Editing the Moving World: Model Editing for Video LLMs

Qian Zhang[1*], **Xinye Li**[1,2*], **Xiaokai Wu**[1], **Junhao Xu**[1], **Zhanyue Qin**[1],
Qingbin Liu[2], **Junxian Cai**[2], **Xi Chen**[2], **Bolin Zhang**[1,4], **Zhiying Tu**[1,4],
Dianhui Chu[1,4], **Xiaoyan Yu**[3], **Dianbo Sui**[1,4†]
[1]Harbin Institute of Technology [2]Tencent [3]Nanyang Technological University
[4]Shandong Key Laboratory of Digital Service Computing Technology and Systems
2023212072@stu.hit.edu.cn, suidianbo@hit.edu.cn

Abstract

Model Editing, also known as knowledge editing, is receiving increasing attention in the field of Large Language Models (LLMs). However, existing model editing approaches predominantly focus on knowledge-level or static visual domains, overlooking dynamic semantics. This paper exploratively applies six representative model editing methods (FT, IKE, MEND, SERAC, MEMIT and AlphaEdit) to Video Large Language Models (Vid-LLMs) and introduces the first benchmark specifically designed for Vid-LLMs editing—**VMEB** (**V**id-LLMs **M**odel **E**diting **B**enchmark)—systematically extending model editing research from static modalities to dynamic video scenarios. We position this work as a forward-looking benchmark and a foundational diagnostic study: in the video paradigm, our evaluation dimensions encompass traditional metrics including Reliability, Locality, and Generality, while also introducing a video-specific metric: Robustness. Based on experimental results, we analyze the strengths and limitations of existing model editing approaches, and identify new challenges and research directions for the future development of the model editing field within the context of multimodal and video paradigms. Our benchmark is available at https://github.com/Sakabamrisa/VMEB.

1 Introduction

Model Editing (Knowledge Editing) has rapidly emerged as a popular research direction for adapting Large Language Models (LLMs) to the ever-evolving real-world knowledge (Zhao et al., 2023; Yao et al., 2023; Hernandez et al., 2024; Wang et al., 2024a). Early work concentrates on updating factual triples, with approaches such as ROME (Meng et al., 2022b) and MEND (Mitchell et al., 2022a) suggesting that targeted parameter interventions can inject new facts while, to a certain extent, preserving unrelated knowledge. More recent studies have extended editing to richer downstream tasks (Mao et al., 2023; Chen et al., 2024; Li et al., 2024c; Wang et al., 2024b; Huang et al., 2024b) and to diverse knowledge representations beyond simple triples—e.g., events (Peng et al., 2024; Liu et al., 2024a), scripts (Li et al., 2025) and free-form text (Deng et al., 2025; Jiang et al., 2025).

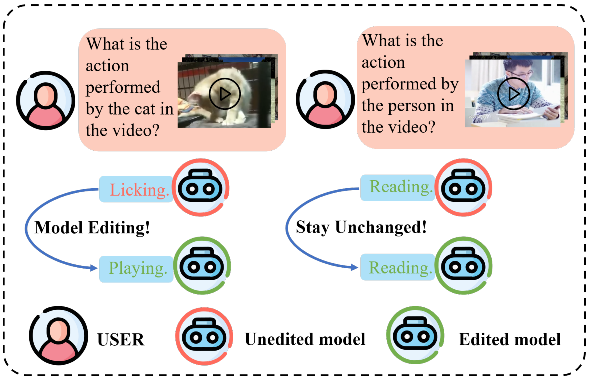

Figure 1: Overview of the Vid-LLMs editing task. The goal is to update the model's understanding of a specific video-text input. **Red-colored answers** indicate suboptimal outputs that require editing, while **green-colored answers** represent correct responses.

The paradigm is first transferred to Multimodal Large Language Models (MLLMs) by Cheng et al. (2024). Follow-up benchmarks such as VLKEB (Huang et al., 2024a)—which adds the *Portability* metric—and MMKE-Bench (Du et al., 2024)—which broadens the range of editable knowledge types—have strengthened evaluation protocols for MLLM editing. Nevertheless, these studies focus almost exclusively on static visual inputs, leaving the temporal dimension largely unexplored.

