VidUEU-Agent: A Data-Curation Agent for Multimodal Understanding, Editing, and Unified Tasks

Tengju Ru[1,*] **Weitong Lian**[1,*] **Zecong Tang**[2,*,‡] **Lingyi Meng**[1,*]
Haoran Li[2] Zhejun Cui[2] Yichen Zhu[1] Hangshuo Cao[1]
Qi Kang[1] Yechi Liu[3] Kaixuan Wang[2] Yu-Jie Yuan[4]
Chunwei Wang[4] Yu Zhang[1,†] Bo Dai[2,†]

[1] Zhejiang University [2] The University of Hong Kong
[3] Institute of Automation, Chinese Academy of Sciences [4] Huawei Technologies Ltd.

zhangyu80@zju.edu.cn bdai@hku.hk

* Equal contribution. ‡ Project lead. † Corresponding authors.

Abstract

The advancement of multimodal learning is heavily constrained by the scarcity and high annotation cost of high-quality instruction data. While raw videos offer an abundant and dynamic source of visual knowledge, transforming them into precise instruction data remains highly challenging. To bridge this gap, we present VidUEU-Agent, an automated pipeline that synthesizes instruction data from raw videos for multimodal understanding, editing, and unified tasks. VidUEU-Agent follows a four-stage pipeline: it performs global perception to filter low-quality shots, mines keyframes and multimodal signals, constructs metadata and routes the metadata to synthesize instruction samples. More importantly, our agent framework can autonomously iterate the data distribution based on downstream training results, thereby optimizing data quality, balance, and training effectiveness. Scaling this pipeline, we construct VidUEU-100K, a large-scale dataset of 100K training samples spanning multiple domains and tasks, providing a robust foundation for developing comprehensive multimodal capabilities. Extensive experiments demonstrate that fine-tuning baseline models on VidUEU-100K yields significant and consistent performance gains across diverse representative benchmarks, validating the effectiveness of our proposed agent framework and dataset.

1 Introduction

With the rapid advancement of large language models (LLMs), multimodal large language models (MLLMs) have made significant strides Yin et al. [2024]. These models use vision encoders to bridge multimodal inputs, enabling joint understanding and reasoning Wu et al. [2023], Liang et al. [2024]. Concurrently, multimodal generative models have also evolved rapidly, expanding capabilities from high-fidelity text-to-image synthesis to controllable text-to-video generationZhang et al. [2023], Chen et al. [2025a], Lipman et al. [2022]. However, generative models and MLLMs have progressed along separate tracks, making it challenging to unify understanding and generation in one system Zhao et al. [2025]. Unified multimodal models (UMMs) provide a consistent framework for multimodal understanding and generation, showing great potential for multi-step reasoning and controllable image editing Yang et al. [2025], Chen et al. [2025b], Wu et al. [2025], Deng et al. [2025a].

The advancement of multimodal capabilities relies on the availability of high-quality datasets Yang et al. [2025]. For UMMs, there remains a lack of a scalable and reproducible paradigm for producing such high-quality instruction data while minimizing manual annotation cost Yin et al. [2024], Deng

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

et al. [2024]. Raw videos provide a scalable source of multimodal knowledge, covering broad domains and temporally ordered interactions beyond static images Tong et al. [2025a]; however, they are also redundant, noisy, and unstructured, and do not directly provide task-oriented instruction data. The key challenge is therefore to convert large-scale raw videos into instruction data that jointly supports understanding and generation while remaining suitable for stable model training.

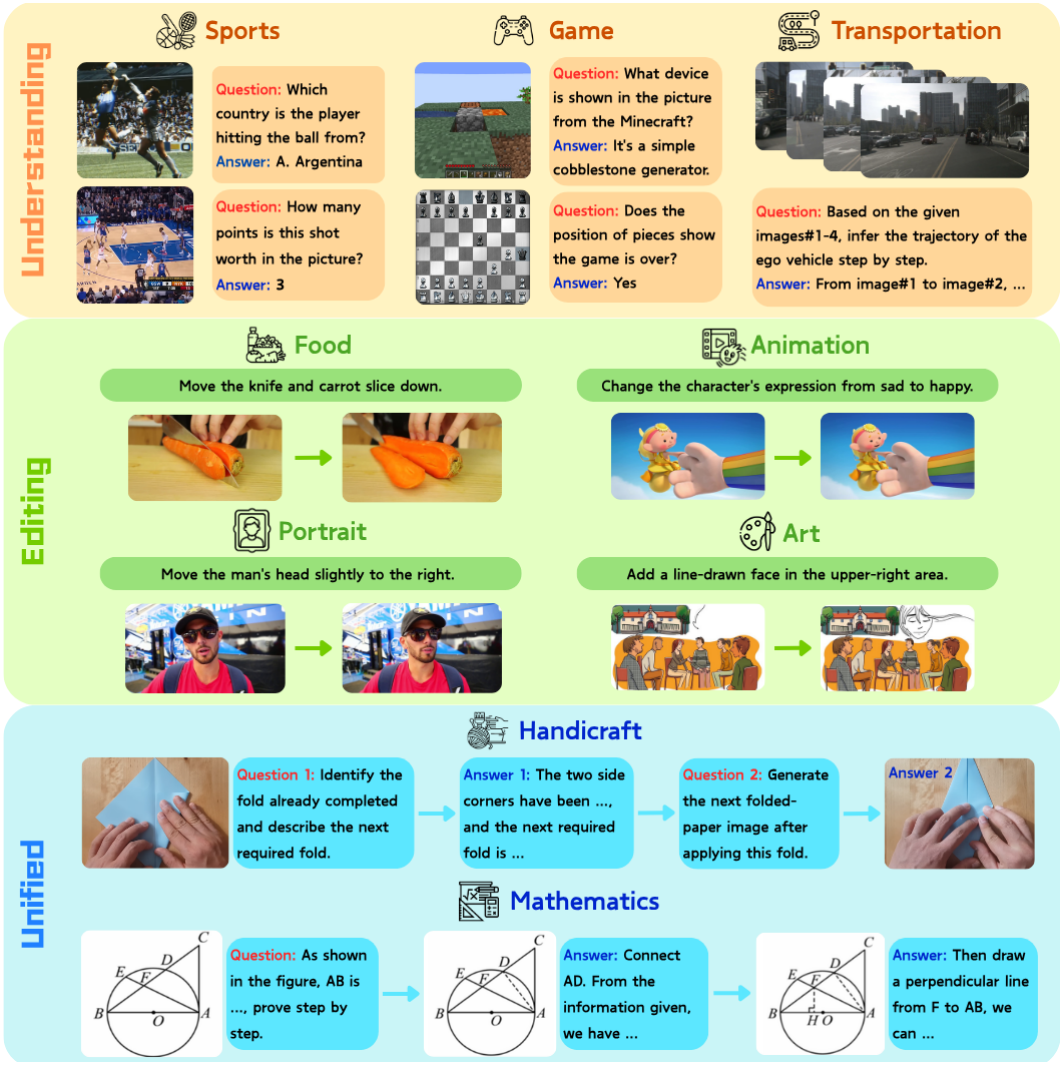


Figure 1: Example samples of VidUEU-100K. We illustrate three capability paradigms: Understanding (top), Editing (middle), and Unified (bottom).

In recent years, LLM-based agent systems have demonstrated strong planning and tool-use capabilities, enabling them to decompose complex goals into executable steps and to achieve large-scale data generation, verification, and filtering Guo et al. [2024], Xu et al. [2025], Kumar et al. [2024], Zhang et al. [2025a]. By interacting with external tools and making feedback-driven decisions, agents can automate complex data-curation workflows Tang et al. [2025], Riaz et al. [2025]. These properties make agent systems a fit for transforming raw videos into structured instruction data.

Motivated by the strengths of agent systems and video data, we introduce VidUEU-Agent, an automated data synthesis framework that leverages agent planning and tool use to convert raw videos into multimodal instruction data. VidUEU-Agent follows a four-stage pipeline: it performs global perception to obtain coarse video understanding and filter low-quality shots, mines keyframes and multimodal signals through frame filtering and tool use, constructs metadata from the keyframe sequence and filtered signals, and routes the metadata to synthesize instruction samples for understanding,

editing, and unified tasks. VidUEU-Agent further autonomously iterates the data distribution based on downstream training results, improving data quality, balance, and training effectiveness.

Further, we scale this pipeline to construct VidUEU-100K, a large-scale high-quality multimodal instruction dataset built from 8,268 diverse videos collected from the web and open datasets. The dataset contains nearly 20k dense metadata annotations, enabling 40K understanding samples, 30K editing samples, and 30K unified samples, and it spans nine task paradigms and five scenario categories, ensuring comprehensive and balanced data for improving the overall capabilities of UMMs. We further fine-tune a series of baseline models on VidUEU-100K and validate the effectiveness of our data through multiple benchmarks and case studies.

In summary, our contributions are as follows:

- We propose VidUEU-Agent, a scalable and automated framework that leverages the planning and tool-using capabilities of agents to synthesize high-quality instruction data for multimodal understanding, editing, and unified tasks from video sources.
- By scaling this pipeline, we construct VidUEU-100K, a large-scale dataset featuring 20k dense metadata annotations and 100K diverse instruction samples, covering diverse domains and complex tasks across understanding, editing, and unified capabilities.
- We fine-tune a series of baseline models on VidUEU-100K, where extensive experiments across multiple benchmarks demonstrate the effectiveness of our data in enhancing model performance in various multimodal tasks.

2 Related Work

2.1 Data Construction based on Agent Systems

Early methods often used powerful LLMs to directly generate instruction–response pair Xu et al. [2023], Liu et al. [2023], Zhao et al. [2024]. Despite their efficiency, this approach lacks grounding in verifiable external signals, introducing uncertainty in data reliability and controllability. Ji et al. [2023], Rawte et al. [2023], Wang et al. [2023]. Recent trends shift toward vision agents and tool use. Gao et al. [2025], Deng et al. [2025b]. Such systems equip LLMs with perception tools for performing complex visual reasoning and information extraction to improve reliability and interpretability Reddy et al. [2025], Deng et al. [2025b]. With this tool-use capability, agent-based data synthesis is becoming a promising approach for automated construction of multimodal datasets Sun et al. [2025]. Prior work has explored using model agents to automatically collect bounding boxes, rewrite image captions, or synthesize question–answer pairs, greatly reducing reliance on manual annotation Chen et al. [2023], Joshi et al. [2025]. LLaVA-Video Zhang et al. [2025b] further synthesizes video-understanding QA from sampled frames, but remains focused on video understanding tasks. However, existing methods struggle to jointly synthesize understanding, editing, and unified instruction data for UMMs, which hinders the development of these models across broader tasks Zhao et al. [2025].

2.2 Multimodal Understanding and Generation Datasets

Large-scale, high-quality instruction datasets have become one of the key drivers of recent progress in multimodal models Li et al. [2023], Chen et al. [2023], Liu et al. [2024a]. In multimodal understanding, existing work enhances model capabilities in visual question answering and fine-grained recognition Liu et al. [2023], Han et al. [2024], Yue et al. [2024], Chen et al. [2024a], Liu et al. [2024c]. In multimodal generation, existing work advances high-fidelity generation and controllable editing Zhao et al. [2023], Podell et al. [2023], Cao et al. [2025], Yu et al. [2023], Li et al. [2024]. Recently, the multimodal field has shifted toward unified multimodal models Yang et al. [2025]. Accordingly, unified architectures demand stable generalization across understanding and generation capabilities Tong et al. [2025b], Zhuang et al. [2025]. However, current image-text datasets are primarily sourced from static images, inherently lacking real-world dynamics Chen et al. [2025c], Shen et al. [2026]. In contrast, videos, featuring inherent temporal dynamics and causal interactions, serve as a more promising medium for high-quality instruction data Tong et al. [2025a].

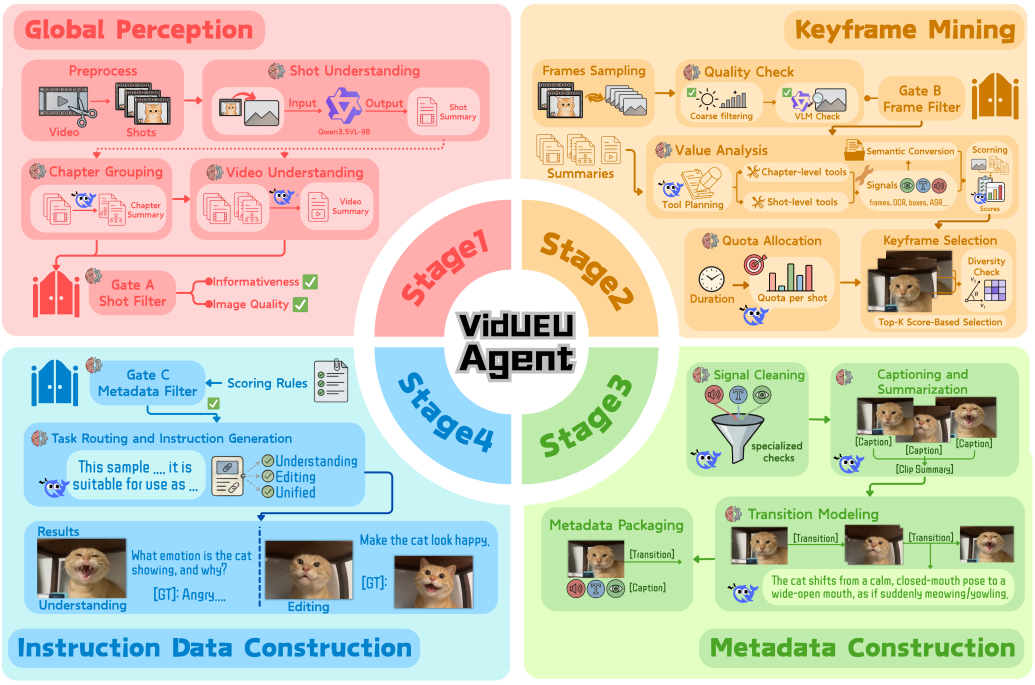


Figure 2: Overview of VidUEU-Agent. The agent performs **global perception** with quality gating, then conducts **keyframe mining** using multimodal signals and tool verification, builds **metadata**, and finally synthesizes **instruction data** for understanding, editing, and unified tasks.

3 VidUEU: Agent & Dataset

3.1 VidUEU-Agent

We propose **VidUEU-Agent**, a data synthesis agent for raw videos, designed to curate instruction data for multimodal understanding, editing, and unified tasks. VidUEU-Agent first parses raw videos through global perception, removes low-quality shots, and then constructs metadata from selected keyframes and filtered multimodal signals. It then maps each metadata entry to suitable instruction data formats, ensures data quality through multi-stage quality screening and consistency filtering, and autonomously iterates the data distribution using downstream training feedback.

3.1.1 Stage 1: Global Perception

As a preprocessing step, VidUEU-Agent first uses PySceneDetect Castellano to detect shot boundaries and split each long video into a sequence of shot-level clips. It then performs global perception at three levels. At the shot level, VidUEU-Agent samples the middle frame of each shot and prompts a VLM to generate a shot summary. At the chapter level, the agent reviews all shot summaries of the video, groups adjacent shots according to their scene content, maps them into several chapters, and generates a chapter summary for each chapter. At the video level, the agent summarizes the whole video based on all shot-level and chapter-level summaries to obtain a video summary. Finally, VidUEU-Agent applies Gate A, namely the shot filter, to examine each shot according to informativeness and image quality rules, and removes low-quality shots from the subsequent stages. The prompts for three-level summarization, as well as chapter construction, are provided in Appendix A.1.

