av} + Q^{c,s}, \phi^{c,s} + \Psi^{\text{LLM}}_{lora}$	$\phi^a, \psi^{av} + Q^{c,s}, \phi^{c,s} + \Psi^{\text{LLM}}_{lora}$
Trainable Params	174.36M	589.09M	651.50M
Training Objective	$\mathcal{L}_{ntp} + \mathcal{L}_{align}$	$\mathcal{L}_{ntp} + \mathcal{L}_{align} + \mathcal{L}_{diff}$	$\mathcal{L}_{ntp} + \mathcal{L}_{align} + \mathcal{L}_{diff}$
Training Samples	600K + 1.5M	450K + 450K	630K
Training Epochs	1	1	1
Warm-up Epochs	0.03	0.03	0.03
Batch Size	256	64	64
Learning Rate	1e-3	1e-4	1e-4
Weight Decay	0.0	0.0	0.0
GPU Days (A100)	6.5	14.2	9.3

Stage I: Multimodal Pretraining. This stage aims to equip the model with basic audio-video understanding and generation capabilities. (1) *Understanding*: We use 600K audio-text pairs to enhance the audio understanding ability. Tasks include audio captioning and question answering, with data sourced from AudioSet [23], AudioCaps [30], VGGSound [8], WavCaps [49], Clotho [19], ESC50 [55], GTZAN [67], MACS [48], and UrbanSound8K [62]. Training is conducted using next-token prediction loss. (2) *Generation*: We use 1.5M audio-video-caption triples from TAVGBench [47] to pretrain the generation component, using caption alignment loss $\mathcal{L}_{align}$ only. During this stage, the trainable components include the audio projector, query embeddings, and condition projector. We set the learning rate to 1e-3 and train for one epoch.

Stage II: Audio-Video FineTuning. This stage aims to enhance the model's ability to capture and maintain synchronization between audio and video in both understanding and generation tasks. (1) *Understanding*: We use 450K audio-video-text triplets from TAVGBench [47] and train on the sounding video captioning task using next-token prediction loss. (2) *Generation*: The same 450K triplets are reused for text-to-sounding-video generation, with a combination of caption loss and diffusion loss. In this stage, we additionally fine-tune the LoRA modules of the LLM, along with previously trainable components. The learning rate is set to 1e-4, and training runs for one epoch.

Stage III: Multimodal Instruction Tuning. This stage is designed to equip the model with instruction-following and basic reasoning capabilities. (1) *Understanding*: We first construct the JavisInst-Und subset, a synchrony-aware audio-video QA dataset spanning three levels: entity, relation, and global comprehension. It consists of 110K samples derived and processed from TAVGBench [47]. Details can be found in Appendix C. To further enhance performance, we also incorporate 135K audio-video understanding samples from VideoLLaMA2 [14], including training splits from AVQA [90], MusicAVQA [42], and AVSD [3]. To prevent catastrophic forgetting, we additionally include 25K image understanding samples from LLaVA-OneVision [33], 75K video understanding samples from LLaVA-Video-178K [96], 100K audio comprehension samples from Stage I's dataset, and 35K audio-video captioning samples from TAVGBench. (2) *Generation*: Since no existing AV-instruction dataset is available for generation tasks, we construct the JavisInst-Gen subset based on repurposed data from TAVGBench. It covers a diverse set of tasks including text-to-audio-video generation, in-context generation, and interleaved multi-turn conversation. More details are provided in Appendix C. All newly introduced modules are trained with a learning rate of 1e-4 for one epoch.

B.3 Evaluation Setup

In Sec. 5, we comprehensively evaluate the multimodal understanding and generation performance of JavisGPT, and here we introduce the detailed evaluation setups.

Multimodal comprehension. We follow the evaluation protocol of VideoLLaMA2 [14], and conduct unified benchmarking across visual understanding, audio understanding, and synchronized audio-video understanding for various multimodal large language models. The evaluation covers open-ended and multiple-choice formats, with single- or multi-turn dialogues. (1) *vi deo understanding* tasks are evaluated on three subsets: ActivityNet-QA [6], Perception Test [54], and MVBench [36]; (2)

audio understanding is evaluated on Clotho-AQA [39] and TUT2017 [50]; and (3) *audio-video understanding* includes AVQA [90], MusicAVQA [42], and AVSD [3]. All models are evaluated in the zero-shot setting using greedy decoding to ensure reproducibility. Following VideoLLaMA2 [14], we report the average accuracy as the primary evaluation metric. For multiple-choice datasets, accuracy is calculated via lexical match, while open-ended responses are evaluated with the assistance of GPT-3.5 [1]. To ensure fair comparison, input videos are uniformly sampled to 16 frames, and audio is preserved at 16 kHz across all models.

Multimodal generation. For audio-video generation evaluation, we follow the protocol of JavisDiT [40], using 1,000 text-to-audio-video samples from JavisBench-mini [40] for a comprehensive assessment. All models are evaluated with 4-second videos at 240P resolution and 24fps, and audio sampled at 16kHz.