Concurrently, Video Large Language Models (Vid-LLMs) have advanced video understanding by harnessing LLMs' ability to model long sequences with rich temporal structure, enabling sophisticated reasoning over dynamic content (Tang et al., 2024; Fu et al., 2024a; Weng et al., 2024). Extending

† Dianbo Sui is the corresponding author.
* Equal contribution.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6072–6091

Model Editing to Vid-LLMs is thus a timely yet non-trivial challenge:

1. In the video paradigm, more complex motion patterns and spatial relationships must be considered, alongside the need to handle multi-layered abstract knowledge (Zhang et al., 2024; Bai et al., 2025b,a).
2. The editing methods must be capable of effectively injecting knowledge across frames and maintaining stability across time spans, tailored to the unique architecture of Vid-LLMs (Fu et al., 2024a).
3. The editing task requires a more general definition, moving beyond the traditional triple-format task definition (Meng et al., 2022a; Cheng et al., 2024).

Therefore, we take the first step toward video-centric Model Editing by presenting **VMEB**, the first comprehensive **V**id-LLMs **M**odel **E**diting **B**enchmark. In detail, VMEB systematically assesses editing performance in three widely used Vid-LLMs of different scales—LLaVA-NeXT-Video (7B) (Zhang et al., 2024), Qwen2.5-VL (3B & 7B) (Bai et al., 2025b) and Qwen3-VL (8B) (Bai et al., 2025a), one of the state-of-the-art MLLMs. Besides, evaluation in VMEB covers existing dimensions (Yao et al., 2023; Cheng et al., 2024) of Reliability, Locality, and Generality, while introducing video-specific axes that probe Robustness.

Based on VLMEB, we test representative editing methods: Fine-Tuning (FT), IKE (Zheng et al., 2023), MEND (Mitchell et al., 2022a), SERAC (Mitchell et al., 2022b), and two locate-then-edit methods: MEMIT (Meng et al., 2023) and AlphaEdit (Fang et al., 2025), emphasizing edits that transcend conventional factual updates or static multimodal model editing. Through various experiments, we find that: All current methods require improvement across one or more metrics. For instance, FT causes a high degree of destruction to the model, SERAC exhibits poor editing capability, and both AlphaEdit and MEMIT show suboptimal editing effectiveness for Vid-LLMs, primarily due to the structural incompatibility between the discrete-token localization paradigm they employ and the distributed nature of visual semantics across continuous embedding sequences in Vid-LLMs. Furthermore, some methods, like FT, IKE and MEND, maintain high scores even when the visual context is removed. Such low visual dependency demonstrates that these methods are not genuinely updated in terms of visual understanding; rather, they may merely shift their answer distributions to satisfy task requirements.

In general, we summarize our contributions as follows:

- **First exploration of video-centric Model Editing.** We take the initial step in extending Model Editing research from static modalities to Vid-LLMs, framing the unique challenges that arise in dynamic settings.
- **VMEB benchmark.** We propose VMEB—a comprehensive benchmark that rigorously evaluates how well existing editing methods perform on Vid-LLMs, across both existing and video-specific dimensions.
- **Extensive empirical analysis.** Through systematic experiments, we analyze our settings, tasks and performance—providing insights that we hope will catalyze further research in this emerging area.

We hope the proposed VMEB will spur further research on temporally grounded model editing and shed light on how knowledge updates can be effectively injected, preserved, and generalized within large multimodal models.

2 Related Work

2.1 Video Large Language Models

Video Large Language Models (Vid-LLMs) extend Large Language Models (LLMs) to video domains, addressing a multitude of video understanding tasks such as Video Question Answering (Video QA) and Video Captioning (Tang et al., 2025).

Early systems, like Flamingo (Alayrac et al., 2022) and FrozenBiLM (Yang et al., 2022), pair frozen language backbones with video encoders, delivering strong zero-shot results without task-specific fine-tuning, indicating frozen LMs as effective cores for video–language reasoning. Subsequent work shifts to instruction-tuned chat paradigms; models like VideoChat (Li et al., 2024a), Video-LLaMA (Zhang et al., 2023), Video-LLaVA (Lin et al., 2024), and Video-ChatGPT (Maaz et al., 2024) use lightweight adapters for spatiotemporal features and align with video-instruction pairs, enabling multi-turn dialogue on actions, causality, and temporal order.