3.1.2 Stage 2: Keyframe Mining

We perform keyframe mining on the shots that pass Gate A. **Frame Sampling.** For each retained shot, VidUEU-Agent samples frames at a fixed interval and applies Gate B frame filtering, which combines tool-based quality statistics with VLM-based check to remove low-quality frames. For each frame that passes Gate B, VidUEU-Agent generates a caption. **Tool Planning.** Based on the

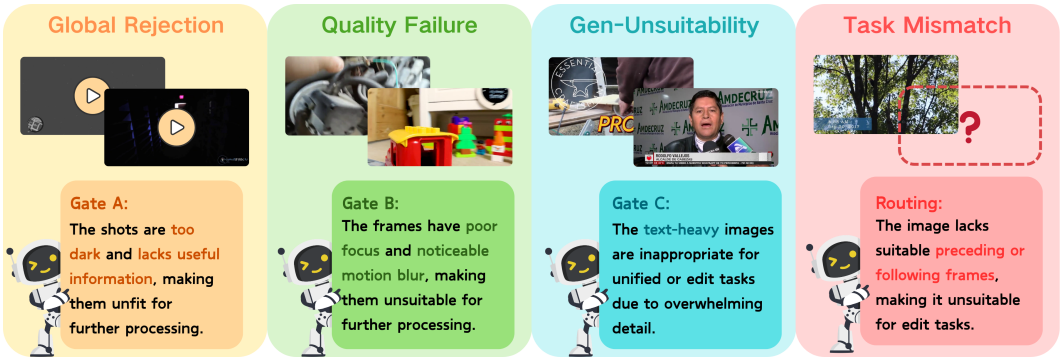


Figure 3: Failure Case Examples: Illustrating four common rejected cases in the processing pipeline: Global Rejection, Quality Failure, Gen-Unsuitability, and Task Mismatch.

retained frames and the summaries from global perception, VidUEU-Agent performs hierarchical tool planning by assigning tools at both the chapter level and the shot level to collect targeted information. Following this plan, VidUEU-Agent applies the selected tools to every retained frame in the corresponding shot. **Frame Scoring and Keyframe Selection.** VidUEU-Agent verbalizes the tool signals and scores each frame using these signals together with the three-level summaries. Based on the video duration, VidUEU-Agent sets the keyframe budget and allocates it across shots using the summaries. For each shot, VidUEU-Agent keeps the top-scoring candidates within the assigned quota, computes their image embeddings, measures cosine similarities, and applies weighted facility-location greedy selection to choose high-quality, semantically representative keyframes. The prompts for tool planning, signal verbalization, frame scoring, and quota allocation are provided in Appendix A.2. The tool pool, verbalization rules, and scoring rules are described in Appendix B.

3.1.3 Stage 3: Metadata Construction

To construct high-quality metadata and mitigate noise from unreliable signals, the agent first performs specialized checks on the raw multimodal signals bound to the keyframe sequence. For each keyframe, the agent performs specialized checks on the associated signals using predefined rules and determines which signals to retain. Then, based on the keyframe sequence and the filtered signals, it generates transitions describing changes between consecutive keyframes and a clip summary that describes the video clip based on the keyframe captions and verbalized tool signals, and verifies the quality and factuality of the generated text using the signals. Finally, it binds the keyframes, their captions, the keyframe-bound multimodal signals, the transitions, and the clip summary to form a metadata list for the video clip. The prompts for clip summary generation and pairwise transition generation are provided in Appendix A.3, and the metadata schema is given in Appendix C.

3.1.4 Stage 4: Instruction Data Construction

For the generated metadata, the agent applies Gate C to perform quality filtering and categorization. Specifically, this gate calls a VLM to score metadata by overall information content, logical coherence, and overall quality, outputs a decision with reasons about its suitability as instruction data, and removes unsuitable metadata. Subsequently, for each metadata entry, the agent determines the suitable task type(s) and outputs a task list with one or more of understanding, editing, and unified. The prompts for task routing and instruction generation are provided in Appendix A.4. Finally, the agent matches instruction templates based on the metadata and its task list, and requires the model to cite the signal IDs used during instruction generation to improve traceability, thereby producing high-quality instruction data. Figure 3 illustrates representative rejected cases across Gate A, Gate B, and Gate C, covering global rejection, low-quality visual inputs, generation-unsuitable frames, and task-routing mismatches. Instruction-data synthesis templates and examples are described in Appendix D, and all quality-control filtering rules are provided in Appendix E.

3.1.5 Autonomous Data Iteration

To ensure the final 100K samples are high-quality, balanced, and effective for model training, VidUEU-Agent employs a self-iterative data refinement loop. In each iteration, we train models on 30K samples, evaluate them on representative benchmarks, and return results to the agent. The agent reflects on the data's weaknesses, compares new results with the baseline and previous iterations, and updates the target distribution over capabilities, domains, and task types. Under the current policy, the agent adjusts video-type sampling in global perception to prioritize underrepresented types, shifting the domain distribution. In keyframe mining, the agent modifies sampling density and keyframe budget, and adds policy-aware weights to keyframe scoring to align with the desired keyframe distribution. During instruction data construction, task routing adjusts capability and task-type distributions, and quota-based sampling down-samples types whose ratios exceed the target ratios. The agent audits the realized distribution and records statistics, feedback, and policy changes to guide the next iteration. The final 100K dataset is scaled from the Iteration 4's target distribution.

3.2 VidUEU-100K

Based on the above framework, with DeepSeek-V3 as the primary agent LLM DeepSeek-AI et al. [2025] and Qwen3.5-9B as the VLM Qwen Team [2026], we construct the **VidUEU-100K** dataset. It contains 100K high-quality instruction-tuning samples, including 40K understanding samples, 30K editing samples, and 30K unified samples, as well as 20k fine-grained metadata.

3.2.1 Data Collection & Sources

Large-scale video sources and a scalable agent pipeline enable VidUEU-100K to cover diverse scenarios, including rare and long-tail visual cases. We source videos from large-scale video datasets, including OpenVid-1M Nan et al. [2024] and Panda-70M Chen et al. [2024b], together with videos from *Wikimedia* and YouTube. To improve coverage in specific domains, we further use specialized open-source datasets such as FineGym Shao et al. [2020], BDD100K Yu et al. [2020], and CodeVidCL Raja et al. [2025]. To build VidUEU-100K, we use 8,268 raw source videos, from which VidUEU-Agent derives approximately 8,505 semantic segments for dataset instance generation.

3.2.2 Dataset Composition & Statistics

To ensure comprehensive and balanced data, we analyze VidUEU-100K from three complementary dimensions: capability, domain, and task type. For capability, we divide the 100K instruction-tuning samples into **Understanding**, **Editing**, and **Unified**: Understanding covers fine-grained perception, visual reasoning, and multi-image logical consistency; Editing focuses on instruction-based image editing while preserving irrelevant regions; and Unified involves interleaved text–image tasks that require models to understand visual context, infer state transitions, generate target visual states, and select images consistent with the intended change. For domain, we categorize samples into Technology & Tutorial, Sports & Action, Daily Life & Human Activities, Nature & Scenery, and Gaming & ACG. For task type, the agent generates nine categories: Fine-grained Perception, Visual Reasoning, Multi-Image Logic, Editing, Grounded Edit-to-Select Generation, Interleaved Understand–Recall–Generate–Select, Stepwise Sequence Generate-and-Select, Middle-Step Gap Generate-and-Fill, and Instructional Sequential Generate-and-Select. Detailed definitions and distributions of all capability, domain, and task categories are provided in Appendix F.

4 Experiment

4.1 Experimental Setup

We LoRA-fine-tune InternVL3-8B Zhu et al. [2025] on the understanding subset and Bagel-7B-MoT Deng et al. [2025a] separately on the three capability subsets of VidUEU-100K. We then evaluate the corresponding capabilities on MMT-Bench Ying et al. [2024], MME Fu et al. [2025], LLaVA-Bench Liu et al. [2024b], MMDU Liu et al. [2024d], GEdit-Bench-En Liu et al. [2025], and WEAVEbench Chow et al. [2025]. Detailed settings are provided in Appendix G.

Table 1: Comparison of fine-tuned InternVL3-8B & Bagel, and baseline models on benchmarks.

Type	Model	Size	MMT-Bench↑	MME↑	LLaVA-Bench↑	MMDU↑
Und. Only	Qwen2.5-VL	3B	0.577	2133.6	72.3	42.5
	Qwen2.5-VL	7B	0.607	2308.6	**82.7**	59.5
	LLaMA-3.2	11B	0.456	1761.7	73.6	37.0
	InternVL3	1B	0.512	1871.9	46.4	39.0
	InternVL3	2B	0.532	2157.9	62.1	45.6
	InternVL3	8B	0.546	2308.0	69.0	53.1
	Ours(InternVL3)	8B	0.596	**2369.3**	70.8	**61.1**
Unified	Janus-Pro	7B	0.480	1784.4	58.7	42.6
	TokenFlow-XL	14B	0.532	1937.0	55.2	25.7
	Bagel	14B	0.645	1573.6	47.5	28.5
	Ours(Bagel)	14B	**0.651**	1900.2	57.9	33.2

Table 2: Comparison of fine-tuned Bagel and baseline models on GEdit-Bench-En.

Type	Model	G_SC↑	G_PQ↑	G_O↑
Edit Only	Instruct-Pix2Pix	3.575	5.491	3.684
	MagicBrush	4.677	5.656	4.518
	AnyEdit	3.178	5.820	3.212
	Step1X-Edit	7.091	6.763	6.701
Unified	Gemini-2.0-flash	6.732	6.606	6.315
	OmniGen	5.963	5.888	5.061
	Bagel	7.358	6.831	6.527
	Ours(Bagel)	**7.565**	**7.328**	**6.951**

4.2 Results on Understanding Capabilities

We fine-tune InternVL3-8B and Bagel-7B-MoT on the understanding subset to evaluate the effect of VidUEU-100K on multimodal understanding. As shown in Table 1, the data produced by VidUEU-Agent substantially improves the understanding ability of both the understanding-only model and the unified model. For InternVL3-8B, the gains are most notable on MME (2369.3) and MMDU (61.1), achieving the best results among all compared models on both benchmarks. For Bagel, fine-tuning improves all reported understanding scores and achieves the best MMT-Bench score (0.651).

Overall, these results show that the supervision generated by VidUEU-Agent effectively improves multimodal understanding and generalizes across different model families. Particularly, since MMDU evaluates multi-image reasoning and answer consistency grounded in visual evidence, the MMDU improvements of InternVL3-8B and Bagel show that VidUEU-Agent can mine semantically related keyframes from videos and model their relations in instruction data.

4.3 Results on Editing Capabilities

We fine-tune Bagel on the editing subset to evaluate the effect of VidUEU-100K on instruction-based image editing. As shown in Table 2, the data produced by VidUEU-Agent improves Bagel on all GEdit-Bench metrics, reaching 7.565 on G_SC, 7.328 on G_PQ, and 6.951 on G_O, achieving the best results among all com -pared models. These results show that VidUEU-100K improves both edit correctness and output quality, rather than only optimizing one aspect of image editing.

The two core metrics reveal a consistent but uneven gain pattern. The improvement on G_SC shows that VidUEU-Agent can extract meaningful visual changes from videos and convert them into instructions that guide the target edit more effectively. The larger gain on G_PQ further indicates that video state-transition supervision helps preserve image realism and local visual structure during editing. This is because adjacent keyframes usually share most scene content but differ in specific objects, attributes, or states, matching the edit-and-preserve nature of instruction-based image editing.

4.4 Results on Unified Capabilities

We fine-tune Bagel on the unified subset to evaluate the effect of VidUEU-100K on interleaved understanding, generation, visual memory, and cross-turn reasoning. As shown in Table 3, the data produced by VidUEU-Agent improves Bagel's Overall score on WEAVEbench from 0.446 to 0.501. The fine-tuned model outperforms the Bagel and Bagel-Zebra baselines, reaching 0.480 on Logic and 0.419 on Game.

Table 3: Comparison of fine-tuned Bagel and baseline models on WEAVEbench.

Type	Model	Size	Science↑	Creation↑	Logic↑	Game↑	Overall↑
Unified	OmniGen2	7B	**0.511**	**0.682**	**0.551**	**0.511**	**0.609**
	Ovis-U1	3B	0.402	0.557	0.364	0.357	0.422
	Bagel	14B	0.378	0.475	0.406	0.365	0.446
	Bagel-Zebra	14B	0.399	0.456	0.393	0.396	0.449
	Ours(Bagel)	14B	0.426	0.501	0.480	0.419	0.501

Table 4: Results of autonomous data iteration.

Setting	MMDU↑	GEdit ↑	WEAVE ↑
Iteration 1	29.7	6.541	0.448
Iteration 2	29.2	6.796	0.452
Iteration 3	31.4	6.873	0.458
Iteration 4	32.8	6.901	0.499
Final	**33.0**	**7.065**	**0.526**

Table 5: Ablation study of VidUEU-Agent.

Setting	MMDU↑	GEdit ↑	WEAVE ↑
Ours	**33.2**	**6.951**	**0.501**
w/o A	32.4	6.773	0.497
w/o B	24.7	5.175	0.404
w/o S	30.8	5.848	0.462
w/o C	30.5	6.303	0.454
N	22.0	5.041	0.393
w/o R	29.4	6.031	0.470

These results highlight temporal supervision in unified multimodal tasks. The gains on Logic and Game suggest that the model tracks states, follows rules, models object relations, and reasons over events across turns. This matches the structure of adjacent keyframes, which preserve scene continuity while showing changes in objects, attributes, actions, and outcomes. The modest Creation gain suggests that such supervision is more effective for structured transitions than open-ended creative generation.

4.5 Case Study

For understanding, InternVL3-8B shows hallucination and cross-image misbinding, while the fine-tuned model relies more on the visible evidence. For editing, Bagel often fails to follow background replacement instructions, whereas the fine-tuned model produces more realistic target edits while preserving irrelevant regions. For unified tasks, the fine-tuned model better executes edits and context use in early turns, but long interleaved image–text chains remain challenging because early errors may propagate to later turns.

Overall, VidUEU-100K improves instruction alignment, compositional binding, and local-edit preservation, while long-horizon interleaved interaction remains an open challenge.

4.6 Results on Autonomous Data Iteration

We further evaluate the autonomous data iteration process of VidUEU-Agent. Iterations 1–4 each construct 30K samples and jointly fine-tune BAGEL from the same initial checkpoint before evaluation. The benchmark results are then fed back into the agent, which updates the target distributions across capabilities, domains, and task types for the next iteration. The Final 100K dataset is constructed by scaling the target distribution obtained in Iteration 4, and a single BAGEL model is jointly fine-tuned on the complete dataset. Table 4 reports results on MMDU, GEdit-Bench-En, and WEAVEbench, whose aggregate test-split scores guide iteration and constitute development-guided evaluations; full results and feedback-free evaluations on MMBench, MMStar, and VSI-Bench appear in Appendix H.

The iteration results show a clear upward trend after repeated feedback-driven refinement. Although Iteration 2 slightly decreases on MMDU, it improves on GEdit and WEAVE, suggesting that early distribution updates first benefit generation-related capabilities. From Iteration 3 to Iteration 4, the gains become more consistent across all three benchmarks, indicating that the agent gradually improves the balance among capabilities, domains, and task types. Final further improves upon Iteration 4 with 100K samples.

4.7 Ablation Study

We conduct ablations to evaluate key design choices in VidUEU-Agent. Ablations train Bagel on 10K samples per capability; Ours uses separately trained Bagel models on the Final 100K subsets. We separately remove Gate A shot filtering (A), Gate B frame quality filtering (B), score-and-similarity-based greedy keyframe sampling (S), and Gate C metadata filtering (C). We compare with a no-agent



Figure 4: Case study examples comparing baseline models and VidUEU-100K fine-tuned models.

variant (N) that uses fixed-interval sampled frames as metadata and directly uses a VLM to generate QA pairs without tool evidence. Finally, we remove the consistency hard rule (R) requiring each instruction sample to cite used signal IDs. Table 5 reports the main results on MMDU, GEdit-Bench-En G_O score, and WEAVEbench overall score; the full results with the other three understanding benchmarks are provided in Appendix H.4. Baseline dataset comparisons appear in Appendix H.5.

The ablation results suggest that all components contribute to performance. Removing Gate A causes only a mild drop, suggesting that the collected videos contain relatively few meaningless or extremely dark shots after source-level curation. Removing Gate B clearly hurts all benchmarks, especially editing and unified tasks, showing that the agent must filter the many low-quality frames in raw videos. Removing score-similarity greedy sampling hurts editing more, indicating that editing data benefits from keyframes that are both informative and non-redundant, rather than from dense but weakly differentiated frame pairs. Removing Gate C degrades all tasks, showing that the final metadata filtering step is necessary to prevent inconsistent, incomplete, or task-mismatched evidence from entering instruction synthesis. The no-agent variant drops the most, showing that direct VLM-based QA generation lacks the structured quality control, tool evidence, and task routing needed to construct reliable multimodal supervision. Removing the consistency hard rule also hurts performance, indicating that evidence citation suppresses weakly grounded or low-quality samples.

5 Conclusion

In this paper, we study the bottleneck of high-quality instruction data for multimodal tasks, and propose a scalable agent-based solution that leverages raw videos as a rich and dynamic data source.

Specifically, VidUEU-Agent constructs instruction data through four stages: global perception, keyframe mining, metadata construction, and instruction data construction, and refines the data distribution through autonomous data iteration using downstream feedback. Using VidUEU-Agent, we construct VidUEU-100K, a large-scale dataset with 100K instruction samples spanning understanding, editing, and unified capabilities. Extensive experiments across representative benchmarks show that fine-tuning baseline models with data produced by VidUEU-Agent brings consistent performance improvements, validating the effectiveness and generalizability of the proposed agent pipeline. We hope VidUEU-Agent and VidUEU-100K provide a reproducible and extensible foundation for scalable multimodal data construction, and facilitate the development of unified multimodal models that more reliably integrate understanding and generation.