A comprehensive evaluation measures of quality, consistency, and synchrony for audio-video generation results are provided. Here, we briefly introduce the mechanisms of each evaluation dimension:

- **Audio / Video Quality**: measuring the perceptual quality of the generated audio and video, including (1) *Fréchet Video Distance (FVD)*: $\mathrm{FVD} = \|\mu_r - \mu_g\|_2^2 + \mathrm{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$, where (μ_r, Σ_r) and (μ_g, Σ_g) are the mean and covariance of ground-truth and generated video features extracted by a pretrained I3D encoder [7]. Lower is better, indicating the generated video distribution is closer to the real one; (2) *Kernel Video Distance (KVD)*: similar to FVD, but estimates distribution differences via a kernel-based method (Kernel Inception Distance style), which is more stable on smaller datasets; lower is better; and (3) *Fréchet Audio Distance (FAD)*: same concept as FVD, but computed on audio features extracted by a pretrained AudioClip model [26], measuring distribution distance between generated and real audio; lower is better.
- **Text Consistency**: evaluating how well the generated audio and video semantically match the input text description, including (1) *ImageBind [24] text-video cosine similarity*: $\mathrm{sim}(t, v) = \frac{f_{\text{text}}(t)\cdot f_{\text{video}}(v)}{\|f_{\text{text}}(t)\|\cdot\|f_{\text{video}}(v)\|}$; (2) *ImageBind text-audio cosine similarity:* same process but with the audio encoder f_{audio}; (3) *CLIP-Score*: using CLIP [57] to compute semantic similarity between text and video (video frames are sampled, encoded, and averaged); and (4) *CLAP-Score*: using CLAP [84] to compute semantic similarity between text and audio.
- **Audio–Video Semantic Consistency**: measuring the semantic alignment between generated audio and generated video, including (1) *ImageBind audio-video cosine similarity*, encoding both modalities into the same space and computing cosine similarity between video and audio features; and (2) *Audio-Visual Harmony Score (AVHScore)*: introduced in TAVGBench [47] as a way to quantify how well the generated audio and video align semantically in a shared embedding space. It is defined by computing the cosine similarity between each video frame and the entire audio, then averaging across all frames: $\mathrm{AVHScore} = \frac{1}{N}\sum_{i=1}^{N}\cos\big(f_{\text{frame}}(v_i),\ f_{\text{audio}}(a)\big)$. A higher AVHScore indicates stronger audio–video semantic consistency. Note that we remove the CAVP-Score [44] used in JavisDiT [40] because this metric keeps a range from 0.798 to 0.801 and cannot capture the difference when evaluating semantic consistency.
- **Audio–Video Spatio-Temporal Synchrony**: evaluating spatiotemporal alignment in generated audio-video pairs, focusing on *JavisScore*: a new metric proposed in JavisDiT [40]. The core idea is to use a sliding window along the temporal axis to split the audio-video pair into short segments. For each segment, compute cross-modal similarity with ImageBind and take the mean score: $\mathrm{JavisScore} = \frac{1}{N}\sum_{i=1}^{N}\sigma(a_i, v_i), \quad \sigma(v_i, a_i) = \frac{1}{k}\sum_{j=1}^{k}\underset{\text{min}}{\text{top-}k}\{\cos\left(E_v(v_{i,j}), E_a(a_i)\right)\}$.

C JavisInst-Omni Dataset Details

C.1 Motivation

To advance instruction tuning for multimodal large language models in sounding video scenarios, we introduce two new datasets: JavisInst-Und and JavisInst-Gen, which support open-ended QA for a wide range of understanding and generation tasks, respectively. This is motivated by our observation that existing datasets suffer from significant limitations in terms of scale, capability coverage, and task diversity. Tab. A2 summarizes the comparison with existing mainstream datasets. Below, we highlight the key differences and contributions of our proposed datasets.

Table A2: **Comparison on sounding video datasets for instruction tuning.**

Dataset	Quantity	Diversity	AV-Synchrony	Multi-Turn	Reasoning
Comprehension					
MusicAVQA [42]	32K	✗	✗	✗	✗
AVQA [90]	40K	✓	✗	✗	✗
AVSD [3]	8K	✓	✗	✓	✗
JavisInst-Und (Ours)	**110K**	✓	✓	✓	✗
Generation					
MosIT [82]	5K	✗	✗	✓	✓
JavisInst-Gen (Ours)	**90K**	✓	✓	✓	✓

Audio-video comprehension. While AVQA [90], MusicAVQA [42], and AVSD [3] provide some diversity or dialogue support, they mainly fall short in modeling and evaluating audio-video synchrony (AV-Synchrony). Specifically, MusicAVQA [42] is limited to musical scenarios and lacks content diversity. AVQA [90] covers a wide range of everyday scenes, but the majority of its questions are simple existential queries such as "what" or "where", with little emphasis on event-level synchrony between audio and video, and it does not support multi-turn interactions. AVSD [90] partially addresses the multi-turn dialogue limitation, but still does not emphasize AV-synchrony, and its relatively small scale (8K samples) makes it unsuitable for large-scale training. To address these limitations, we introduce JavisInst-Und, which not only inherits the strengths of prior datasets but also emphasizes on synchrony-aware comprehension and expands the scale to 110K samples, providing a solid foundation for large-scale synchronized audio-video understanding.