Recent unified models focus on long-duration video understanding. LLaVA-Next-Video (Zhang et al., 2024) adapts image backbones for temporal reasoning, while MovieChat (Song et al., 2024) scales to long clips. The Qwen-VL series

(Qwen2.5 (Bai et al., 2025b) and Qwen3 (Bai et al., 2025a)) achieves SOTA performance by integrating dynamic resolution with advanced positional modeling like Interleaved-MRoPE.

Despite this rapid progress, Vid-LLMs still face challenges. These include fine-grained temporal localisation, multimodal hallucination, and efficient inference for very long sequences.

2.2 Multimodal Editing

Model (or Knowledge) Editing seeks to inject, revise, or remove specific facts in a pretrained model without exhaustive retraining (Sinitsin et al., 2020). Techniques span two axes—intrinsic, which directly alter network parameters, and extrinsic, which operate through prompts or external memory. A related line of work performs parameter-level fusion across pretraining checkpoints rather than targeted edits (Wang et al., 2026). Recently, growing interests extends the Model Editing to MLLMs. In detail, the editing paradigm is first transplanted to Multimodal LLMs (MLLMs) by Cheng et al. (2024). VLKEB extends evaluation with a *Portability* metric, assessing whether visual edits transfer across related contexts (Huang et al., 2024a), while MMKE-Bench broadens the spectrum of edit types to match real-world multimodal diversity (Du et al., 2024). Besides, Wang et al. (2024c) define the debiasing problem of MLLM as an editing problem and propose a novel set of evaluation metrics for MLLM debias editing. Yet both benchmarks focus on static imagery, overlooking the temporal dynamics intrinsic to video. Recent Vid-LLMs such as Llava-Next-Video and Qwen2.5-VL illustrate the feasibility of temporal reasoning (Zhang et al., 2024; Bai et al., 2025b), but no framework yet measures how well edits persist over time. We bridge this gap with **VMEB**, the first Vid-LLMs Model Editing Benchmark, which extends classic metrics—Reliability, Locality, and Generality—with temporal-robustness axes (e.g., frame skipping, speed perturbations), offering a comprehensive testbed for editing in dynamic multimodal settings.

3 Vid-LLMs Editing

We illustrate the proposed task of Vid-LLMs editing in Figure 2. We will introduce the task definition (§3.1), and dataset construction details (§3.2 and Appendix B).

3.1 Task Definition

We define the mapping $y_o = f(v_e, x_e; \theta)$ as the inference process of Vid-LLMs parameterized by θ, where v_e refers to the editing video input, x_e refers to the editing text prompt input and y_o represents the original output answer of the model. After the model undergoes editing, θ becomes the edited parameter θ', and we want the output to correspondingly change to $y_e = f(v_e, x_e; \theta')$.

To evaluate the effectiveness of model editing, we design a dataset $\mathcal{D}_{edit}$, defined as a quadruple (v_e, x_e, y_o, y_e). Concurrently, we use $\mathcal{M}$ as a notation symbol, where the superscript represents the scope of the data and the subscript indicates the evaluation domain. Drawing inspiration from Yao et al. (2023) and Huang et al. (2024a), our evaluation metrics specifically designed for Vid-LLMs editing are presented as follows:

Reliability. To directly verify the effectiveness of the model editing method, we define the percentage of edited models outputting the target answer as the value of the reliability metric[1], which is described as the following:

$$\mathcal{M}_{\text{rel}} = \underset{(v_e, x_e, y_o, y_e) \sim \mathcal{D}_{\text{edit}}}{\mathbb{E}} \mathbb{1}\left\{f\left(v_e, x_e; \theta'\right) = y_e\right\} \tag{1}$$

where θ' refers to the edited parameters.

Locality. When editing models, we aim for edits that are not only effective but also precise. To evaluate an editing method's ability to preserve unrelated parts of the model while making targeted changes, we introduce the locality metric, which is divided into $\mathcal{M}^{\text{t}}_{\text{loc}}$ and $\mathcal{M}^{\text{v}}_{\text{loc}}$.

$\mathcal{M}^{\text{t}}_{\text{loc}}$ describes the stability of the foundation language model—which serves as the core component of all models—after editing. Recent research has shown that maintaining language model stability during editing is crucial for preserving general capabilities while implementing targeted changes (De Cao et al., 2021; Mitchell et al., 2022b).