Acknowledgments and Disclosure of Funding

Funding. This research was supported by STI 2030-Major Projects (Grant No. 2022ZD0208800). We also acknowledge support from the Undergraduate Future Academic Star Program and the Undergraduate Natural Science Cultivation Program of Zhejiang University. This research was also partially supported by the HKU Startup Fund and Institute of Data Science.

Competing Interests. The authors declare no competing interests.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

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, 2025b. URL `https://arxiv.org/abs/2502.13923`.

Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions, 2023. URL `https://arxiv.org/abs/2211.09800`.

Pu Cao, Feng Zhou, Lu Yang, Tianrui Huang, and Qing Song. Image is all you need to empower large-scale diffusion models for in-domain generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18358–18368, 2025.

Brandon Castellano. Pyscenedetect. `https://github.com/Breakthrough/PySceneDetect`.

Hang Chen, Qian Xiang, Jiaxin Hu, Meilin Ye, Chao Yu, Hao Cheng, and Lei Zhang. Comprehensive exploration of diffusion models in image generation: a survey. *Artificial Intelligence Review*, 58 (4):99, 2025a.

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*, 2025b.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions, 2023. URL `https://arxiv.org/abs/2311.12793`.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, 2024a.

Tsai-Shien Chen, Aliaksandr Siarohin, Willi Menapace, Ekaterina Deyneka, Hsiang-wei Chao, Byung Eun Jeon, Yuwei Fang, Hsin-Ying Lee, Jian Ren, Ming-Hsuan Yang, et al. Panda-70m: Captioning 70m videos with multiple cross-modality teachers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13320–13331, 2024b.

Wei Chen, Lin Li, Yongqi Yang, Bin Wen, Fan Yang, Tingting Gao, Yu Wu, and Long Chen. Comm: A coherent interleaved image-text dataset for multimodal understanding and generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8073–8082, 2025c.

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, 2025d. URL https://arxiv.org/abs/2501.17811.

Wei Chow, Jiachun Pan, Yongyuan Liang, Mingze Zhou, Xue Song, Liyu Jia, Saining Zhang, Siliang Tang, Juncheng Li, Fengda Zhang, Weijia Wu, Hanwang Zhang, and Tat-Seng Chua. WEAVE: Unleashing and Benchmarking the In-context Interleaved Comprehension and Generation, November 2025. URL http://arxiv.org/abs/2511.11434. arXiv:2511.11434 [cs].

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining, 2025a. URL https://arxiv.org/abs/2505.14683.

Mengjie Deng, Guanting Dong, and Zhicheng Dou. Toolscope: An agentic framework for vision-guided and long-horizon tool use. *arXiv preprint arXiv:2510.27363*, 2025b.

Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, Quanquan Gu, James Zou, Kai-Wei Chang, and Wei Wang. Enhancing Large Vision Language Models with Self-Training on Image Comprehension. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 131369–131397. Curran Associates, Inc., 2024. doi: 10.52202/079017-4175. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ed45d6a03de84cc650cae0655f699356-Paper-Conference.pdf.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=DgH9YCsqWm.

Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaojian Ma, Tao Yuan, Yue Fan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Multi-modal agent tuning: Building a vlm-driven agent for efficient tool usage, 2025. URL https://arxiv.org/abs/2412.15606.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu,

Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL `https://arxiv.org/abs/2407.21783`.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, N. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges. In *International Joint Conference on Artificial Intelligence*, 2024. URL `https://api.semanticscholar.org/CorpusID:267412980`.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolo, April 2024. URL `https://doi.org/10.5281/zenodo.10983461`.

Soham Joshi, Shwet Kamal Mishra, and Viswanath Gopalakrishnan. Text-vqa aug: Pipelined harnessing of large multimodal models for automated synthesis. *arXiv preprint arXiv:2511.02046*, 2025.

Kat Kampf and Nicole Brichtova. Experiment with gemini 2.0 flash native image generation. https://developers.googleblog.com/experiment-with-gemini-20-flash-native-image-generation/, March 2025. Google Developers Blog.

Somnath Kumar, Yash Gadhia, Tanuja Ganu, and Akshay Nambi. Mmctagent: Multi-modal critical thinking agent framework for complex visual reasoning. *arXiv preprint arXiv:2405.18358*, 2024.

Ang Li, Charles Wang, Deqing Fu, Kaiyu Yue, Zikui Cai, Wang Bill Zhu, Ollie Liu, Peng Guo, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. Zebra-cot: A dataset for interleaved vision-language reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c6XIVI3TiQ.

Lei Li, Yuwei Yin, Shicheng Li, Liang Chen, Peiyi Wang, Shuhuai Ren, Mukai Li, Yazheng Yang, Jingjing Xu, Xu Sun, Lingpeng Kong, and Qi Liu. M^3it: A large-scale dataset towards multi-modal multilingual instruction tuning, 2023. URL https://arxiv.org/abs/2306.04387.

Zejian Li, Chenye Meng, Yize Li, Ling Yang, Shengyuan Zhang, Jiarui Ma, Jiayi Li, Guang Yang, Changyuan Yang, Zhiyuan Yang, et al. Laion-sg: An enhanced large-scale dataset for training complex image-text models with structural annotations. *arXiv preprint arXiv:2412.08580*, 2024.

Chia Xin Liang, Pu Tian, Caitlyn Heqi Yin, Yao Yua, Wei An-Hou, Li Ming, Xinyuan Song, Tianyang Wang, Ziqian Bi, and Ming Liu. A comprehensive survey and guide to multimodal large language models in vision-language tasks. *arXiv preprint arXiv:2411.06284*, 2024.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306, 2024a.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2024b. URL https://arxiv.org/abs/2310.03744.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, Guopeng Li, Yuang Peng, Quan Sun, Jingwei Wu, Yan Cai, Zheng Ge, Ranchen Ming, Lei Xia, Xianfang Zeng, Yibo Zhu, Binxing Jiao, Xiangyu Zhang, Gang Yu, and Daxin Jiang. Step1x-edit: A practical framework for general image editing, 2025. URL https://arxiv.org/abs/2504.17761.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024c.

Ziyu Liu, Tao Chu, Yuhang Zang, Xilin Wei, Xiaoyi Dong, Pan Zhang, Zijian Liang, Yuanjun Xiong, Yu Qiao, Dahua Lin, and Jiaqi Wang. Mmdu: A multi-turn multi-image dialog understanding benchmark and instruction-tuning dataset for lvlms. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 8698–8733. Curran Associates, Inc., 2024d. doi: 10.52202/079017-0278. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/1057053100de064a44286239724f7865-Paper-Datasets_and_Benchmarks_Track.pdf.

Yadong Lu, Jianwei Yang, Yelong Shen, and Ahmed Awadallah. Omniparser for pure vision based gui agent, 2024. URL https://arxiv.org/abs/2408.00203.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. *arXiv preprint arXiv:2407.02371*, 2024.

OpenAI. Introducing gpt-4.1 in the api, April 2025. URL https://openai.com/index/gpt-4-1/.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi

Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024. URL https://arxiv.org/abs/2410.21276.

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

Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation, 2024. URL https://arxiv.org/abs/2412.03069.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Sana Javaid Raja, Adeel Zafar, and Aqsa Shoaib. A dataset for programming-based instructional video classification and question answering. In *Proceedings of the First Workshop of Evaluation of Multi-Modal Generation*, pages 1–9, 2025.

Vipula Rawte, Amit Sheth, and Amitava Das. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*, 2023.

N Dinesh Reddy, Dylan Snyder, Lona Kiragu, Mirajul Mohin, Shahrear Bin Amin, and Sudeep Pillai. Orion: A unified visual agent for multimodal perception, advanced visual reasoning and execution, 2025. URL https://arxiv.org/abs/2511.14210.

Haris Riaz, Sourav Sanjukta Bhabesh, Vinayak Arannil, Miguel Ballesteros, and Graham Horwood. MetaSynth: Meta-prompting-driven agentic scaffolds for diverse synthetic data generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18770–18803, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.962. URL https://aclanthology.org/2025.findings-acl.962/.

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020.

Hengyu Shen, Tiancheng Gu, Bin Qin, Lan Wu, Yuling Wu, Shuo Tan, Zelong Sun, Jun Wang, Nan Wu, Xiang An, Weidong Cai, Ziyong Feng, and Kaicheng Yang. Danqing: An up-to-date large-scale chinese vision-language pre-training dataset, 2026. URL https://arxiv.org/abs/2601.10305.

Haoran Sun, Haoyu Bian, Shaoning Zeng, Yunbo Rao, Xu Xu, Lin Mei, and Jianping Gou. Dataseta-gent: A novel multi-agent system for auto-constructing datasets from real-world images. *arXiv preprint arXiv:2507.08648*, 2025.

Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. Synthesizing post-training data for llms through multi-agent simulation, 2025. URL https://arxiv.org/abs/2410.14251.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 402–419, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58536-5.

Jingqi Tong, Yurong Mou, Hangcheng Li, Mingzhe Li, Yongzhuo Yang, Ming Zhang, Qiguang Chen, Tianyi Liang, Xiaomeng Hu, Yining Zheng, Xinchi Chen, Jun Zhao, Xuanjing Huang, and Xipeng Qiu. Thinking with video: Video generation as a promising multimodal reasoning paradigm, 2025a. URL https://arxiv.org/abs/2511.04570.

Shengbang Tong, David Fan, Jiachen Li, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17001–17012, 2025b.

Guo-Hua Wang, Shanshan Zhao, Xinjie Zhang, Liangfu Cao, Pengxin Zhan, Lunhao Duan, Shiyin Lu, Minghao Fu, Xiaohao Chen, Jianshan Zhao, Yang Li, and Qing-Guo Chen. Ovis-u1 technical report, 2025. URL https://arxiv.org/abs/2506.23044.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023.

Chenyuan Wu, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, Junjie Zhou, et al. Omnigen2: Exploration to advanced multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025.

Jiayang Wu, Wensheng Gan, Zefeng Chen, Shicheng Wan, and Philip S Yu. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 2247–2256. IEEE, 2023.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation, 2024. URL https://arxiv.org/abs/2409.11340.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

Weikai Xu, Chengrui Huang, Shen Gao, and Shuo Shang. Llm-based agents for tool learning: A survey: W. xu et al. *Data Science and Engineering*, pages 1–31, 2025.

Yan Yang, Haochen Tian, Yang Shi, Wulin Xie, Yi-Fan Zhang, Yuhao Dong, Yibo Hu, Liang Wang, Ran He, Caifeng Shan, et al. A survey of unified multimodal understanding and generation: Advances and challenges. *Authorea Preprints*, 2025.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, November 2024. ISSN 2095-5138. doi: 10.1093/nsr/nwae403. URL https://doi.org/10.1093/nsr/nwae403. _eprint: https://academic.oup.com/nsr/article-pdf/11/12/nwae403/61201557/nwae403.pdf.

Kaining Ying, Fanqing Meng, Jin Wang, Zhiqian Li, Han Lin, Yue Yang, Hao Zhang, Wenbo Zhang, Yuqi Lin, Shuo Liu, Jiayi Lei, Quanfeng Lu, Runjian Chen, Peng Xu, Renrui Zhang, Haozhe Zhang, Peng Gao, Yali Wang, Yu Qiao, Ping Luo, Kaipeng Zhang, and Wenqi Shao. Mmt-bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask agi, 2024. URL https://arxiv.org/abs/2404.16006.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. *arXiv preprint arXiv:2411.15738*, 2024. URL https://arxiv.org/abs/2411.15738.

Qiying Yu, Quan Sun, Xiaosong Zhang, Yufeng Cui, Fan Zhang, Yue Cao, Xinlong Wang, and Jingjing Liu. Capsfusion: Rethinking image-text data at scale. *arXiv preprint arXiv:2310.20550*, 2023.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9556–9567, 2024.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing, 2024. URL https://arxiv.org/abs/2306.10012.

Tianyi Zhang, Zheng Wang, Jing Huang, Mohiuddin Muhammad Tasnim, and Wei Shi. A survey of diffusion based image generation models: Issues and their solutions. *arXiv preprint arXiv:2308.13142*, 2023.

Yidan Zhang, Mutian Xu, Yiming Hao, Kun Zhou, Jiahao Chang, Xiaoqiang Liu, Pengfei Wan, Hongbo Fu, and Xiaoguang Han. Vc-agent: An interactive agent for customized video dataset collection. *arXiv preprint arXiv:2509.21291*, 2025a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2025b. URL https://arxiv.org/abs/2410.02713.

Hao Zhao, Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Long is more for alignment: A simple but tough-to-beat baseline for instruction fine-tuning. *arXiv preprint arXiv:2402.04833*, 2024.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

Wenliang Zhao, Yongming Rao, Zuyan Liu, Benlin Liu, Jie Zhou, and Jiwen Lu. Unleashing text-to-image diffusion models for visual perception. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5729–5739, 2023.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

Xianwei Zhuang, Yuxin Xie, Yufan Deng, Liming Liang, Jinghan Ru, Yuguo Yin, and Yuexian Zou. Vargpt: Unified understanding and generation in a visual autoregressive multimodal large language model, 2025. URL https://arxiv.org/abs/2501.12327.

Appendix

A VidUEU-Agent Pipeline

A.1 Stage 1: Global Perception

The prompts for shot-level summarization, chapter construction, chapter-level summarization, video-level summarization, and Gate A shot admission are shown in Fig. 5, Fig. 6, and Fig. 7.

Shot Profiling Prompt

```
You are building a visual-only shot profile for a video pipeline.
Use only the images.
Return strict JSON with keys:
vlm_summary, shot_role, scene_type, visual_quality_hint, informativeness_hint,
candidate_sampling_hint.
Allowed hint values for visual_quality_hint, informativeness_hint, and
candidate_sampling_hint:  low, medium, high.
```

Figure 5: Prompt for shot-level summarization.

A.2 Stage 2: Keyframe Mining

The implementation prompts and rules for hierarchical tool planning, tool signal verbalization, frame value scoring, and quota allocation with weighted facility-location selection are shown in Fig. 9, Fig. 10, Fig. 11, and Fig. 12.

A.3 Stage 3: Metadata Construction

The prompts for clip summary generation and pairwise transition generation are shown in Fig. 8 and Fig. 13.

A.4 Stage 4: Instruction Data Construction

The prompts for task routing and instruction generation are shown in Fig. 14 and Fig. 15.

Chapter Construction and Summary Prompts

```
[Chapter Construction Prompt / Pairwise Merge Decision Prompt]
You are a strict JSON decision engine for merging two adjacent video shots into
chapters.
Decide whether the two shots belong to the same chapter.
Return exactly one valid JSON object and nothing else.  Do not use markdown fences.
The JSON schema is:
{
"same_chapter":  true,
"confidence":  0.0,
"hard_boundary":  false,
"reason_type":  "visual_continuation",
"notes":  "short reason"
}
Rules:  same_chapter and hard_boundary must be JSON booleans; confidence must be in
[0.0, 1.0]; reason_type must be a short snake_case string; notes must be under 20
words; if evidence is mixed or uncertain, still return valid JSON with lower
confidence; base the decision only on the provided shot summaries and hints.
LEFT role:  <left_shot.shot_role>
LEFT vlm:  <left_shot.vlm_summary>
LEFT scene_type:  <left_shot.scene_type>
LEFT informativeness_hint:  <left_shot.informativeness_hint>
RIGHT role:  <right_shot.shot_role>
RIGHT vlm:  <right_shot.vlm_summary>
RIGHT scene_type:  <right_shot.scene_type>
RIGHT informativeness_hint:  <right_shot.informativeness_hint>
[Chapter Summary Prompt]
You are a robust chapter summarizer for a structured video understanding pipeline.
Read all provided shot summaries for this chapter and merge them into one concise,
fluent, informative English paragraph.  Summarize the chapter as a coherent local
segment, not disconnected bullet points.  Preserve important subjects, actions, stages,
transitions, and outcomes; remove redundancy; reflect progression when present; stay
grounded in the shot summaries only.
Writing requirements:  write plain English only; output exactly one short paragraph;
do not use bullet points, numbering, JSON, markdown, labels, or quotation marks; keep
the summary compact, specific, and easy for downstream planning, scoring, and
visualization.
Return the chapter summary only.
[Video-level Summary Prompt]
You are a robust video summarizer for a structured video understanding pipeline.
Read all provided chapter summaries and merge them into one concise, fluent,
informative English paragraph.  Summarize the global content of the whole video rather
than isolated chapter fragments.  Capture the overall subject, major stages, important
transitions, and final outcomes when supported; remove redundancy; reflect global
progression when present; stay grounded in the chapter summaries only.
Writing requirements:  write plain English only; output exactly one short paragraph;
do not use bullet points, numbering, JSON, markdown, labels, or quotation marks; keep
the summary compact, globally informative, and easy for downstream planning, scoring,
and visualization.
Return the video summary only.
```

Figure 6: Prompt for chapter construction, chapter-level summarization, and video-level summarization.