Audio-video generation. In fact, there currently exists no instruction dataset specifically designed for synchronized audio-video generation. Although the MosIT dataset (see Tab. A2) includes multi-turn dialogues and reasoning capabilities, it does not contain direct training samples for generating audio-video content. To fill this gap, we introduce JavisInst-Gen, a dataset of 90K samples covering a wide range of everyday conversational scenarios. It supports synchronized AV generation with multi-turn interaction and reasoning capabilities, addressing the current limitations in the field.

JavisInst-Und and JavisInst-Gen are primarily constructed from TAVGBench[47] and InstV2V[13]. We leverage GPT-4o [1] and video-to-audio tools [95] for QA generation and data processing, and Fig. A3 showcases some representative examples for multi-turn comprehension and generation.

C.2 JavisInst-Und

Taxonomy. As detailed in Tab. A3, we construct a hierarchical taxonomy that captures information from local to global levels, covering 3 major categories and 10 subcategories: (1) *Entity-level*, including existence, alignment, grounding, and counting tasks, which aims to capture fine-grained auditory and visual characteristics of each individual sounding event. (2) *Relation-level*, includes spatial, temporal, and causal relationships in modeling the spatiotemporal and causal interactions between different sounding events. and (3) *Global-level*, including theme, emotion, and atmosphere analysis, aiming to capture the overall emotion and thematic expression of audio-video content.

Construction. We propose an efficient method for constructing JavisInst-Und by reusing audio-video samples from AudioSet [23] and leveraging the corresponding text captions provided in TAVGBench [47]. With the powerful language capabilities of GPT-4o [1], we generate diverse QA pairs for audio-video understanding.

• *Single-turn audio-video understanding*: For each of the 10 predefined dimensions, we design prompt templates (as exemplified in Fig. A1) and adopt a few-shot prompting strategy to guide GPT-4o in generating category-specific QA pairs based on the given audio-video caption. We randomly generate QA exercises from 3-4 categories for each sounding video caption.

• *Multi-turn audio-video understanding*: For videos from the same source in TAVGBench, we construct multi-turn dialogues in two ways: (1) randomly compositing multiple single-turn QA samples into a multi-turn session ("Composite"); and (2) first curating a single-turn instruction-based

Table A3: **Clarification of the category taxonomy of JavisInst-Und**.

Aspect	**Category**	**Description**	**Example** (explanation omitted)
Entity	Existence	Whether an object with a specific audio characteristic appears in the video.	*Que*: Is the cat meowing? *Ans*: No.
	Alignment	Whether the object making a certain sound and the object in the video are the same one.	*Que*: Does the lion in the video make the roaring sound in the audio? *Ans*: Yes.
	Grounding	The position of the object with a certain audio characteristic in the video.	*Que*: Where is the dog that is barking? *Ans*: Near a red car.
	Counting	The number of some objects in the audio and video.	*Que*: How many people can be identified in the video and audio? *Ans*: Two.
Relation	Spatial	The relative position of the different objects that make the sounds.	*Que*: Where is the barking dog in relation to the cat that is meowing? *Ans*: Over the cat.
	Temporal	The temporal relationship between the audio and video.	*Que*: Does the honking sound appear before the pedestrian crosses the street? *Ans*: Yes.
	Causal	The activity relationships between different objects.	*Que*: What causes the loud crash sound? *Ans*: The child dropping a glass.
Global	Emotion	The emotions expressed by the characters appearing in the video.	*Que*: What emotion is the woman expressing? *Ans*: Happy.
	Atmosphere	The common atmosphere in the video and audio.	*Que*: How about the atmosphere shown in the video and audio? *Ans*: Chaotic.
	Theme	Whether an object with a specific audio characteristic appears in the video.	*Que*: What is the overall theme of the video? *Ans*: Family picnic.
MultiTurn	Composite	2-3 rounds of conversations about the same sounding video.	*Que*: Where is the source of the machine gun firing sound? *Ans*: From the helicopter. *Que*: When does the sound of the machine gun firing occur? *Ans*: While the helicopter is flying.
	Gen2Und	Model first generates a sounding video, then answers a corresponding question.	*Que*: Generate a video that follows these guidelines: as the boat tow truck pulls ... *Ans*: Sure! *Que*: Does the boat being lifted out of the water produce any sound in the audio? *Ans*: No.

audio-video generation dialogue (see Appendix C.3), and then augmenting it with the corresponding single-turn QA samples above to create 2–3 turn "Gen2Und" conversations, further improving scene diversity.

In total, we construct a dataset of 110K QA samples in JavisInst-Und (see Fig. 5 for distribution), supporting a broad range of audio-video understanding tasks and applications.