$\mathcal{M}^{\text{v}}_{\text{loc}}$ describes the stability of the model's visual decoding and projection layers after editing. This metric is particularly important in multimodal models where visual understanding must remain intact despite text-based edits (Meng et al., 2022b; Yao et al., 2023).They are calculated as follows:

[1]Accuracy is calculated per token, then averaged across all entries for the final rate.

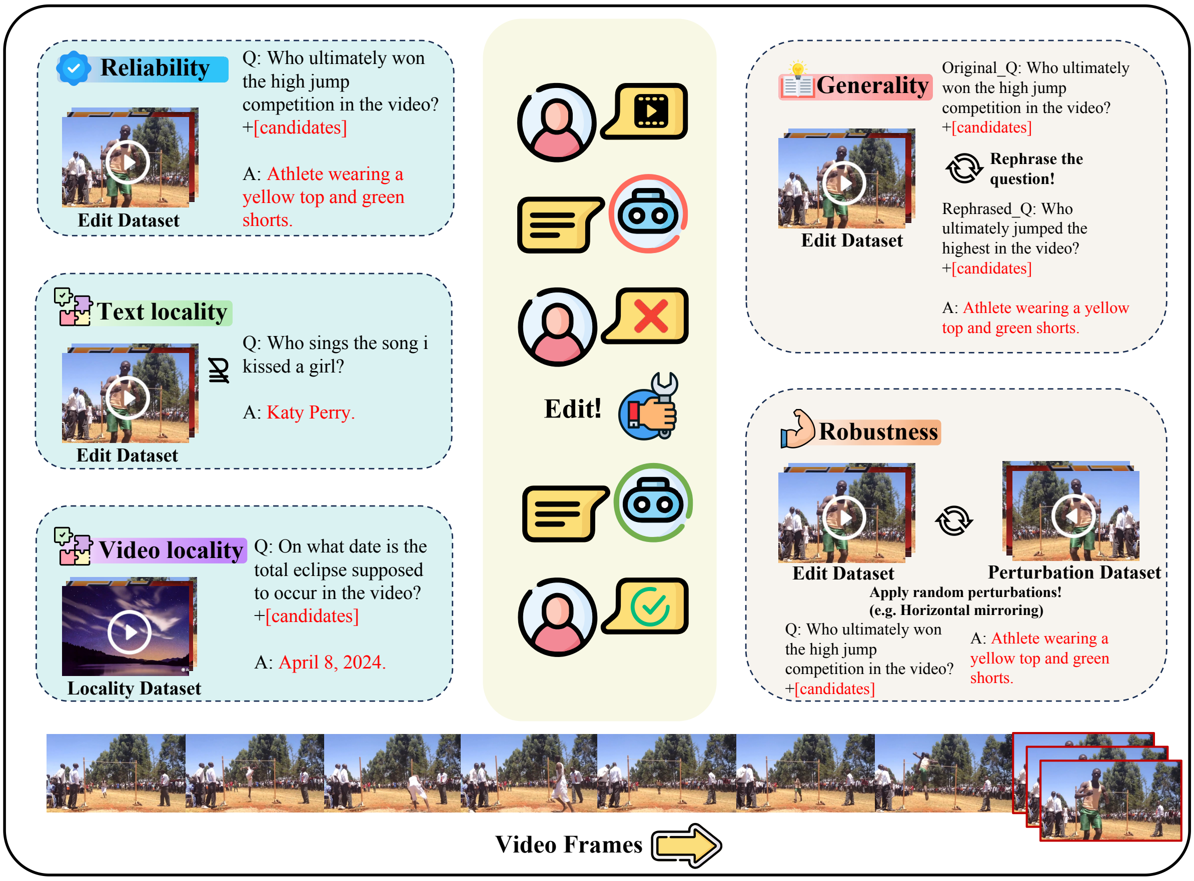

Figure 2: Framework of our Vid-LLMs Editing Tasks

$$\mathcal{M}_{\text{loc}}^{\text{t}} = \mathop{\mathbb{E}}_{(x_l, y_l) \sim \mathcal{D}_{\text{loc}}^{\text{t}}} \mathbb{1}\{f(x_l; \theta') = f(x_l; \theta)\} \quad (2)$$