Shot Admission Prompt

```
You are a strict shot-admission judge for a video frame selection pipeline.
Judge whether the current shot is worth sending to downstream frame sampling.
Keep shots only if they are visually usable and likely to yield informative frames.
Reject shots that are mainly low-information filler, weak transitions, visually poor,
or not worth spending frame-sampling budget on.
Return strict JSON with keys usable_score, quality_score, informativeness_score,
decision, reason, evidence.
Video summary: <video_profile.video_summary>
Video type: <video_profile.video_type>
Chapter role: <getattr(chapter, 'chapter_role', '')>
Chapter summary: <getattr(chapter, 'chapter_summary', '')>
Shot summary: <shot.vlm_summary>
Shot role: <shot.shot_role>
Scene type: <shot.scene_type>
Visual quality hint: <shot.visual_quality_hint>
Informativeness hint: <shot.informativeness_hint>
Sampling hint: <shot.candidate_sampling_hint>
Representative frame count: <len(shot.rep_frames or [])>
```

Figure 7: Prompt for Gate A shot admission.

Clip Summary Generation Prompt

```
You are writing a concise <level> summary for a video segment.
Summarize ONLY from the provided <level> evidence.
Keep it factual, concrete, and short.
Return 1-3 sentences only.
<level_title> evidence:
<evidence_lines>
```

Figure 8: Prompt for clip summary generation.

Tool Planning Prompts

```
[Tool Planning Prompt (chapter-level)]

You are the chapter-level tool planner for a structured video keyframe mining
pipeline.

A chapter is a contiguous semantic segment ...  usually covering one local topic,
stage, or scene block.

Your job is NOT to choose frames.  Your job is to choose the 2 most important tools
for each chapter ...  so the pipeline can understand that chapter efficiently ...

Planning principles:

- Choose the smallest sufficient tool set for each chapter.

- Reflect what needed to understand the whole chapter broadly, not every possible
detail.

- Use the video summary as global context, but judge each chapter independently.

- This stage does not decide score_focus.

- Reasons must be short, concrete, and grounded in the provided summaries.

Output requirements:

- Return valid JSON only.

- Use English only.

[Tool Planning Prompt (shot-level)]

You are the shot-level tool planner for a structured video keyframe mining pipeline.

The video has been decomposed into chapters and shots.  Later stages will choose
keyframes ...

Your job is NOT to choose frames.  Your job is to choose the final tool set for each
shot so the pipeline can understand that specific shot ...

You will be given:

- the current chapter summary

- the chapter-level recommended tools

- all shot summaries inside that chapter

Planning principles:

- Evaluate each shot independently based on its own summary.

- Treat chapter-level tools as recommendations, not fixed final output.

- Prefer adding at least 1 extra suitable tool when the shot content supports it; skip
only when none is suitable.

- Remove a chapter-level tool only when it is unsuitable for this shot.

- Add only the minimum necessary extra tools from the provided extra candidate list.

- Keep the final tool set small but sufficient.

- final tool_actions must contain the complete final tool set for the shot, with no
duplicates.

- score_focus means the 1 or 2 most important tools inside the selected set; it must be
a subset of selected tools.

- Reasons must be short, concrete, and grounded in the provided summaries.

Output requirements:

- Return valid JSON only.

- Use English only.
```

Figure 9: Prompt for hierarchical tool planning.

Tool Signal Semanticizer Prompt

```
You are a frame-level tool-signal semanticizer for structured video keyframe mining.

You will receive multiple candidate frames from the same shot, the tools actually
executed for this shot, and each active tool's purpose.

Your task is NOT to score frames.

Your task is to convert raw tool evidence, values, scores, and tool-specific hints
into frame-specific semantic descriptions that are faithful to the actual tool outputs
and easy to understand.

Core requirements:

- Be strictly evidence-grounded.

- Explain the meaning of tool outputs rather than merely copying raw snippets.

- Use tool purpose as context, but base every description on actual per-frame evidence.

- Explain numeric values, categorical levels, or structured fields in plain language
when present.

- Preserve supported directionality, such as stronger vs weaker motion or higher vs
lower confidence.

- Do not invent thresholds, hidden formulas, unseen objects, actions, relations, or
text.

- Do not confuse detector confidence, motion magnitude, or tool values with semantic
importance.

- Use only tools that were actually executed and provided in the input.

- For tools not executed or without useful frame-specific evidence, return an empty
string for that field.

- semantic_text must be a short merged frame-specific description from useful
active-tool evidence.

- Do not score, rank, or compare quality.

- Output JSON only.

- Use English only.
```

Figure 10: Prompt for tool signal verbalization.

Value Scoring Prompt

```
You are a frame value scorer for structured video keyframe mining.

You will receive multiple candidate frames from the same shot; video, chapter, and shot summaries;
the tools selected for this shot and their purposes; score_focus, which names the 1 or 2 relatively
more important selected tools; frame-level semantic descriptions rewritten from raw tool outputs;
and an extra factual caption for each frame.

Your task is to assign one ABSOLUTE value score in the range [0, 1] to each frame.

How to reason:

- Understand the video, chapter, and shot from the summaries.

- Inspect each frame's semantic_text, tool-specific descriptions, and factual caption.

- Judge what concrete information each source contributes, how clear and specific it is, and how
useful it is for understanding the shot.

- Treat tool-derived signals as auxiliary evidence; keep the frame caption as a central focus.

- Integrate all evidence to decide the final score.

How to use score_focus:

- score_focus is not a score dimension list.

- When meaningful evidence from a score_focus tool is present, give it extra attention.

- Do not ignore useful evidence from other selected tools.

Scoring principles:

- Score each frame on its own absolute usefulness as a final keyframe for this shot.

- Differentiate frames when they differ in clarity, specificity, or decision value.

- Prefer frames that are clear, informative, representative, non-redundant, decision-useful, and
easy to interpret.

- Reward frames that capture important entities, actions, states, transitions, text, speech-derived
meaning, or step-specific evidence relevant to the shot.

- Reward frames where caption, semantic_text, and tool semantics reinforce a clear grounded
interpretation.

- Penalize frames that are vague, redundant, weakly grounded, semantically thin, generic,
repetitive, or less useful despite technical cleanliness.

- Do not treat tool presence, detector confidence, motion magnitude, or other tool-side values as
value by themselves.

Scoring meaning:

- 0.0-0.1:  almost no useful evidence.

- 0.1-0.25:  very weak; evidence is vague, redundant, or not useful enough.

- 0.25-0.45:  limited; partially useful but missing clarity or informativeness.

- 0.45-0.65:  moderate; understandable and useful but not especially strong.

- 0.65-0.8:  strong; clearly informative, representative, or decision-worthy.

- 0.8-0.92:  very strong; strong candidate for final keyframe selection.

- 0.92-1.0:  exceptional; unusually clear, information-dense, and highly useful.

High-value frames usually clearly show the key subject, state, action, step, transition, result,
important text, speech-derived meaning, object evidence, or motion/change evidence; provide
non-trivial information beyond generic background or redundant nearby frames; and help justify
retaining the frame as a representative.

Output JSON only.

Use English only.
```

Figure 11: Prompt for frame value scoring.

Quota Planning Prompt

```
You are a hierarchical quota planner for structured video keyframe mining.

You must plan quotas at three levels:

- global

- chapter

- shot

Goal:

Under the allowed total quota range, choose one fixed final global quota and allocate
it across shots so later stages can retain the most valuable representative frames.
You are not choosing frames directly.

Critical planning rules:

1.  Choose one integer global_budget inside the allowed total quota range.

2.  Do not default to quota=1 merely because there is only one chapter or shot.

3.  The sum of all shot target_quota values must equal global_budget exactly.

4.  Use video, chapter, and shot summaries to judge where more quota is needed for
distinct coverage, richer content, or stronger representativeness.

5.  Duration is supporting evidence, not the only reason; longer duration justifies
more quota only when summaries suggest broader coverage, multiple stages, transitions,
or meaningful states/results.

6.  Weak, redundant, or semantically flat shots may receive quota 0.

7.  Prefer coverage of distinct stages, states, results, events, or subtopics over
near-duplicates.

8.  Judge each shot independently while keeping the final allocation globally
balanced.

How to use the input:

- Start from the video summary to understand broad content and final selection
coverage.

- Use chapter summaries to judge which chapters deserve broader coverage.

- Reason shot by shot using each shot summary and duration.

- Assign more quota when a shot suggests multiple steps, transitions, states, results,
or distinct moments.

- Assign low or zero quota to weak, repetitive, or semantically flat shots.

- Reason fields should explain quotas in terms of summary meaning and coverage need.

Output JSON only.

Use English only.
```

Figure 12: Prompt and rule for quota allocation and weighted facility-location selection.

Pairwise Transition Prompt and Normalization Rules

```
You are analyzing two consecutive keyframes A -> B from a short video segment.

Write exactly ONE transition string describing meaningful visible changes from A to B.

Use images as primary evidence. Captions, video type, global summary, and modality
advice are secondary hints; if they conflict with the images, trust the images.

Describe only visible changes that affect scene understanding, including object
appearance/disappearance, position, state, visible text, camera/view change, or shot
change. Ignore tiny flicker, compression noise, blur differences, glare, reflections,
and exposure shifts unless they affect meaningful content. Do not infer hidden causes,
intentions, or off-screen events.

Normalize changes into atomic edits. Each edit should use one target and one allowed
verb:

add, remove, move, rotate, open, close, turn on, turn off, change text, zoom, pan,
tilt, cut.

Use image-frame locations such as left side, center-right, upper-left, or bottom
center. Avoid object-centric or anatomical left/right.

Decision rules:

- If B is a clear new shot, use cut to new shot.

- If the main change is camera/view motion, prefer zoom/pan/tilt over object move.

- Use add/remove for appearance changes, change text for visible text changes, and
open/close or turn on/off for clear state changes.

- For driving or dashcam scenes, do not treat ego-motion or optics artifacts as object
edits.

- Edits(k) must match the number of edits; if there is no meaningful change, output
Edits(0): none.

Context:

- Video type: <video_type>

- Global summary: <summary>

- Modality advice: <advice>

Captions:

A: <cap_a>

B: <cap_b>

OUTPUT FORMAT:

Edits(k): 1) ...; 2) ... | Shot: <none/minor/scene_cut> | View: <none/minor/major>
| Subject: <same/unclear/different> | Uncertainty: <none or short note>

Return ONLY one line in the exact format above.
```

Figure 13: Prompt for pairwise transition generation.

Task Routing Prompt

```
You are a quota allocator for dataset instance generation.

You MUST base your decision ONLY on the EVIDENCE PACK lines below.

Each evidence line begins with an evidence_id in brackets, e.g. [S_SEGEMENT_SUM], [F00_CAP],
[P00].

GOAL (highest priority: maximize VALID OUTPUT COUNT): assign quotas within the limits for THIS
segment; prefer producing MORE dialog instances when feasible ...; if allocating edit quota, choose
suitable frame pairs ONLY from the provided candidate pair list.

<global_mix_note>

IMPORTANT: GLOBAL_MIX is SOFT guidance. Do NOT set dialog quotas to 0 just to satisfy a mix ratio.
If unsure, choose MORE dialog rather than fewer.

EDIT ALLOCATION POLICY (do NOT be overly conservative): candidate edit pairs are already
pre-screened; do NOT re-apply stricter hidden rules. If there is at least 1 candidate pair and no
explicit evidence saying edits are unsuitable, allocate a NON-ZERO edit quota. When uncertain,
prefer a SMALL non-zero edit quota ...; allocate edit ONLY from the provided candidate pair list.

Practical heuristic: if candidate_pairs_count >= 4, usually allocate edit 3-4 unless hard evidence
says unsuitable; if candidate_pairs_count is 1-3, usually allocate edit 2-3; if
candidate_pairs_count == 0, set edit = 0.

CRITICAL INDEXING: this segment has N keyframes indexed by POSITION 0..N-1. All frame and pair
indices in your output refer to these keyframe POSITIONS.

EVIDENCE USAGE POLICY:

1) ALWAYS read and use segment-level background evidence FIRST: [S_SEGEMENT_SUM].

2) Then use frame-level evidence [Fxx_*] and pair/change evidence [Pxx]/[Cxx] for specific
instances.

3) Every type with quota > 0 MUST cite at least ONE segment-level evidence id (S_*) in its
rationale.

4) Do NOT treat any tool signal as ground truth; treat all evidence as imperfect.

EDIT PAIR QUALITY: edit generation requires small, controllable edits. ONLY choose pairs from
candidate pairs; prefer reviewed pairs [Pxx] when available; if no candidate pair satisfies
small-edit constraints, set edit quota to 0.

DIALOG TYPES: dialogs must remain grounded in evidence and not invent details.

TYPE-SPECIFIC GUIDELINES: edit only when a small-edit pair exists; unify only when reviewed edit
pairs exist.

INPUTS:

Candidate instruction task types: <candidate_types_json>

Quota limits: <quota_lines>

Edit candidate pairs: <mb_pairs_txt>

Reviewed edit pairs: <reviewed_txt>

EVIDENCE PACK: <evidence_pack_text>

OUTPUT FORMAT (strict JSON only, no extra text):

{

"quotas": {"open_ended_qa": int, "edit": int, "multiple_choice_qa": int, "yes_no_qa": int,
"multi_image_dialogue": int, "unify": int},

"edit_pairs": [{"from": int, "to": int}, ...],

"rationales": [{"type": str, "evidence_refs": [str, ...], "reason": str}]

}

RATIONALES REQUIREMENTS: provide one rationale entry per type with quota > 0; each rationale MUST
include at least one segment-level evidence id (S_*); for dialog rationales, cite at least one
frame-level evidence id (Fxx_*).
```

Figure 14: Prompt for task routing.

Task Generation Prompts and Rules

[Understanding Generation]

Generate instruction-tuning instances from an EVIDENCE PACK, where each line starts with an evidence_id such as [F00_CAP], [S_SEGEMENT_SUM], or [P00].

<fewshot_block>

Rules: cite evidence_refs for every item; use keyframe positions 0..N-1; do not invent unsupported details; output concise JSON only.

SEGMENT META: <seg_meta_json>

DEFAULT IMAGE: image_pos=<default_pos>, image_path=<default_img>

<body>

EVIDENCE PACK: <evidence_pack_text>

Target tasktype: <tasktype>; target item count: <quota>

Output JSON schema: <schema>

[Single-Image and Multiple-Choice Rules]

Single-image questions must be answerable from image_pos only, without process or timeline wording. Segment evidence may be used only as light background; evidence_refs must include frame-level evidence for the same image_pos. Multiple-choice items must have exactly one correct option and three plausible same-type distractors. Avoid fragile exact counting; use "from the camera viewpoint" for left/right relations.

[Dialog Generation and Repair]

<kind_rules>

<base_rules>

<image_block>

EVIDENCE PACK: <dlg_ev>

Generate <quota> dialog(s) of type: <kind>.

Output JSON schema: <schema>

Repair rule: output exactly <want> turns, alternate user/assistant from user, cite evidence_refs in every turn, include frame-level evidence per turn, and avoid before/after/earlier/later/next/previous.

CURRENT TURNS: <turns_json>

[Edit Pair Review and Instruction]

Validate consecutive keyframe pairs for localized instruction-based image editing. Accept only small, controllable, same-shot changes with at most 1-2 visible semantic edits. Reject scene cuts, major viewpoint changes, camera-motion-only changes, and unsupported target content. The instruction must cover the true source-target change, add no extra changes, and end with: "Do not change anything else."

Segment summary: <segment_summary_short>

INPUT PAIRS: <input_pairs_json>

Output JSON fields: pairs[from, to, ok, edit_count, edits, instruction, reason]

[Unified Task Generation]

<system_style>

Generate a grounded multi-step vision-language task from structured evidence. Use only the provided evidence, keep fields concise, and keep explanations consistent with evidence_refs.

TASK TYPE: <task_type>

TASK PAYLOAD: <task_payload_json>

EVIDENCE PACK: <evidence_pack_text>

Style: write as a natural multimodal instruction sample; make select.question choose a candidate frame/image; explain why the target is uniquely supported; avoid speculation; use select.answer format target:FXX.

Return JSON schema: <schema_json>

Figure 15: Prompt for instruction generation.