C.3 JavisInst-Gen

Taxonomy. Table A4 provides a detailed overview of the three types of audio-video generation tasks included in our dataset, along with representative examples: (1) *Instruction-based Generation*. This is the core text-to-audio-video generation task. We distinguish between formal-style (more written and detailed) and colloquial-style (more casual and concise) instructions to better reflect real-world dialogue scenarios. (2) *Conditional Generation.* This task aims to generate new audio-video content based on modalities beyond text (*e.g.*, images, videos, audio, or any combination thereof) as context condition. It broadens application potential and includes tasks such as audio-video extension for long-form video generation and coarse-grained AV editing. (3) *Multi-turn Conversation.* This includes more complex task setups involving proactive reasoning and understand-then-generate (Und2Gen) scenarios, where the model must first comprehend prior context before generating. These tasks further enrich the diversity and complexity of the dataset.

Prompt for Spatial Relationship Question-Answering Data Synthesis

Instruction
Given a detailed auido-video joint description that summarizes the content of a audio and video, generate multi-choice question-answer exercises based on the description to help humans better understand the audio and video.
...

Category: `Spatial Relationship`
In this dimension, you need to focus on the relative position of the different objects that make the sounds. The question can be formatted as: (1) "Where is the <object1> that produced <sound1> in related to the <object2> that made <sound2>?", (2) "Is the <object1> behind <object2> that makes <sound1>?", (3) "Which <object> is the nearest to <object2> that makes <sound2>", etc. If there is NO exact description of the relative position of the sounding objects in the caption, output NO.

Examples
Caption: A lion roars while standing on a rock in the middle of a savanna.
{"Question": "Does the lion in the video make the roaring sound in the audio?", "Options": {"A": "Yes", "B": "No"}, "Answer": "A", "Explanation": "The audio aligns with the visual of the lion roaring in the savanna."}
Caption: The mountains are covered with lush forests and a variety of flowers, with the faint sound of flowing water in the distance.
{"Question": "Does the object producing the water flowing sound appear in the video?", "Options": {"A": "Yes", "B": "No"}, "Answer": "B", "Explanation": "The video only shows a mountain with forests and flowers, not the source of the water flowing sound."}
Caption: The man is shown fishing in a river, holding a fishing rod in his hand while a fish can be seen swimming in the water.
{"Question": "NO", "Answer": "NO"}

Target
Now generate a question-answer sample according to the given caption:
Caption: `{caption}`

Figure A1: **Prompt for spatial relationship question-answering data synthesis.** Here `{caption}` is the placeholder for a given audio-video caption.

Prompt for Proactive Audio-Video Generation Data Synthesis

Please extract an unimportant entity or concept (such as an object, place, or activity) from the given text and perform the following tasks:
1. Remove the extracted concept from the text while keeping the original meaning intact.
2. Generate a sentence expressing human preference
3. Generate a natural response to the preference statement.
Output Format: Please output the result in JSON format with the following keys:
- 'modified_text': The text after removing the extracted concept.
- 'generated_preference': The sentence expressing human preference.
- 'response': The response to that sentence.
The given text is: `{caption}`. Directly output the rewritten content without any additional text.

Figure A2: **Prompt for proactive audio-video generation data synthesis.**

Construction. JavisInst-Gen is also efficiently constructed by organizing data around the three defined types of audio-video generation tasks:

• *Instruction-based Generation*: This is the core task, where the model is required to generate synchronized audio-video content conditioned on a given text caption. To support this, we leverage

Table A4: **Clarification of the category taxonomy of JavisInst-Gen**.

Task	Category	Description	**Example** (response omitted)
Instruct	Formal	Sounding video generation with formal instruction.	***Que***: Compose a thorough video piece featuring narration and imagery. The sound of a motorcycle engine ...
	Colloquial	Sounding video generation with colloquial instruction.	***Que***: Make this into a video right away. A female vocalist is performing ...
Condition	V2A	Generate an audio for input video.	***Que***: Mount an audio track to the visuals. The intense and fast-paced ...
	A2V	Generate a video for input audio.	***Que***: Produce a video narrative: a robotic vacuum cleaner is parked ...
	AI2V	Animate an image with input audio.	***Que***: Make a video with the picture and soundtrack. A motorcycle revving ...
	I2AV	Animate an image and make an audio.	***Que***: Compose a video from this image, with sound. As the toy police car sits ...
	AV-Ext	Temporally extend the sounding video.	***Que***: Continue from the stopping point. A man wearing a pink t-shirt ...
	AV-Edit	Edit the sounding video as supposed.	***Que***: Make the concept art look like ancient Egyptian hieroglyphs.
MultiTurn	Proactive	Model first extracts user preference, then generates a corresponding sounding video.	***Que***: What do you think about oranges as a fruit? ***Ans***: They're really invigorating and full of flavor! ***Que***: Create a video where a car driving ... with a fruit balanced on top of it.
	Rethink	Model first supplements a detailed request, then generates a corresponding sounding video.	***Que***: Create a video where a person crumples paper and ... ***Ans***: Do you want the sound of paper being crumpled up can be heard as ...? ***Que***: Yes!
	Und2Gen	Model first answers an audio-video-related question, and the generate/extend a corresponding sounding video.	***Que***: What emotion is the crowd expressing? ***Ans***: Surprise and a mixture of applause. ***Que***: Show me a video of a crowd with silence and sacredness.

GPT-4o [1] to construct 3,000 diversified prompt templates, and randomly sample 45K audio-video captions from TAVGBench [47] to combine with these templates, forming a wide range of instruction-based generation instances. To enhance linguistic diversity, approximately 20% of the instructions are further paraphrased by GPT-4o-mini (for cost efficiency), making them more aligned with natural, conversational instructions typically found in real-world applications.

• *Conditional Generation*: This task follows a similar construction strategy to instruction-based generation, where we synthesize a large number of conditional generation cases by randomly combining prompt templates with existing audio-video captions, followed by modality-specific input processing. For example, in A2V tasks, the audio is extracted as input; and in I2AV, the first frame of the video is used as input. For AV-Extension, we sample two overlapping clips (*e.g.*, 1–2s overlap) from a $\tilde{1}$0s video in TAVGBench, using the first as input and the second as the generation target. As for AV-Edit, since TAVGBench (and other related datasets) lack suitable editing instructions, we adopt an alternative approach: we randomly sample 5K video editing pairs from InstV2V[13], and apply FoleyCrafter[95] to generate corresponding audios for both the pre-edit and post-edit video clips, thus enriching training scenarios for AV-based editing tasks.

• *Multi-turn Conversation*: We carefully design three representative and practical forms of multi-turn generation: proactive generation, rethink generation, and understanding-to-generation (Und2Gen). In proactive generation, the model is expected to identify user preferences from prior dialogue and incorporate them into the generated sounding video. In rethink, the model receives an ambiguous initial instruction and proactively suggests a more complete and specific generation prompt.

To construct such data, we reuse sounding videos from TAVGBench as generation targets and employ GPT-4o to synthesize dialogues based on their corresponding captions. Specifically: (1) for proactive scenarios, we remove one key entity or concept from the caption and create a preceding dialogue

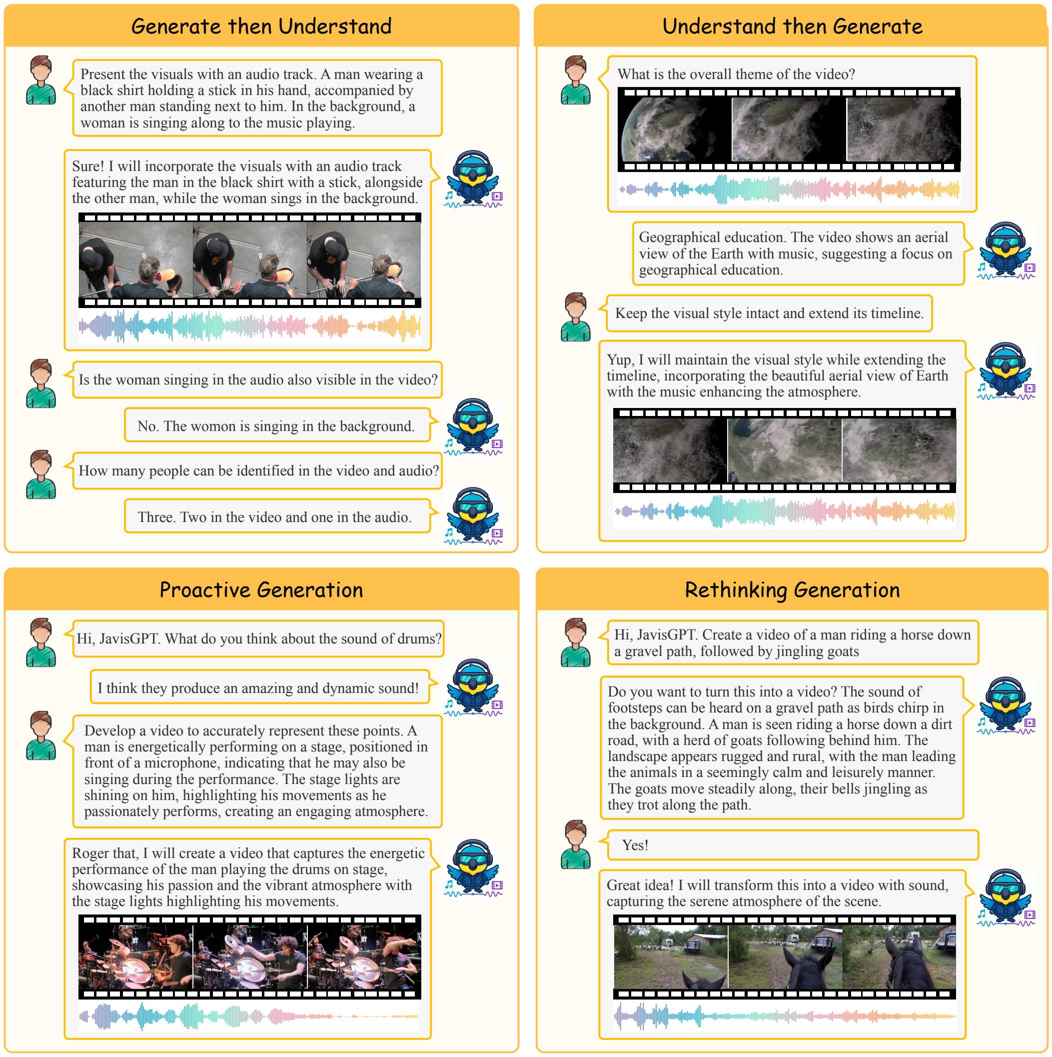


Figure A3: **Examples of multi-turn comprehension and generation from JavisInst-Omni.**

turn that embeds a user preference, simulating preference-aware generation (see prompt template in Fig. A2). (2) for rethink, we generate a highly abstract version of the caption as the user's vague instruction. Then, the original detailed caption is reformatted using instruction generation templates to serve as the model's suggested clarification.