A.5 LLM–VLM Design and Model Selection

LLM–VLM Design Comparison. We compare the metadata-driven LLM design with a direct VLM alternative for filtering, task routing, and instruction generation. Both settings use the same input data, prompt templates, generation parameters, and GPT-4o evaluation protocol, each producing 200 final samples. As shown in Table 6, the LLM design outperforms the VLM alternative in both efficiency (12.4 vs. 5.6 samples/min) and quality (4.18 vs. 2.72), supporting the effectiveness of metadata-driven routing and instruction synthesis.

Table 6: Comparison between the metadata-driven LLM design and the direct VLM alternative.

Setting	Final Samples/min ↑	Quality ↑
Original LLM setting	**12.4**	**4.18**
Direct VLM setting	5.6	2.72

VLM Model Selection. To examine the efficiency–quality trade-off in VLM selection, we compare Qwen3.5-9B and Qwen3.5-27B under the same agent workflow, input data, prompt templates, and generation parameters. Both models are evaluated on a single H20 GPU using their respective maximum stable concurrency settings. Each setting generates 200 samples, with quality assessed by GPT-4o on a five-point scale. As shown in Table 7, increasing the model size from 9B to 27B yields only a marginal quality improvement (4.16 to 4.21), while reducing throughput from 12.4 to 4.1 samples/min/GPU. Therefore, we select Qwen3.5-9B as the VLM in our pipeline for its favorable balance between generation quality and computational efficiency.

Table 7: Efficiency and quality comparison of VLM candidates under the same agent workflow.

VLM	Samples/min/GPU ↑	GPT-4o Quality ↑
Qwen3.5-9B	**12.4**	4.16
Qwen3.5-27B	4.1	**4.21**

B Tool Invocation Framework

B.1 Tool Pool and Output Types

Tool invocation supplements frame-level semantic evidence when visual captions alone are insufficient. The current tool pool includes PaddleOCR Cui et al. [2025], Whisper ASR Radford et al. [2022], Ultralytics YOLOv8 Jocher et al. [2024], RAFT optical flow Teed and Deng [2020], and OmniParser Lu et al. [2024]. Table 8 summarizes the primary role, applicable scenarios, and retained output of each tool.

B.2 Tool Evidence Verbalization and Scoring Rules

Table 9 summarizes the rules used in stage 2 to verbalize tool signals and score each frame.

C Metadata Schema Definition

A minimal metadata example is shown in Fig. 16. The `video` field comes from Stage 1 and contains the video summary, start and end timestamps, video summary, and a coarse label. The `chapters` field comes from Stage 1 and contains ordered shots, their time ranges, chapter summaries, and labels indicating each chapter's role in the video. The `shots` field comes from Stage 1 and contains shot-level time ranges, shot summaries, and hints about image quality and informativeness. The `keyframes` field comes from Stage 2 and contains selected representative frames with their timestamps, captions, quality judgments, tool-produced semantic signals, and value scores. The `transitions` field comes from Stage 3 and contains ordered frame pairs, whether they remain within the same shot, whether the main subject is consistent, the concrete transition between the frames, and a reliability estimate.

Table 8: Summary of auxiliary perception tools used for frame-level and shot-level information extraction.

Tool	Role	Applicable scenarios
PaddleOCR	Scene-text extraction from video frames	Used to read titles, labels, menu items, tables, subtitles, document text, and screen text, especially in slides, screen recordings, documents, and software demonstrations
Whisper ASR	Speech-content recovery around a shot	Used when speech provides semantic cues near candidate frames, including narration, spoken instructions, or dialogue context
YOLOv8	Object and entity detection	Used to identify people, objects, tools, and scene entities that provide visual grounding for frame understanding
RAFT	Motion estimation between adjacent candidate frames	Used when temporal dynamics or motion strength is informative for candidate-frame analysis
OmniParser	Structured GUI understanding	Used for GUI screenshots and software-interface frames where interface layout and interactive regions are important

Minimal Metadata Schema

`segment`	`video_id, start, end, summary, type`
`chapters`	`chapter_id, time_span, summary, role`
`shots`	`shot_id, chapter_id, time_span, summary, role, scene_type, quality_hint, info_hint`
`keyframes`	`frame_id, shot_id, timestamp, caption, quality, value_score, tool_semantics, evidence_id`
`transitions`	`source, target, atomic_changes, same_shot, viewpoint_change, subject_consistent, reliability`

Figure 16: Minimal metadata schema used to bind summaries, keyframes, tool evidence, and transitions.

D Instruction Templates and Examples

D.1 Instruction-Data Synthesis Templates

This subsection gives the detailed templates used to generate task instructions.

Fine-Grained Perception. This template produces single-image perception questions, answered as yes/no or multiple choice, to test recognition of visible entities, attributes, positions, text, counts, and local details.

```
{
  "image_pos": <Frame[ID]>,
  "category": "presence|attribute|spatial|text|count",
  "question": "<single-image yes/no or multiple-choice question>",
  "options": ["<A>", "<B>", "<C>", "<D>"],
  "answer": "<Yes|No|A|B|C|D>",
  "evidence_refs": ["Frame[ID]_Caption", "Frame[ID]_Tool[ID]_Evidence"]
}
```

Visual Reasoning. This template produces single-image visual reasoning questions, answered as open-ended QA, to test inference from concrete visual cues. The answer gives a concise conclusion followed by concrete visual cues.

```
{
  "image_pos": <Frame[ID]>,
```

Table 9: Rules used to verbalize tool signals and score each frame.

Tool evidence	Verbalization rule	Value-scoring use	Reliability boundary
OCR	Use recognized textual content as primary evidence for readable text; describe the visible meaning, function, or role of the text rather than enumerating all detected tokens.	Reward readable and specific text evidence that reveals task-relevant titles, steps, labels, states, commands, or interface cues; assign higher value when text evidence substantially improves understanding beyond generic visual appearance.	Keep noisy, incomplete, or ambiguous text evidence conservative; do not reward repeated boilerplate, recognition artifacts, or isolated fragments unless they reveal important state or intent.
ASR	Use nearby spoken content as temporal contextual evidence; treat speech as information associated with the frame interval, rather than as on-screen text or direct visual evidence.	Reward specific and temporally relevant speech that contributes concrete instructions, explanations, actions, or contextual meaning; assign higher value when spoken content matches the apparent step, state, or action around the frame.	Stay conservative when the alignment between speech and frame is uncertain; do not over-reward vague transcripts, weakly aligned speech, or information already evident from the visual content alone.
YOLOv8	Use detected object categories as primary evidence for visible entities; use confidence or location only as supporting information, and describe only entities that are visually supported.	Reward clear and relevant visible entities that explain the subject, objects in use, workspace context, or interaction target; assign higher value when detected entities are specific, diverse, and informative.	Do not treat detection confidence, quantity, or coverage alone as evidence of value, and do not infer unsupported relationships, intentions, or actions.
RAFT	Use optical-flow evidence as primary support for motion or temporal change; interpret motion strength as an indicator of change intensity rather than semantic importance or action type.	Reward motion evidence when it marks a meaningful transition, interaction moment, or state change; assign higher value when motion helps distinguish the frame from static or redundant neighboring frames.	Describe staticness or motion-heavy appearance only when supported by motion evidence; do not reward motion magnitude alone without semantic relevance.
OmniParser	Use parsed interface structure and element descriptions as primary evidence for GUI layout, component meaning, and visible application state; prioritize actionable or state-revealing elements over low-level visual labels.	Reward actionable interface structure such as buttons, menus, form fields, toggles, tabs, or clear application states; assign higher value when parsed elements materially improve understanding of what the interface shows or allows.	Treat element counts, positions, and bounding regions as supporting evidence rather than meaning by themselves; do not reward dense but uninformative interface clutter.

```
  "category": "reasoning|function|activity|relation",
  "question": "<why/how/what can be inferred from the image>",
  "answer": "<2-4 sentences grounded in visible cues>",
  "evidence_refs": ["Frame[ID]_Caption", "Frame[ID]_Tool[ID]_Evidence"]
}
```

Multi-Image Logic. This template produces multi-image, multi-turn logic questions, answered through ordered dialogue, to test cross-image comparison, verification, and evidence binding. It asks the model to compare, connect, or jointly verify facts across images while maintaining strict role order and evidence grounding.

```
{
  "image_pos_list": [<Frame[ID]>, <Frame[ID]>, ...],
  "turns": [
    {
      "role": "user",
      "text": "Goal: <cross-image verification goal>.
       Plan: Step 1: <inspect Image1>.
       Step 2: <inspect Image2>.
       Step 3: <compare them>.
       Images: Image1:<ImageHere> Image2:<ImageHere>.
       Step 1: <question>",
      "evidence_refs": ["Frame[ID]_Caption, ..."]
    },
    {
      "role": "assistant",
      "text": "<answer grounded in the referenced image>",
      "evidence_refs": ["Frame[ID]_Caption", ...]
    },
    {
      "role": "user",
      "text": "Step 2: <pairwise comparison question>",
      "evidence_refs": ["Frame[ID]_Caption", "Frame[ID]_Caption", ...]
    },
    {
      "role": "assistant",
      "text": "<cross-image answer>",
      "evidence_refs": ["Frame[ID]_Caption", "Frame[ID]_Caption", "Video_Summary", ...]
    }
  ]
}
```

Editing This template produces image editing instructions from a source–target pair, answered by generating the edited image, to test localized and controllable visual generation. The change must be small and controllable, preserve stable scene content, and be expressible as one or two atomic edits.

```
{
  "from_pos": <source Frame[ID]>,
  "to_pos": <target Frame[ID]>,
  "ok": true,
  "edit_count": 1|2,
  "edits": ["<atomic local edit>", ...],
  "instruction": "<local edit or infilling instruction>.
                 Do not change anything else.",
  "reason": "<why the pair is small and controllable>",
  "evidence_refs": ["Transition[ID]", "Frame[ID]_Caption", "Frame[ID]_Caption"]
}
```

Grounded Edit-to-Select Generation. This template produces a two-image grounded editing task from a source image and a target image. The answer is a generated target image followed by a single-choice candidate image selection. It evaluates whether the model can preserve stable visual invariants, identify grounded visible edits, generate the target visual state, and select the candidate image matching the intended edited result.

```
{
  "Round_Plan": {
    "question": {
      "text": "What should be changed from the source image to reach the tar
      -get state?",
      "Frames": ["Frame[SOURCE_ID]", "Frame[TARGET_ID]"]
    },
    "answer": {
      "goal": "<objective target visual state>",
      "invariants": ["<stable entity/background>", ...],
      "edits": ["<grounded visible change>", ...]
```

```
    }
  },
  "Round_Gen": {
    "question": {
      "text": "Generate the target image after applying the grounded edit.",
      "Frames": ["Frame[SOURCE_ID]"]
    },
    "answer": {
      "image": "<generated target image>"
    }
  },
  "Round_Select": {
    "question": {
      "text": "Which candidate image best matches the generated target image?",
      "Frames": ["Frame[CANDIDATE_ID]", "Frame[CANDIDATE_ID]", ...]
    },
    "answer": "target:Frame[ID]",
    "rationale": "<why the candidate matches the generated target image>"
  },
  "evidence_refs": ["Frame[ID]_Caption", "Transition[ID]", ...]
}
```

Interleaved Understand–Recall–Generate–Select. This template produces a two-image visual understanding and recall task. The answer contains an open-ended QA response, a generated later-state image, and a single-choice candidate image selection. It evaluates whether the model can extract visible facts, reason about earlier-to-later changes, generate the recalled later state, and select the candidate frame consistent with that generated state.

```
{
  "Round_Understand": {
    "question": {
      "text": "What visible facts describe the earlier and later image states?",
      "Frames": ["Frame[EARLIER_ID]", "Frame[LATER_ID]"]
    },
    "answer": {
      "facts": ["<visible earlier-state fact>", "<visible later-state fact>"]
    }
  },
  "Round_Plan": {
    "question": {
      "text": "What grounded visual changes connect the earlier state to the later state?",
      "Frames": ["Frame[EARLIER_ID]", "Frame[LATER_ID]"]
    },
    "answer": {
      "edits": ["<grounded earlier-to-later change>", ...]
    }
  },
  "Round_QA": {
    "question": {
      "text": "<open question about the recalled later detail>",
      "Frames": ["Frame[EARLIER_ID]"]
    },
    "answer": "<grounded later-state answer>",
    "rubric": ["<answer constraint>", ...]
  },
  "Round_Gen": {
    "question": {
      "text": "Generate the recalled later-state image.",
      "Frames": ["Frame[EARLIER_ID]"]
    },
    "answer": {
      "image": "<generated later-state image >"
    }
  },
  "Round_Select": {
    "question": {
      "text": "Which candidate frame matches the generated later-state image?",
      "Frames": ["Frame[CANDIDATE_ID]", "Frame[CANDIDATE_ID]", ...]
    },
    "answer": "target:Frame[ID]",
    "rationale": "<why the selected candidate matches the generated later state>"
  },
  "evidence_refs": ["Frame[ID]_Caption", "Transition[ID]", ...]
}
```

Stepwise Sequence Generate-and-Select. This template produces a multi-image step reasoning task over a coherent visual sequence. The answer is a list of generated step-state image followed by

single-choice selections for each step. It evaluates whether the model can decompose a visual process into grounded steps, generate each intermediate state, track temporal order, and select the correct candidate frame for each step.

```
{
  "Round_Overall": {
    "question": {
      "text": "What is the overall coherent visual sequence?",
      "Frames": ["Frame[START_ID]", "Frame[END_ID]"]
    },
    "answer": "<overall description of the visual sequence>"
  },
  "Round_Steps": {
    "question": {
      "text": "What are the grounded step-by-step state changes?",
      "Frames": ["Frame[STEP_0]", "Frame[STEP_1]", "Frame[STEP_2]", ...]
    },
    "answer": [
      "<step 1 grounded transition>",
      "<step 2 grounded transition>"
    ]
  },
  "Round_Gen": [
    {
      "question": {
        "text": "Generate the target image for step 1.",
        "Frames": ["Frame[STEP_0]"]
      },
      "answer": {
        "image": "<generated step-1 image >"
      }
    },
    {
      "question": {
        "text": "Generate the target image for step 2.",
        "Frames": ["Frame[STEP_1]"]
      },
      "answer": {
        "image": "<generated step-2 image >"
      }
    }
  ],
  "Round_Select": [
    {
      "question": {
        "text": "Which candidate frame matches the generated image for step 1?",
        "Frames": ["Frame[CANDIDATE_1]", "Frame[CANDIDATE_2]", ...]
      },
      "answer": "target:Frame[ID]",
      "rationale": "<why this frame is the correct step-1 state>"
    },
    {
      "question": {
        "text": "Which candidate frame matches the generated image for step 2?",
        "Frames": ["Frame[CANDIDATE_1]", "Frame[CANDIDATE_2]", ...]
      },
      "answer": "target:Frame[ID]",
      "rationale": "<why this frame is the correct step-2 state>"
    }
  ],
  "evidence_refs": ["Frame[ID]_Caption", "Transition[ID]", ...]
}
```

Middle-Step Gap Generate-and-Fill. This template produces a three-image temporal gap-filling task. The answer is a generated middle-state image followed by a single-choice selection of the real middle candidate frame. It evaluates whether the model can reason bidirectionally from both adjacent transitions, generate the visual state that fits between two endpoints, and select the candidate satisfying both sides of the sequence.

```
{
  "Round_Explain": {
    "question": {
      "text": "What visual state should fit between the earlier and later frames?",
      "Frames": ["Frame[EARLIER_ID]", "Frame[LATER_ID]"]
    },
    "answer": "<why the middle state fits between earlier and later frames>"
  },
  "Round_Gen": {
```

```
    "question": {
      "text": "Generate the middle-step image that fits between the earlier and later frames.",
      "Frames": ["Frame[EARLIER_ID]", "Frame[LATER_ID]"]
    },
    "answer": {
      "image": "<generated middle-state image >"
    }
  },
  "Round_Select": {
    "question": {
      "text": "Which candidate frame matches the generated middle-step image?",
      "Frames": ["Frame[CANDIDATE_1]", "Frame[CANDIDATE_2]", ...]
    },
    "answer": "target:Frame[ID]",
    "rationale": "<why the selected frame matches both adjacent transitions>"
  },
  "evidence_refs": ["Frame[A]_Caption", "Frame[B]_Caption", "Frame[C]_Caption",
                    "Transition[ID]", ...]
}
```

Instructional Sequential Generate-and-Select. This template produces a multi-image future-state instruction generation task from context frames and future target frames. The answer combines generated edit instructions, generated future-state image, and per-step single-choice candidate selections. It evaluates whether the model can infer immediate future visual changes from context, generate each future state, define visual checks, and select the matching future frame.

```
{
  "Round_Overall": {
    "question": {
      "text": "What future visual sequence should follow from the context frames?",
      "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
    },
    "answer": "<overall future-step instruction from the context>"
  },
  "Round_Edit": {
    "question": {
      "text": "What visible edit should be made at each future step?",
      "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
    },
    "answer": ["<step 1 edit instruction>", "<step 2 edit instruction>"]
  },
  "Round_Reason": {
    "question": {
      "text": "Why does each future step follow from the context or previous state?",
      "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
    },
    "answer": ["<reason for step 1>", "<reason for step 2>"]
  },
  "Round_Target": {
    "question": {
      "text": "What visual state should each step reach?",
      "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
    },
    "answer": ["<desired visual state after step 1>", "<desired visual state after step 2>"]
  },
  "Round_Check": {
    "question": {
      "text": "How can each generated future step be visually verified?",
      "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
    },
    "answer": ["<visible condition for step 1>", "<visible condition for step 2>"]
  },
  "Round_Gen": [
    {
      "question": {
        "text": "Generate the target image for future step 1.",
        "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]"]
      },
      "answer": {
        "image": "<generated future-step-1 image>"
      }
    },
    {
      "question": {
        "text": "Generate the target image for future step 2.",
        "Frames": ["Frame[CONTEXT_1]", "Frame[CONTEXT_2]", "Frame[FUTURE_1]"]
      },
      "answer": {
```

```
        "image": "<generated future-step-2 image>"
      }
    }
  ],
  "Round_Select": [
    {
      "question": {
        "text": "Which candidate frame matches the generated image for future step 1?",
        "Frames": ["Frame[CANDIDATE_1]", "Frame[CANDIDATE_2]", ...]
      },
      "answer": "target:Frame[ID]",
      "rationale": "<why this frame is the immediate next future state>"
    },
    {
      "question": {
        "text": "Which candidate frame matches the generated image for future step 2?",
        "Frames": ["Frame[CANDIDATE_1]", "Frame[CANDIDATE_2]", ...]
      },
      "answer": "target:Frame[ID]",
      "rationale": "<why this frame matches the next future state>"
    }
  ],
  "evidence_refs": ["Frame[ID]_Caption", "Transition[ID]", ...]
}
```

D.2 Examples

This section gives one example for each template.