As for Und2Gen, the model first receives a sounding video and engages in one round of understanding dialogue (from Appendix C.2), followed by a generation task such as producing a new audio-video clip or performing AV-extension based on prior understanding. Inspired by NExT-GPT [82], we use BERTScore [93] to identify semantically similar caption pairs in TAVGBench, and instruct GPT-4o to construct conditional generation prompts accordingly.

In total, JavisInst-Gen contains 90K high-quality QA samples (see detailed statistics in Fig. 5), supporting a wide range of real-world generation scenarios.

D Architecture of Joint Comprehension and Generation of Sounding Videos

D.1 Design Choices of Generative MLLM

Designing a unified architecture for both comprehension and generation tasks remains a central challenge. As there are no previous exploration in JAV domain (we are the first one), we mainly summarize and compare three representative paradigms in Fig. A4 under the text-image context:

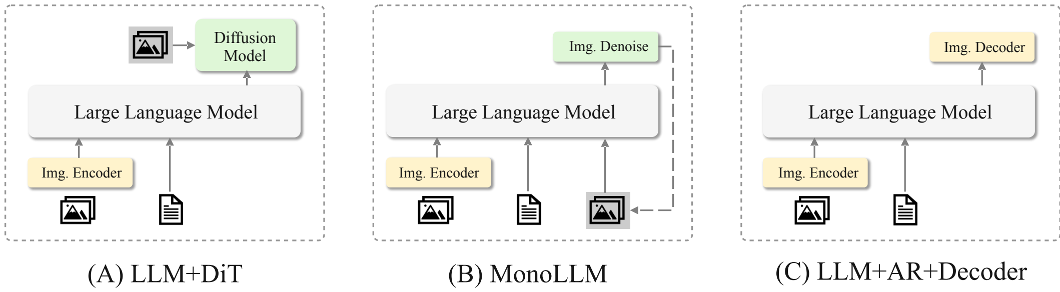


Figure A4: **Choices of unified architecture for joint understanding and generation. (A)** Our JavisGPT adopts the widely verified LLM+DiT architecture [82, 52] for efficiency and efficacy. **(B)** MonoLLMs [97, 85] that integrate textual auto-regression and multimodal diffusion into a single LLM, causing severe optimization difficulty. **(C)** The LLM+AR+Decoder paradigm [78, 83] brings extra training efforts on encoder-decoder modules for JAVG and high inference cost to predict numerous multimodal tokens in the auto-regression behavior.

LLM+DiT. Our JavisGPT adopts a modular yet effective LLM + DiT architecture, which has been widely verified in recent multimodal works [82, 52] for its balance between efficiency and generation quality. This architecture separates semantic understanding (handled by the LLM) from generative decoding (handled by a diffusion transformer), allowing task specialization and easier optimization. Moreover, the use of learnable queries and a conditioned DiT enables flexible generation from various modalities with strong temporal coherence.

MonoLLMs. Recent works [97, 85, 46] attempt to unify textual autoregression and multimodal diffusion within a single large language model. However, these MonoLLMs suffer from severe optimization difficulties due to conflicting training objectives and heterogeneous output formats [9]. The lack of architectural decoupling often leads to poor convergence and unstable generation, especially for high-dimensional outputs like audio or video.

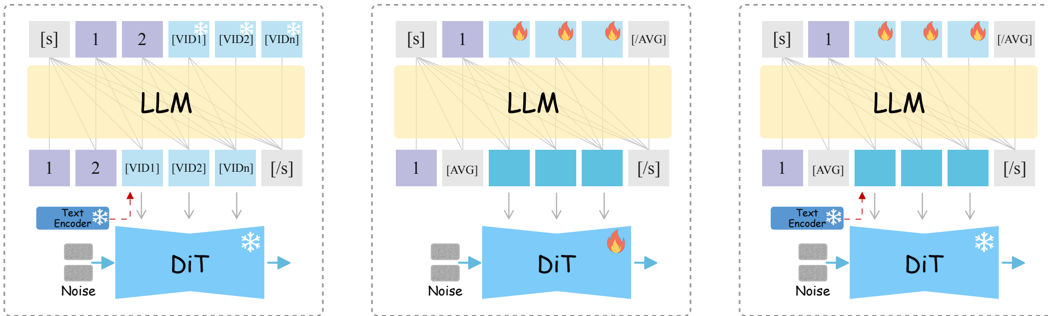
LLM+AR+Decoder. The autoregressive encoder-decoder paradigm [78, 83, 76] introduces additional training complexity, which requires maintaining a heavy encoder-decoder pipeline to bridge multimodal embeddings with autoregressive prediction on discrete [83] or continuous [76] token embeddings and trying to avoid information loss. Besides, this design also incurs significant inference costs, as it must sequentially predict large volumes of multimodal tokens. For instance, when using 512 tokens per frame, generating a one-minute video clip (even without audio) at 24 fps would require generating $512 \times 24 \times 60 = 737{,}280$ tokens, making it less practical for real-time or high-resolution generation.

In conclusion, the LLM+DiT paradigm is the best option for current JAVG field, which has also been verified by our comprehensive experiments in Sec. 5. However, the LLM+AR+Decoder architecture may show potentials to further bridge the unified task of comprehension and generation. The key insight is simple and straightforward: after unifying understanding and generation AV representations into the same space, the generation embeddings can be also "seen" by comprehension tasks. As discussed in Appendix A, this feature can mutually strengthen both comprehension and generation capabilities after scaling up model capacity and data quantity. We hope this can bring more inspiration in the community.

D.2 Design Choices of LLM-DiT Combination

In Fig. A5, we also compare several designs of LLM+DiT combination, which further highlights the advance of our JavisGPT's architecture in performance, diversity, and training stability.

NExT-GPT. NExT-GPT [82] formulates generation as a classification task over a fixed vocabulary of video IDs (*e.g.*, <VID1>, <VID2>, $\cdots$, <VIDn>), forcing all instruction prompts to regress to this small set of special tokens. This rigid formulation limits the expressiveness of conditional instructions and severely constrains the diversity and fidelity of generated audio-video outputs, which is consistent to NExT-GPT's poor generation performance in Sec. 5.1. Additionally, it restricts generalization to novel or fine-grained instructions outside the pre-defined token set.



(A) NExT-GPT (B) MetaQuery (C) JavisGPT

Figure A5: **Choices for MLLM + DiT combination. (A)** NExT-GPT [82] suffers from degraded generation quality and diversity due to its forced regression of all generation instructions to a fixed set of special tokens (*i.e.*, <VID1>, <VID2>, $\cdots$, <VIDn>). **(B)** MetaQuery [52] removes the support of a text encoder, which leads to training instability. **(C)** Our JavisGPT combines learnable query embeddings with a supportive text encoder, enabling stable training and the generation of high-quality and diverse audio-video content.

MetaQuery. MetaQuery [52] introduces a learnable query mechanism but removes the auxiliary support of the text encoder of DiT, relying solely on query tokens to interface with the generation model. This approach often leads to training instability, especially when scaling to complex multimodal data like audio-video. The absence of semantically grounded text features weakens the model's ability to align instructions with generated content.

JavisGPT. Our JavisGPT integrates the advantages of both sides by combining learnable query embeddings with the semantic guidance of a pretrained text encoder. The text encoder offers strong grounding for conditional semantics, while the queries serve as adaptive, trainable intermediaries to control the generative diffusion model. This hybrid design results in stable training dynamics and supports the generation of high-quality, diverse, and instruction-aligned sounding videos. Moreover, it enables better generalization to a wide range of instructions, including free-form, multi-turn, or mm-conditional prompts.

E Supplementary Investigation on JavisGPT

E.1 Details for Human Evaluation on Interleaved Conversation

In Sec. 5.1, we present a comparative evaluation with UnifiedIO-2 [43] and NExT-GPT [82] in interleaved comprehension and generation scenarios. Since no existing benchmark is available for this setting, we conduct a human evaluation, as detailed below:

Data Collection. We use GPT-4o to construct 100 multi-turn dialogue samples based on the four scenarios illustrated in Fig. A3, including Gen2Und, Und2Gen, Proactive, and Rethink (25 samples each).

Evaluation Criteria. Three unbiased human annotators independently rated the model outputs on a scale of 0–5 across five dimensions:

1. *Instruction Following*: whether the model follows the user instruction to generate a coherent textual response and a matching audio-video output;
2. *Question Answering*: whether the model accurately understands the input audio-video content and provides correct answers;
3. *Generation Quality*: the overall quality, coherence, and synchrony of the generated audio-video content;
4. *Context Reasoning*: whether the model can integrate user preferences implied in the dialogue history into its response or generation;
5. *Proactive Thinking*: whether the model can recognize vague or under-specified instructions and proactively ask clarifying questions or complete the intent.

Result Analysis. The evaluation results are shown in Fig. 6, from which we draw several key conclusions:

- NExT-GPT shows poor instruction-following ability and often fails to generate audio-video outputs—only 33 out of 100 cases succeeded. This can be attributed to insufficient fine-tuning (only 5K samples used).
- Both NExT-GPT and UnifiedIO-2 suffer from low generation quality. As discussed in Appendix D.2, NExT-GPT's reliance on fixed special tokens for all instructions limits both generation diversity and fidelity. UnifiedIO-2, lacking a dedicated DiT-based generator, resorts to generating keyframes and stitching them into video, resulting in subpar temporal coherence and realism.
- JavisGPT significantly outperforms both baselines in question-answering, benefiting from the scale and diversity of the JavisInst-Omni dataset.
- Due to the lack of training on preference reasoning and proactive dialogue, both NExT-GPT and UnifiedIO-2 exhibit weak context reasoning and proactive generation, whereas JavisGPT consistently shows stronger contextual understanding and responsiveness, underscoring its practical utility and superior alignment with real-world conversational needs.