Instruction-Data Example (Fine-Grained Perception)



Input Image

Question: Which object is visible in the foreground? A. Plate. B. Coffee cup. C. Phone. D. Book.

Answer: B. Coffee cup.

Figure 17: Fine-grained perception example. The sample asks the model to identify a foreground object from a selected visual frame.

Instruction-Data Example (Visual Reasoning)



Input Image

Question: What can be inferred about the activity in the image?

Answer: Someone is cooking food over heat. The food is positioned above a cooking surface and the visible setup supports an ongoing preparation process.

Figure 18: Visual reasoning example. The sample requires inferring the ongoing activity from visual evidence in the image.

Instruction-Data Example (Multi-Image Logic)



Image 1



Image 2



Image 3

Round 1: Inspect

Question: Do the first two images show the same street environment?

Answer: Yes. Both show a city street with moving traffic and similar road-side structures.

Round 2: Compare

Question: What changes across the images?

Answer: The positions of vehicles and pedestrians shift along the street while the road layout remains stable.

Round 3: Verify

Question: Is the comparison supported by all images?

Answer: Yes. The shared street layout binds the images together, and the moving road users provide the changing evidence.

Figure 19: Multi-image logic example. The sample compares multiple related frames and verifies the shared scene structure and temporal changes.

Instruction-Data Example (Editing)

Question:



Source

Instruction: Edit the source image so the woman's facial expression changes from neutral/speaking to smiling. Do not change anything else.

Answer:



Figure 20: Image editing example. The sample edits the source image according to a localized visual instruction while preserving unrelated content.

Instruction-Data Example (Grounded Edit-to-Select Generation)

Round 1: Plan

Question: What should be changed from the source image to reach the target state?



Source



Target

Answer: Keep the same Newton's cradle setup, but transfer the motion from the left displaced ball to the right displaced ball after the collision.

Round 2: Generate

Question: Generate the target image after applying the grounded edit.



Source

Answer:



Round 3: Select

Question: Which candidate image best matches the generated target image?



Candidate A



Candidate B



Candidate C

Answer: Candidate B. It best matches the later state where the right metal ball swings outward while preserving the same cradle and floor background.

Figure 21: Grounded edit-to-select generation example. The sample plans a grounded visual edit, generates the target state, and selects the matching candidate.

Instruction-Data Example (Interleaved Understand–Recall–Generate–Select)

Round 1: Understand

Question: What visible facts describe the earlier and later image states?



Earlier



Later

Answer: The earlier image shows food and hands around a table. The later image shows a clay character in the same warm stop-motion style, now framed as an upper-body view.

Round 2: Recall

Question: What grounded visual change connects the two states?

Answer: The scene changes from a food-focused tabletop view to a character-focused upper-body view.

Round 3: Generate

Question: Generate the recalled later-state image.



Earlier

Answer:



Round 4: Select

Question: Which candidate matches the generated later-state image?



Candidate A



Candidate B



Candidate C

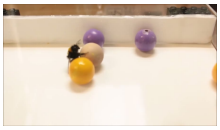
Answer: Candidate C. It matches the recalled later character-focused view.

Figure 22: Interleaved understand-recall-generate-select example. The sample alternates between visual understanding, grounded recall, generation, and candidate selection.

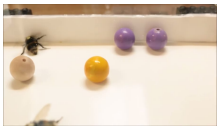
Instruction-Data Example (Stepwise Sequence Generate-and-Select)



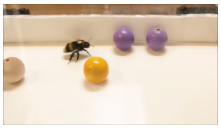
Start



Step



Step



End

Round 1: Overall

Question: What is the overall coherent visual sequence?

Answer: The sequence shows a bumblebee approaching a small ball, contacting it, and rolling it along the surface.

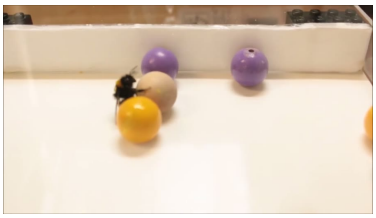
Round 2: Steps

Question: What are the grounded step-by-step state changes?

Answer: Step 1: the bee approaches the ball. Step 2: it touches and grips the ball. Step 3: it rolls the ball forward.

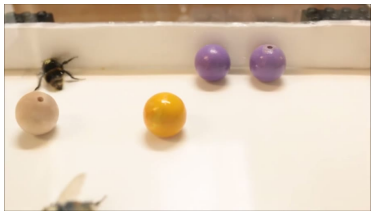
Round 3: Generate

Question: Generate the next real action frame in which the bee is rolling the ball.



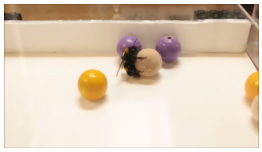
Current

Answer:

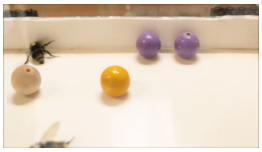


Round 4: Select

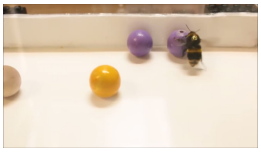
Question: Which candidate matches the generated step image?



Candidate A



Candidate B



Candidate C

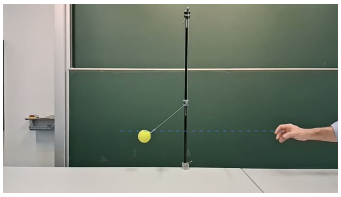
Answer: Candidate B. It matches the bee rolling the ball in the next visible action state.

Figure 23: Stepwise sequence generate-and-select example. The sample follows a continuous action sequence, generates the next step, and selects the matching candidate.

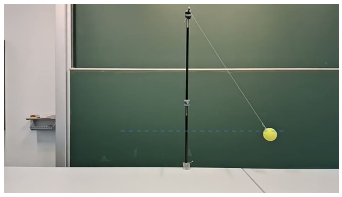
Instruction-Data Example (Middle-Step Gap Generate-and-Fill)

Round 1: Explain

Question: What visual state should fit between the earlier and later images?



Earlier

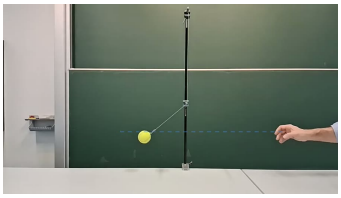


Later

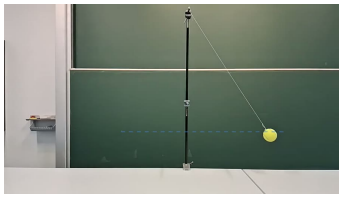
Answer: The middle state should show the pendulum between the earlier left-side position and the later right-side position.

Round 2: Generate

Question: Generate the middle-step image that fits between the endpoints.

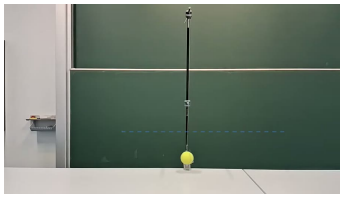


Earlier



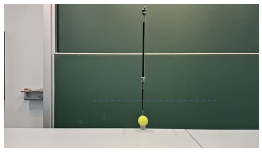
Later

Answer:

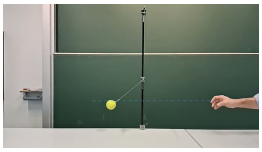


Round 3: Select

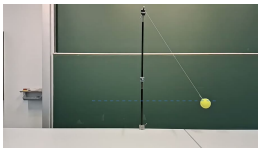
Question: Which candidate matches the generated middle-step image?



Candidate A



Candidate B



Candidate C

Answer: Candidate A. It shows the pendulum passing through the intermediate position between the earlier and later swing states.

Figure 24: Middle-step gap generate-and-fill example. The sample infers and generates the missing intermediate frame between two temporally ordered endpoints.

Instruction-Data Example (Instructional Sequential Generate-and-Select)

Round 1: Overall

Question: What future visual sequence should follow from the context images?



Context 1



Context 2

Answer: The flower-blooming process should continue through three visible states in the same outdoor close-up: the petals open wider, the bloom expands, and the flower reaches a fuller open state.

Round 2: Edit

Question: What visible edit should be made at each future step?

Answer: Step 1: open the petals slightly farther. Step 2: continue unfolding the red petals. Step 3: show the flower in a fuller open bloom.

Round 3: Reason

Question: Why do these future steps follow from the context or previous state?

Answer: The context shows the same flower in a stable natural scene, so the future images should advance the blooming state while preserving the red flower and green-leaf composition.

Round 4: Generate

Question: Generate the three future target images that follow from the context.



Context 1



Context 2

Answer:







Round 5: Select

Question: Which three candidate images match the generated future sequence?



A1



A2



A3



B1



B2



B3



C1



C2



C3

Answer: Select A1, A2, and A3. They match the generated blooming sequence: petals open wider, the flower expands, and the bloom reaches a fuller open state.

Figure 25: Instructional sequential generate-and-select example. The sample follows a stable natural blooming process, generates future visual steps, and selects the matching sequence.

E Quality Control and Filtering Rules

E.1 Gate A Prompt

The detailed Gate A prompt is shown in Fig. 26.

Shot Admission Prompt

```
You are a strict shot-admission judge for a video frame selection pipeline.
Judge whether the current shot is worth sending to downstream frame sampling.
Keep shots only if they are visually usable and likely to yield informative frames.
Reject shots that are mainly low-information filler, weak transitions, visually poor,
or not worth spending frame-sampling budget on.
Return strict JSON with keys usable_score, quality_score, informativeness_score,
decision, reason, evidence.
INPUTS:
Video summary: <video_summary>
Video type: <video_type>
Chapter role: <chapter_role>
Chapter summary: <chapter_summary>
Shot summary: <shot_summary>
Shot role: <shot_role>
Scene type: <scene_type>
Visual quality hint: <visual_quality_hint>
Informativeness hint: <informativeness_hint>
Sampling hint: <candidate_sampling_hint>
Representative frame count: <representative_frame_count>
OUTPUT FORMAT (strict JSON only, no extra text):
{
"usable_score": float,
"quality_score": float,
"informativeness_score": float,
"decision": str,
"reason": str,
"evidence": str
}
```

Figure 26: Prompt for shot admission.

E.2 Gate B Rules & Prompt

After sampling, candidate frames enter a two-stage technical quality filter. The fast stage uses OpenCV statistics, including Laplacian variance, brightness, grayscale standard deviation, and histogram entropy, to reject frames with obvious blur, extreme exposure, low contrast, or low informativeness. The model stage uses VLM to inspect higher-level visual usability, The detailed Gate B prompt is shown in Fig. 27.

Quality Assessment Prompt

```
You are a strict technical image quality inspector for dataset filtering.

Judge only technical quality and cleanliness of the visible image, not aesthetics,
narrative value, or semantic importance.  Inspect the image using domain-appropriate
standards for real photos, animation, game renders, screen/UI frames, and mixed cases.
Do not treat intentional stylization, high/low-key lighting, bloom, flat shading, or
non-photorealism as defects unless they harm usability.

Evaluate only meaningful regions, including important subjects, informative structures,
readable text, interfaces, faces, hands, and products.  Borderline, ambiguous, or
mildly degraded cases should fail under strict filtering.  For each dimension, output
pass or fail with one short reason, and generate a 1-2 sentence factual caption for
downstream frame-value scoring.

QUALITY DIMENSIONS:

1.  exposure_brightness

2.  focus_sharpness

3.  motion_blur_camera_shake_ghosting

4.  noise_grain_low_light_degradation

5.  artifacts_compression_resampling_ai_generated

6.  color_fidelity_white_balance_color_cast

7.  contrast_dynamic_range_detail_visibility

8.  occlusion_obstruction_crop_integrity

9.  corruption_abnormal_regions_overlay_contamination

OUTPUT FORMAT:

- First output exactly 9 plain-text lines in the above order.

- Each line must contain the dimension name, pass or fail, and one short reason.

- Then output exactly one JSON object and nothing else.

- Do not use markdown fences.

- Do not output the inferred image domain.

- In the JSON, caption must contain the factual caption, and all quality fields must be
exactly pass or fail.

{

caption":  "1-2 short factual English caption sentences describing the visible scene.

exposure_brightness":  "pass

focus_sharpness":  "pass

motion_blur_camera_shake_ghosting":  "pass

noise_grain_low_light_degradation":  "pass

artifacts_compression_resampling_ai_generated":  "pass

color_fidelity_white_balance_color_cast":  "pass

contrast_dynamic_range_detail_visibility":  "pass

occlusion_obstruction_crop_integrity":  "pass

corruption_abnormal_regions_overlay_contamination":  "pass

}
```

Figure 27: Prompt for quality assessment.

E.3 Facility-location Greedy Selection

We use standard weighted facility-location greedy selection to choose representative frames within each shot. For each shot, let the candidate-frame set be $F = \{f_i\}_{i=1}^{n}$, and let k denote the selection budget assigned to this shot by the quota planner. The goal of RepresentativeSelector is to select a representative subset S from the candidate set under the constraint $|S| \leq k$, so that the selected frames cover the important candidates within the shot as much as possible. The objective function is defined as:

$$\mathcal{F}(S) = \sum_{i} r_i \max_{j \in S} \mathrm{sim}(i, j),$$

where r_i denotes the quality weight of candidate frame f_i, and $\mathrm{sim}(i, j)$ denotes the similarity between candidate frame f_i and the selected representative frame f_j.

The candidate-frame quality weight is computed from the preceding semantic value assessment:

$$r_i = \mathrm{value_score}_i.$$

The value score $\mathrm{value_score}_i$ is produced by the agent based on semanticized tool signals, while the inter-frame similarity $\mathrm{sim}(i, j)$ is computed from the semantic representations of candidate frames. Specifically, the system first extracts image embeddings and caption embeddings, normalizes the visual and textual semantic representations, concatenates them, and then computes cosine similarity:

$$\mathrm{sim}(i, j) = \cos(z_i, z_j).$$

During selection, the algorithm starts from an empty set $S = \emptyset$ and maintains the current coverage degree c_i of each candidate frame by the selected set. At each iteration, the algorithm traverses all unselected candidate frames f_j and computes the marginal gain of adding it to the selected set:

$$\Delta(j) = \sum_{i} r_i \max\big(0, \mathrm{sim}(i, j) - c_i\big).$$

The candidate frame with the largest marginal gain is added to S, and the coverage degree is updated as:

$$c_i \leftarrow \max\big(c_i, \mathrm{sim}(i, j)\big).$$

This process is repeated until the shot-level budget k is reached or all candidate frames have been exhausted.

E.4 Signal Filtering Prompt

The detailed signal filtering prompt is shown in Fig. 28.