The results further highlight the strong capability of our JavisGPT.

E.2 Investigation on Backbone LLM

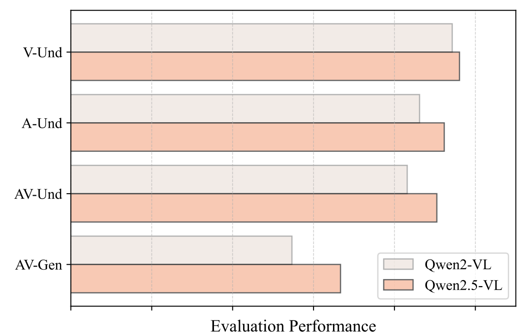


Figure A6: **Ablation on Backbone LLMs**.

In the manuscript, we mainly take the Qwen2.5-VL [5] as the backbone LLM to build our JavisGPT. Here we provide a further investigation on the base LLM with a slightly weaker Qwen2-VL [77] model. After the same training data and procedure, we compare the performance variation in Fig. A6 for further analysis. Accordingly, upgrading the language backbone from Qwen2-VL to Qwen2.5-VL consistently improves performance across all evaluation tracks, including unimodal understanding (V-Und, A-Und), audio-visual comprehension (AV-Und), and audio-visual generation (AV-Gen). Notably, the gain is most pronounced in the generation task (AV-Gen), where Qwen2.5-VL outperforms Qwen2-VL by a substantial margin. These results highlight that a more capable language foundation model can significantly benefit not only text-based reasoning but also cross-modal alignment and response generation. The consistent improvements across both understanding and generation tasks validate the critical role of the LLM backbone in scaling multimodal capabilities of our JavisGPT.

E.3 Ablation on the ST-Prior Query

In this paper, to enhance the synchrony between generated audio and video, we follow the design of JavisDiT [40] and introduce an additional set of learnable spatiotemporal queries to bridge the backbone LLM and the downstream JAV-DiT. Here, we explore the effect of removing the ST-Prior queries on corresponding performance changes in sounding video generation.

Table A5: **Ablation on the effectiveness of ST-Prior queries**.

Strategy	Quality		Consistency			Synchrony
	FVD↓	FAD↓	TV-IB↑	TA-IB↑	AV-IB↑	JavisScore↑
w/o ST-Prior Query	319.3	**7.6**	**0.146**	0.178	**0.203**	0.150
w/ ST-Prior Query	**317.5**	**7.6**	0.145	**0.180**	0.202	**0.157**

According to Tab. A5, removing the temporal-spatial prior query has minimal impact on overall generation quality and consistency, but leads to a slight drop in audio-visual synchrony. This result aligns with our motivation: the temporal-spatial prior primarily serves to enhance the temporal and spatial alignment of generated audio and video content.

Figure A7: **More qualitative results for joint understanding and generation.**

F Qualitative Case Studies of JavisGPT

More understanding and generation cases can be found in Fig. A7.

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: The main claims in the abstract and introduction accurately reflect the contributions and scope of the paper. We ensure that the stated contributions are directly supported by our method, experiments, and analysis presented in the main text.

   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: The limitations of the work are comprehensively discussed in Appendix A.

   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: [NA]

Justification: This paper does not introduce new assumptions or proofs.

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: The paper provides all necessary implementation details required to reproduce the main experimental results, including dataset descriptions, model architectures, training configurations, and evaluation protocols in Sec. 5 and Appendix B. Additionally, we include an anonymous code repository for reproducibility: `https://anonymous.4open.science/r/JavisGPT`.

   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: We provide open access to both the code and data via an anonymous repository at `https://anonymous.4open.science/r/JavisGPT`.

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, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides all necessary implementation details required to reproduce the main experimental results, including dataset descriptions, model architectures, training configurations, and evaluation protocols in Sec. 5 and Appendix B. Additionally, we include an anonymous code repository for reproducibility: `https://anonymous.4open.science/r/JavisGPT`.

   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: We follow previous works to fix the random seed as 42 for fair comparison and reproducibility. The results in Sec. 5 show a large improvement against baseline models, emphasizing the statistical significance.

   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: We report the type of compute resources used for all experiments in Sec. 5 and Appendix B to facilitate reproducibility and resource estimation.

   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: We have carefully study the NeurIPS Code of Ethics and conforms to it.

   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: [Yes]

    Justification: We have discussed the positive and negative societal impact of the proposed work in Appendix A.

    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: [Yes]

Justification: We recognize the potential risks associated with releasing model and data and discuss in Appendix A.

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: [Yes]

Justification: We conforms to the terms of service and license policy of existing datasets in their official websites.

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: [Yes]

Justification: We introduces new assets, including a benchmark, pretrained models, evaluation metrics, which are accompanied by comprehensive documentation at `https://anonymous.4open.science/r/JavisGPT`.

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. We only perform human evaluation on three volunteers.

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. We only perform human evaluation on three volunteers.

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.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: LLMs were used only for language editing and formatting assistance.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.