E.5 Gate C

The detailed Gate C prompt is shown in Fig. 29.

Signal Filtering Prompt

```
You are a precise semantic filtering assistant.

Given a list of semantic tool signals and a shot summary, filter the signals according
to their relevance to the shot.

Keep only signals that are clearly supported by, relevant to, or useful for
understanding the shot summary.  Remove signals that are irrelevant, redundant, too
vague, unsupported, inconsistent, or only weakly related.  Be conservative:  if
relevance is uncertain, remove the signal.

FILTERING RULES:

- Keep signals grounded in the shot summary.

- Keep signals that help interpret actions, objects, scene context, intent, spatial
relations, or temporal events.

- Remove signals not mentioned or implied by the shot summary.

- Remove generic signals when a more specific retained signal covers the same meaning.

- Remove contradictory signals.

- Do not invent new signals.

- Do not modify original signal wording except for minor normalization.

INPUTS:

Semantic tool signals:

{{semantic_tool_signals}}

Shot summary:

{{shot_summary}}

OUTPUT FORMAT (strict JSON only, no extra text):

{

"filtered_signals":  [

{

signal":  "<kept semantic tool signal>

reason":  "<brief reason for keeping it>

}

],

"removed_signals":  [

{

signal":  "<removed semantic tool signal>

reason":  "<brief reason for removing it>

}

]

}
```

Figure 28: Prompt for signal filtering.

Metadata Admission Prompt

```
You are a strict metadata-level admission judge for a video-to-dataset generation
pipeline.

Decide whether the provided metadata package is suitable for downstream dataset
instance generation.  Judge the metadata as a whole, using video-level summaries,
chapter summaries, shot summaries, frame captions, semanticized tool signals, OCR, ASR,
object, motion, UI, transition, and pairwise-change evidence when available.

Do NOT judge raw image quality directly unless quality problems are explicitly
reflected in the metadata.  Judge whether the metadata contains enough grounded,
coherent, specific, and useful information to support reliable dataset generation.

ADMISSION GOAL:

Accept metadata only if it can support meaningful grounded dataset items, such as
visual QA, multi-choice QA, reasoning QA, dialogue, temporal/change reasoning, or edit
instructions.  Reject metadata that is too weak, generic, noisy, contradictory,
redundant, or poorly grounded.

DECISION LABELS:

- accept:  suitable for downstream dataset generation.

- reject:  unsuitable for downstream dataset generation.

JUDGMENT CRITERIA:

1.  Grounded visual content:  keep metadata with concrete visible entities, actions,
states, layouts, text, UI elements, scene structure, or temporal changes; reject
generic, empty, vague, or repetitive metadata.

2.  Cross-level coherence:  keep metadata whose video-, chapter-, shot-, and
frame-level descriptions are broadly consistent; reject conflicting evidence that
weakens grounding.

3.  Dataset usefulness:  keep metadata that supports at least one useful task type;
reject metadata that only yields trivial, obvious, or hallucination-prone items.

4.  Evidence density:  keep metadata with enough distinct evidence points across
frames or tools; reject sparse, duplicated, or semantically flat evidence.

5.  Tool-signal reliability:  treat tool signals as auxiliary evidence; keep them when
they reinforce captions or summaries; reject incoherent, failed, irrelevant, or
contradictory tool evidence.

6.  Temporal and structural value:  keep meaningful progression, state changes,
interactions, procedures, spatial layouts, readable text, UI structure, or object
relations; reject filler, static background, near-duplicates, or low-information
motion.

7.  Hallucination safety:  reject metadata that would require inventing unseen details,
intentions, hidden causes, identities, exact text, or precise attributes.

OUTPUT FORMAT (strict JSON only, no extra text):

{

decision":  "accept" | "reject

"primary_reasons":  [string, ...]

}

FIELD RULES:

- primary_reasons must contain 2-4 concise reasons for the decision.
```

Figure 29: Prompt for metadata admission.

To quantify the practical filtering effect of Gate C, we randomly sampled 5,000 metadata entries before filtering. Gate C rejected 37.4% of the entries overall, while the domain-level rejection rates are summarized in Table 10. The primary rejection reasons included unsupported metadata content, noisy textual or tool-recognition signals, generic descriptions with sparse usable information, repetitive imagery with low evidence density, and inaccurate state-transition descriptions across the five domains, respectively.

Table 10: Gate C rejection rates by domain.

Domain	Rejection Rate (%)
Technology and Tutorial	37.3
Sports and Action	39.8
Daily Life and Human Activities	24.8
Nature and Scenery	47.3
Gaming and ACG	38.2
Overall	37.4

F Dataset Taxonomy Details

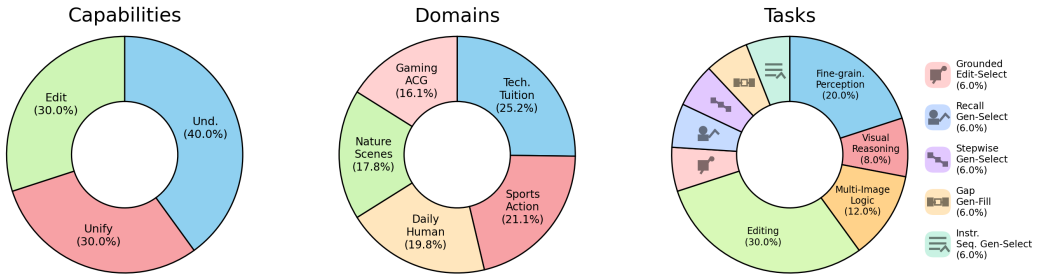


Figure 30: Distribution of the final dataset.

This section provides detailed definitions of the capability, domain, and task categories used to analyze VidUEU-100K.

F.1 Capability Categories

To improve the overall capabilities of multimodal large models, we categorize the 100K instruction-tuning samples into three capability paradigms: understanding, editing, and unified.

- **Understanding:** Focuses on multimodal comprehension tasks, including fine-grained visual perception and complex reasoning over visual content.
- **Editing:** Focuses on instruction-based image editing, requiring models to modify or complete visual content according to textual instructions while preserving irrelevant visual details.
- **Unified:** Involves interleaved text–image sequences, requiring the model to jointly perform multimodal understanding and generation capabilities within a unified stream.

F.2 Domain Categories

To ensure a balanced representation of both naturalistic and digitally synthesized environments, we classify the dataset into five thematic domains based on contextual semantics. Within each domain, we maximize content diversity to mitigate representation bias and ensure broad generalization.

- **Technology & Tutorial:** This category includes instructional demonstrations, software interfaces, scientific knowledge, and procedural workflows.

- **Sports & Action:** This category covers athletic competitions, physical training, outdoor sports, and various dynamic action sequences.
- **Daily Life & Human Activities:** This category encompasses routine human activities, including domestic tasks, commuting, and social interactions.
- **Nature & Scenery:** This category includes natural landscapes, meteorological phenomena, and various flora and fauna.
- **Gaming & ACG:** This category consists of gameplay recordings, digital animations, comics, and interactive interface content.

F.3 Task Categories

Based on the generated metadata, the agent adaptively determines and generates nine task types. The first three tasks belong to Understanding, the fourth corresponds to Editing, and the final five fall under Unified.

- **Fine-grained Perception:** Requires the identification of microscopic visual details, including object attributes, existence, cardinality, and precise spatial arrangements within visual scenes.
- **Visual Reasoning:** Requires inferring logical dependencies, causal relationships, and relational dynamics between subjects across visual contexts.
- **Multi-Image Logic:** Requires maintaining coherent co-reference and logical consistency across a sequence of incrementally introduced images.
- **Editing:** Requires completing missing visual content or editing existing images according to textual instructions and provided visual context.
- **Grounded Edit-to-Select Generation:** Requires generating a target visual state from grounded source–target changes and selecting the candidate image that matches the intended edit.
- **Interleaved Understand–Recall–Generate–Select:** Requires extracting visible facts from earlier visual context, recalling a later state, generating the recalled visual state, and selecting the matching candidate frame.
- **Stepwise Sequence Generate-and-Select:** Requires decomposing a coherent visual process into ordered steps, generating intermediate visual states, and selecting candidates that match each step.
- **Middle-Step Gap Generate-and-Fill:** Requires reasoning over earlier and later frames to infer, generate, and select the missing middle visual state.
- **Instructional Sequential Generate-and-Select:** Requires inferring future visual changes from context frames, generating stepwise future states, and selecting candidates that satisfy each predicted step.

F.4 Data Statistics

The VidUEU dataset contains 100,000 instruction samples, comprising 40,000 understanding samples, 30,000 editing samples, and 30,000 unified interleaved multimodal generation samples. The dataset is derived from 8,268 raw videos, covering approximately 224.2 hours of measurable segment-level video duration and containing approximately 8,505 semantic segments. The detailed distributions of capabilities, domains, and task types are shown in Figure 30.

F.5 Intrinsic Data-Quality Audit

We randomly sampled 800 instances for human evaluation, with 200 instances per quality dimension. Two annotators independently rated each instance on a 0–5 scale. Table 11 reports the annotator scores, their means, and Cohen's κ for inter-annotator agreement. The first three dimensions were assessed at the metadata level, while editing-sample validity was evaluated at the individual editing-instruction level.

Table 11: Human audit of synthesized-data quality.

Quality Dimension	Annotator A	Annotator B	Mean	Cohen's κ
Gate C Metadata Validity	4.12	4.08	4.10	0.69
Keyframe–Tool–Signal Alignment	4.30	4.20	4.25	0.76
Transition Correctness	4.06	4.01	4.04	0.77
Editing-Sample Validity	3.96	4.02	3.99	0.71

G Training and Evaluation Details

G.1 Training Details

InternVL3-8B. For understanding, we fine-tune InternVL3-8B on the understanding subset. We freeze the vision encoder and apply LoRA fine-tuning to the remaining target modules. The model is trained for 2 epochs with AdamW, a learning rate of 5×10^{-5}, a global batch size of 32, and a cosine learning rate scheduler with a warmup ratio of 0.03. Training uses 2 NVIDIA A100-40G with DeepSpeed ZeRO stage 2, bfloat16 precision, and gradient checkpointing.

Bagel-7B-MoT. For understanding, editing and unified tasks, we fine-tune Bagel-7B-MoT on the understanding, editing and unified subsets, respectively. We freeze the VAE encoder, the VAE decoder, and the ViT encoder, apply LoRA to the LLM backbone with $r = 16$ and $\alpha = 32$. For editing and unified training, we additionally fully fine-tune the generation bridging heads, including `llm2vae`, `vae2llm`, and `time_embedder`. The model is trained for 1 epoch with AdamW, a learning rate of 5×10^{-5}, and a cosine learning rate scheduler, using cross-entropy loss for text supervision and MSE loss for image supervision. Training uses 8 A100-40GB GPUs.

G.2 Evaluation Details

We evaluate the fine-tuned models under three capability settings, namely understanding, editing, and unified understanding and generation. We follow the official protocol of each benchmark.

G.3 Compared Baselines

For comparison, we report results against representative understanding and unified baselines, including InternVL3 with other model sizes Zhu et al. [2025], Qwen2.5-VL Bai et al. [2025b], LLaMA-3.2 Grattafiori et al. [2024], Janus-Pro Chen et al. [2025d], and TokenFlow-XL Qu et al. [2024]. We also compare with editing and unified baselines, including Instruct-Pix2Pix Brooks et al. [2023], MagicBrush Zhang et al. [2024], AnyEdit Yu et al. [2024], Step1X-Edit Liu et al. [2025], Gemini-2.0-flash Kampf and Brichtova [2025], OmniGen Xiao et al. [2024], OmniGen2 Wu et al. [2025], Ovis-U1 Wang et al. [2025], and Bagel-Zebra Li et al. [2026].

G.4 Evaluation Protocols

Understanding evaluation. For LLaVA-Bench, we use `Qwen3.5-35B-A3B` Qwen Team [2026] as the VLM judge. For MMDU, we use `Qwen3-VL-235B-A22B-Instruct` Bai et al. [2025a] as the VLM judge.

Editing evaluation. The editing model is evaluated on GEdit-Bench-En. GEdit-Bench-En is used to evaluate instruction-based image editing, with reported dimensions including G_SC, G_PQ, G_O, and Average. We use `GPT-4.1` OpenAI [2025] as the judge for the G_SC, G_PQ, and G_O metrics.

Unified evaluation. The unified model is evaluated on WEAVEbench. We use `GPT-4o` OpenAI et al. [2024] as the VLM judge.

G.5 Evaluation Variability

Table 12 reports the run-to-run standard deviation of VLM-as-a-judge evaluations. For each entry, the evaluated model outputs are fixed, and the standard deviation is computed over three independent judge runs.

Table 12: Automatic-judge run-to-run standard deviation with fixed evaluated outputs. Mean scores are reported in the main experimental tables.

Capability	Model / metric	Baseline std	Ours std
Understanding	InternVL3, LLaVA-Bench	0.317	0.292
Understanding	InternVL3, MMDU	0.150	0.172
Understanding	Bagel, LLaVA-Bench	0.306	0.328
Understanding	Bagel, MMDU	0.133	0.151
Editing	Bagel, GEdit-Bench-En G_SC	0.073	0.157
Editing	Bagel, GEdit-Bench-En G_PQ	0.076	0.146
Editing	Bagel, GEdit-Bench-En G_O	0.072	0.134
Unified	Bagel, WEAVE Science	0.0106	0.0108
Unified	Bagel, WEAVE Creation	0.0053	0.0051
Unified	Bagel, WEAVE Logic	0.0119	0.0119
Unified	Bagel, WEAVE Game	0.0230	0.0217
Unified	Bagel, WEAVE Average	0.0129	0.0122

Table 13: Full results of autonomous data iteration using Bagel.

Setting	MMT-Bench↑	MME↑	LLaVA-Bench↑	MMDU↑	GEdit G_O↑	WEAVE Overall↑
Iteration 1	0.631	1725.4	52.3	29.7	6.541	0.448
Iteration 2	0.629	1767.3	52.3	29.2	6.796	0.452
Iteration 3	0.642	1840.4	56.1	31.4	6.873	0.458
Iteration 4	0.649	1871.9	57.2	32.8	6.901	0.499
Final (separate)	0.651	1900.2	57.9	33.2	6.951	0.501

H Full Results for Autonomous Iteration and Ablation

H.1 Full Results of Autonomous Data Iteration

Table 13 reports the full results for Iterations 1–4 under 30K joint training, together with the Final results from capability-specific training on VidUEU-100K. The Final joint-training results are reported in Table 4.

H.2 Training Protocol and Data Distributions

Iterations 1–4 each use a fixed budget of 30K samples. At each iteration, the agent randomly samples from the generated data according to the updated target proportions of capabilities, domains, and task types. Samples exceeding category quotas are removed, while additional samples are generated for underrepresented categories to form the 30K training set. Each iteration jointly fine-tunes BAGEL from the same initial checkpoint for one epoch on eight A100-40GB GPUs. The Final 100K dataset is constructed by scaling the target distribution obtained in Iteration 4.

The agent prioritizes the capability distribution, allowing deviations of at most one percentage point for domain and task-type distributions. The realized distributions across iterations are reported in Tables 14, 15, and 16. At each iteration, the agent receives training loss and learning curves, aggregate benchmark and subcategory scores, the current data distribution, and compressed historical feedback and distributions, with optional human-audit inputs. Baseline performance is provided only in the first iteration as a fixed reference, while the other feedback signals are updated after each round.

H.3 Feedback-Free Evaluation

MMBench, MMStar, and VSI-Bench are excluded from iterative feedback and used to assess generalization beyond the feedback benchmarks. As shown in Table 17, the results exhibit an overall upward trend across iterations.

Table 14: Realized capability distributions (%) across autonomous iterations.

Setting	Understanding	Editing	Unified
Iteration 1	50.0	25.0	25.0
Iteration 2	35.0	33.0	32.0
Iteration 3	39.0	31.0	30.0
Iteration 4	40.0	30.0	30.0

Table 15: Realized domain distributions (%) across autonomous iterations.

Setting	Technology	Sports	Daily Life	Nature	Gaming
Iteration 1	22.1	17.8	28.2	18.1	13.8
Iteration 2	18.1	21.9	24.8	16.1	19.1
Iteration 3	23.9	20.2	21.9	17.9	16.1
Iteration 4	25.2	21.1	19.8	17.8	16.1

Table 16: Realized task-type distributions (%) across autonomous iterations.

Setting	FP	VR	MIL	Edit	GES	RGS	SGS	GGF	ISG
Iteration 1	28.3	9.6	12.1	25.0	4.6	5.3	4.8	5.1	5.2
Iteration 2	22.4	6.8	5.8	33.0	9.0	6.4	6.1	4.8	5.7
Iteration 3	20.8	8.3	9.9	31.0	7.0	5.6	5.9	5.3	6.2
Iteration 4	19.8	8.3	11.9	30.0	5.6	6.3	5.8	6.2	6.1

Table 17: Evaluation on benchmarks excluded from autonomous feedback.

Setting	MMBench↑	MMStar↑	VSI-Bench↑
BAGEL Base	84.2	67.3	31.4
Iteration 1	84.9	67.4	32.8
Iteration 2	84.7	67.3	32.9
Iteration 3	85.0	67.5	33.6
Iteration 4	85.1	67.6	34.4
Final	**85.3**	**67.8**	**36.8**

H.4 Full Ablation Study

Table 18 reports the full ablation results of VidUEU-Agent components, including the additional understanding benchmarks omitted from the main table. The results show that removing any major component degrades performance, validating the effectiveness of the full pipeline design.

Given the computational demands of multi-stage data synthesis and model training, we evaluate six ablated variants using 10K samples per capability.

We retain the capability-specific Final models trained on the 100K dataset as the full-pipeline reference, enabling broader component coverage without repeated large-scale training.

These comparisons provide indicative evidence of component-level contributions under practical data budgets, while leaving strictly budget-matched evaluation for future investigation.

H.5 Comparison with Baseline Datasets

We compare Bagel under the same training setting using our dataset and three task-specific baseline datasets: LLaVA Visual Instruct 150K Liu et al. [2023] for understanding, InstructPix2Pix Brooks et al. [2023] for editing, and Zebra-CoT Li et al. [2026] for unified tasks, as shown in Table 19. Our dataset achieves the best results across all evaluated settings, indicating stronger overall training effectiveness.

Table 18: Full ablation study of VidUEU-Agent components using Bagel.

Setting	MMT-Bench↑	MME↑	LLaVA-Bench↑	MMDU↑	GEdit G_O↑	WEAVE Overall↑
Ours	0.651	1900.2	57.9	33.2	6.951	0.501
w/o A	0.638	1772.6	53.6	32.4	6.773	0.497
w/o B	0.605	1255.7	42.8	24.7	5.175	0.404
w/o S	0.623	1684.3	50.7	30.8	5.848	0.462
w/o C	0.617	1549.4	51.2	30.5	6.303	0.454
N	0.537	1116.4	38.7	22.0	5.041	0.393
w/o R	0.611	1565.5	51.0	29.4	6.031	0.470

Table 19: Comparison with baseline datasets using Bagel under the same training setting.

Dataset	MMT-Bench↑	MME↑	LLaVA-Bench↑	MMDU↑	GEdit G_O↑	WEAVE Overall↑
LLaVA Visual Instruct	0.477	1823.0	51.8	29.6	–	–
InstructPix2Pix	–	–	–	–	6.882	–
Zebra-CoT	–	–	–	–	–	0.449
Ours(Bagel)	**0.651**	**1900.2**	**57.9**	**33.2**	**6.951**	**0.501**

I Broader Impacts

VidUEU-Agent and VidUEU-100K may help reduce the cost of constructing high-quality multimodal instruction data and support the development of models with stronger understanding, editing, and unified capabilities. However, even with careful human screening, source videos may contain social biases, privacy-sensitive content, or copyright risks, which may still be inherited by the constructed dataset and downstream models. Therefore, future use of VidUEU-Agent should include careful data filtering, license review, privacy protection, and safeguards against malicious use.

J Limitations

Although VidUEU-Agent covers multimodal understanding, editing, and unified tasks, it does not explicitly construct pure text-to-image generation data. This is because the advantage of video sources lies in temporal continuity and state transitions; for pure text-to-image generation, which mainly requires isolated text–image alignment, directly curating image data would be more suitable and less costly than using videos as the primary source. In addition, due to limited computational and annotation resources, we scale the dataset to 100K instruction samples, which remains modest compared with web-scale multimodal corpora. A larger-scale version may further improve data diversity, long-tail coverage, and the stability of multimodal training.

K Existing Assets and Licenses

We systematically inventory the existing assets used in this work, covering source data assets, runtime tools/models, pretrained/fine-tuned model assets, and evaluation benchmarks.

K.1 Source Data Assets

OpenVid. We use OpenVid as source video data for VidUEU-100K, and VidUEU-100K may release keyframes or samples extracted or derived from this source; the license is CC-BY-4.0, the source URL is `https://huggingface.co/datasets/nkp37/OpenVid-1M`, and released OpenVid-derived visual samples retain OpenVid attribution, original source URLs, license metadata, and the terms of the underlying video sources.

Panda-70M. We use Panda-70M as internal source video data to expand real-world video coverage; it is governed by Snap Inc. custom non-commercial research, testing, and validation terms, the source URL is `https://github.com/snap-research/Panda-70M`, and we use it only for non-commercial academic research without redistributing Panda-70M raw videos.

Wikimedia. We use Wikimedia as source images and videos, and frame samples or derived samples may be released when permitted by the corresponding licenses; the licenses include public domain, CC0, CC-BY, and CC-BY-SA, the source URL is https://commons.wikimedia.org/wiki/Commons:Licensing

YouTube videos. We use only explicitly authorized or Creative Commons YouTube videos; the applicable terms include the Standard YouTube License and Creative Commons, the source URL is https://www.youtube.com/static?template=terms, and we retain the original video URL and license metadata without redistributing unauthorized YouTube raw videos as standalone assets.

FineGym. We use FineGym as domain source data for sports and action coverage; its license is CC BY-NC 4.0, the source URL is https://github.com/SDOlivia/FineGym, and we cite the original FineGym paper while using it only for non-commercial academic research.

BDD100K. We use BDD100K as domain source data for driving and road-scene coverage; it follows BDD100K download and data-use terms, the source URLs are https://github.com/ucbdrive/bdd100k and https://bdd-data.berkeley.edu/download.html, and we retain required citations and copyright notices.

CodeVidCL. We use CodeVidCL as domain source data for code/video or software-demonstration coverage; its license is MIT, the source URL is https://aclanthology.org/2025.evalmg-1.1/, and we cite the original paper while using it only for non-commercial academic research.

K.2 Baseline Training Datasets

LLaVA Visual Instruct 150K. We use LLaVA Visual Instruct 150K as an understanding-data baseline; its license is CC-BY-4.0, subject to applicable upstream terms, and its source URL is https://huggingface.co/datasets/liuhaotian/LLaVA-Instruct-150K.

InstructPix2Pix. We use the InstructPix2Pix editing dataset as an editing-data baseline; its source URL is https://github.com/timothybrooks/instruct-pix2pix. The dataset is subject to applicable original data terms.

Zebra-CoT. We use Zebra-CoT as a unified-task training-data baseline; its dataset license is CC-BY-NC-4.0, and its source URL is https://huggingface.co/datasets/multimodal-reasoning-lab/Zebra-CoT.

K.3 Runtime Tools, Models, and Libraries Used by VidUEU-Agent

We strictly comply with the licenses and relevant terms of all assets used, and use them solely for academic research.

DeepSeek API. We use the DeepSeek API as a text LLM service for chapter construction, tool planning, value scoring, task routing, and sample generation; it is governed by the DeepSeek API / Open Platform terms of use, the source URLs are https://api-docs.deepseek.com/api/deepseek-api and https://cdn.deepseek.com/policies/en-US/deepseek-terms-of-use.html.

Qwen3.5-9B. We use Qwen/Qwen3.5-9B as a locally deployed VLM for shot profiling, quality assessment, and transition modeling; its license is Apache-2.0, the source URL is https://huggingface.co/Qwen/Qwen3.5-9B

PySceneDetect. We use PySceneDetect for shot boundary detection; its license is BSD-3-Clause, the source URLs are https://www.scenedetect.com/copyright/ and https://github.com/Breakthrough/PySceneDetect

OpenCV. We use OpenCV for statistical quality filtering and image processing; its license is Apache-2.0, the source URL is https://opencv.org/license/,

PaddleOCR. We use PaddleOCR for reading subtitles, screen text, documents, and UI text in frames; its license is Apache-2.0, the source URL is https://github.com/PaddlePaddle/PaddleOCR/blob/main/LICENSE

Whisper large-v2. We use Whisper large-v2 for ASR to recover speech context near shots; its license is MIT, the source URL is https://github.com/openai/whisper/blob/main/LICENSE

Ultralytics YOLOv8. We use Ultralytics YOLOv8 for object detection and entity grounding; its license is AGPL-3.0, the source URLs are https://www.ultralytics.com/license and https://github.com/ultralytics/ultralytics, and we use yolov8n.pt

RAFT optical flow. We use RAFT optical flow for motion estimation and optical-flow analysis of motion intensity and state changes; its license is BSD-3-Clause, the source URL is https://github.com/princeton-vl/RAFT,

CLIP Radford et al. [2021]. We use CLIP image/text embeddings for representative selection and candidate-frame similarity computation; its license is MIT, the source URL is https://github.com/openai/CLIP/blob/main/LICENSE,

OmniParser. We use OmniParser for GUI parsing, UI element detection or captioning, and interface-state recovery; its license is AGPL-3.0 for the referenced icon-detection component, the source URLs are https://huggingface.co/microsoft/OmniParser/tree/main/icon_detect and https://huggingface.co/microsoft/OmniParser/blob/main/icon_detect/LICENSE

DeepSpeed. We use DeepSpeed as training infrastructure; its license is Apache-2.0, the source URL is https://github.com/microsoft/DeepSpeed

K.4 Pretrained and Fine-tuned Model Assets

InternVL3-8B. We use InternVL3-8B as the understanding baseline and fine-tuning initialization; its license is Apache-2.0, the source URL is https://huggingface.co/OpenGVLab/InternVL3-8B

BAGEL-7B-MoT. We use BAGEL-7B-MoT as the unified model baseline and fine-tuning initialization; its license is Apache-2.0, the source URL is https://huggingface.co/ByteDance-Seed/BAGEL-7B-MoT

K.5 Evaluation Benchmarks

Benchmark instances and instance-level predictions are not used for data generation or prompt construction. Only aggregate test-split scores from MMDU, GEdit-Bench-En, and WEAVEbench are used to guide data distribution updates.

MMT-Bench. We use MMT-Bench as an understanding evaluation benchmark; its license is CC-BY-4.0, the source URLs are https://huggingface.co/datasets/OpenGVLab/MMT-Bench and https://mmt-bench.github.io/

MME. We use MME as an understanding evaluation benchmark; its license is Apache-2.0, the source URLs are https://mme-benchmark.github.io/home_page.html and https://huggingface.co/datasets/lmms-lab/MME

LLaVA-Bench in the Wild. We use LLaVA-Bench in the Wild as an understanding evaluation benchmark; its license is CC-BY-4.0, the source URLs are https://huggingface.co/datasets/liuhaotian/llava-bench-in-the-wild/tree/main and https://llava-vl.github.io/

MMDU. We use MMDU as an understanding evaluation benchmark; its license is CC BY-NC 4.0, the source URL is https://proceedings.neurips.cc/paper_files/paper/2024/hash/1057053100de064a44286239724f7865-Abstract-Datasets_and_Benchmarks_Track.html

GEdit-Bench-En We use GEdit-Bench-En as an instruction-based image editing evaluation benchmark; its license is Apache-2.0, the source URL is https://github.com/stepfun-ai/Step1X-Edit,

WEAVEbench We use WEAVEbench as a unified understanding-generation evaluation benchmark; its license is Apache-2.0, the source URL is https://huggingface.co/datasets/WeiChow/WEAVE

MMBench. We use MMBench for feedback-free multimodal understanding evaluation; its dataset license is CC-BY-4.0, and the source URLs are https://github.com/open-compass/MMBench and https://huggingface.co/datasets/opencompass/MMBench.

MMStar. We use MMStar solely for benchmark evaluation, without training or redistribution; its official dataset does not specify an explicit license, and the source URLs are `https://huggingface.co/datasets/Lin-Chen/MMStar` and `https://github.com/MMStar-Benchmark/MMStar`.

VSI-Bench. We use VSI-Bench for feedback-free visual-spatial understanding evaluation; its dataset license is Apache-2.0, and the source URL is `https://huggingface.co/datasets/nyu-visionx/VSI-Bench`.

K.6 Release Statement

VidUEU-100K is planned to be released under CC-BY-4.0. This license applies to the metadata, annotations, prompts, task JSON files, and redistributable derived data generated by us; for samples containing upstream visual assets, users must still comply with the corresponding source-asset licenses, attribution requirements, ShareAlike requirements, and terms of service. The release package will include a README, dataset card, license/terms, source attribution table, asset license notes.

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 abstract and introduction restrict the main claims to the VidUEU-Agent automated data-construction pipeline, the VidUEU-100K multimodal instruction dataset, and representative fine-tuning experiments on understanding, editing, and unified tasks. These claims are consistent with the experimental results in the main paper, the workflow details in the appendices, and the discussion of limitations.

   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: Please see Section J.

   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.
   - 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.
   - The authors should reflect on the factors that influence the performance of the approach.
   - 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.

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: [N/A]

   Justification: This paper is an empirical study on dataset construction and an automated data-curation pipeline, and it does not introduce theoretical results that require formal assumptions or mathematical 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.

- 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: Please see Section 4 and Appendix G.

   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.
   - The paper should provide sufficient information to reproduce the main experimental results.
   - The authors should describe the experimental setting, including datasets, models, training details, evaluation metrics, and implementation details.
   - The full details can be provided either in the main paper, appendix, code, or supplemental material.

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: [No]

   Justification: We have publicly released the VidUEU-100K dataset at `TODO`, while the implementation code is currently being organized for future release.

   Guidelines:

   - The answer NA means that the paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines 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.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

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: Section 4 summarizes the experimental setup, and Appendix G provides the full training and evaluation details. These details include the evaluated models, tasks, benchmarks, training configurations, and evaluation protocols needed to understand the reported results.

   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: Please see Appendix G.5. We report the run-to-run standard deviation of VLM-as-a-judge evaluations.

   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.
   - The method for calculating the error bars should be explained.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - 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.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: Please see Appendix G.

   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.

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The research conducted in this paper conforms to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity 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: Please see Section I.

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, fairness considerations, privacy considerations, and security considerations.
- The authors should consider possible harms that could arise when the technology is used as intended and functioning correctly, when it gives incorrect results, and from intentional or unintentional misuse.
- If there are negative societal impacts, the authors could also discuss mitigation strategies.

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?

    Answer: [N/A]

    Justification: VidUEU-100K is a derived multimodal instruction dataset and is not identified as a high-risk asset requiring specialized misuse safeguards. The release remains subject to applicable source-data licenses and redistribution restrictions.

    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.
    - Datasets that have been scraped from the Internet could pose safety risks, and 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: Appendix K lists the existing assets used in this work, including datasets, benchmarks, model families or API models, and baseline code. The appendix credits the original creators or owners and describes the relevant licenses and terms of use where available.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - Cite the original paper that produced the code package or dataset.
    - Remember to state which version of the asset is used.
    - If possible, include a URL.
    - State the name of the license for each asset.
    - If data was scraped from a particular source, state the copyright and terms of service of that source.
    - If releasing assets, include a license, copyright information, and terms of use in the package.

13. **New Assets**

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

    Answer: [Yes]

Justification: This paper introduces VidUEU-100K as a new multimodal instruction-tuning data asset. VidUEU-100K is constructed from sampled images of raw videos from OpenVid/OpenVidHD, Panda-70M, Wikimedia/Wikimedia Commons, and YouTube, while respecting upstream licenses and terms of use. We publicly release VidUEU-100K, a multimodal instruction-tuning dataset, at `TODO`. The dataset construction, composition, source attribution, and licensing restrictions are documented in Section 3.2 and Appendix K.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset, code, or model as part of their submissions via structured templates.
- 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.
- If releasing datasets or models, include documentation, licenses, and terms of use.

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: [N/A]

Justification: This paper does not involve crowdsourcing experiments or direct interactions with human subjects. The human audit involves two research annotators assessing sampled data instances for quality, without crowdsourcing or experiments involving human subjects.

Guidelines:

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

15. **Institutional Review Board 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 approvals or an equivalent approval or review based on the requirements of the authors' institution were obtained?

Answer: [N/A]

Justification: This paper does not involve direct human-subject research, crowdsourcing experiments, or participant-level behavioral-data collection.

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 human subjects research.
- If IRB approval was obtained, authors should clearly state this in the paper.
- For initial submissions, do not include information that would break anonymity, 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?

Answer: [Yes]

Justification: Section 3, Appendix K.3 and Appendix K.4 describe the use of LLMs/VLMs as core components of VidUEU-Agent. These components are used in the agent system, LLM-based scoring or judging, baseline comparisons, and partial writing assistance.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as important, original, or non-standard components.
- If the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.
- Please refer to the NeurIPS guidance for what should or should not be described.