$$\mathcal{M}_{\text{loc}}^{\text{v}} = \mathop{\mathbb{E}}_{(v_l, x_l, y_l) \sim \mathcal{D}_{\text{loc}}^{\text{v}}} \mathbb{1}\{f(v_l, x_l; \theta') = f(v_l, x_l; \theta)\} \quad (3)$$

where $\mathcal{D}_{\text{loc}}^{\text{t}}$ and $\mathcal{D}_{\text{loc}}^{\text{v}}$ respectively refers to text-locality and video-locality dataset stated in §3.2.2, x_l represents text prompt inputs that are not related to the editing domain, v_l represents the video input that is out of scope, while y_l represents the model output answers corresponding to x_l or x_l and v_l.

Generality. Edited models require good generalization capabilities, which will be evaluated through modifications to questioning methods or grammatical structures. The accuracy rate will be calculated after posing these varied questions, as follows:

$$\mathcal{M}_{\text{gen}} = \mathop{\mathbb{E}}_{\substack{(v_e, x_e, y_o, y_e) \sim \mathcal{D}_{\text{edit}} \\ x_r \sim \mathcal{N}(x_e)}} \mathbb{1}\{f(v_e, x_r; \theta') = y_e\} \quad (4)$$

where $\mathcal{N}(x_e)$ stands for the in-scope text prompt input, x_r stands for the rephrased text prompt input according the original text prompt input.

Robustness. Robustness is defined as an algorithm's ability to maintain performance and stability when faced with uncertainties such as noise interference, parameter variations, and anomalous conditions. Unlike image-based editing benchmarks (Cheng et al., 2024) that utilize generative models to create semantically equivalent inputs for evaluating *Generality*, we identify fundamental barriers in applying this paradigm to the video domain. Specifically, for medium-to-long duration videos (*e.g.,* exceeding 30 seconds), current state-of-the-art video generation and reconstruction methods—including diffusion and transformer-based architectures (such as Sora and Veo)—struggle to maintain semantic consistency over extended temporal windows (Liu et al., 2024b; Huang et al., 2023), frequently suffering from **semantic drift**, where object identities, spatial relationships, and temporal causalities degrade or hallucinate as the sequence lengthens (Xing et al., 2024).

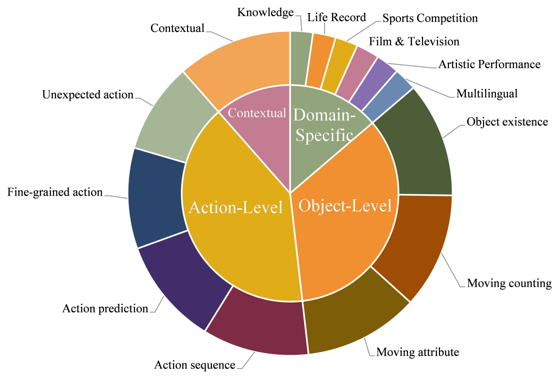

Figure 3: Dataset Composition with Primary and Secondary Categories

This renders generative semantic perturbations unreliable as ground truth, making a generation-based "Video-Generality" metric currently unfeasible.

Consequently, our Robustness metric operates at the **signal level** rather than the semantic level: we apply controlled low-level transformations (*e.g.,* rotation, grayscale conversion, speed changes) that can guarantee semantic preservation by construction, while testing the stability of edited models against visual variations—which have been shown to significantly affect model outputs in multimodal settings (Elsayed et al., 2018). To quantify this capability, we introduce the $\mathcal{M}_{rob}$ metric as follows:

$$\mathcal{M}_{\text{rob}} = \mathop{\mathbb{E}}_{\substack{(v_e, x_e, y_o, y_e) \sim \mathcal{D}_{\text{edit}} \\ v_r \sim \mathcal{N}(v_e)}} \mathbb{1}\left\{ f\left(v_r, x_e; \theta'\right) = y_e \right\} \tag{5}$$

where $\mathcal{N}(v_e)$ represents the original editing video, and v_r represents the new video obtained by applying random perturbation processing to the original video. We plan to incorporate semantic-level perturbations in future work as video generation technology matures.

3.2 Dataset Construction

Our dataset, VMEB, represents a fundamental type of Edit Video-QA, similar to VQA (Visual Question Answering) (Antol et al., 2015), but extends visual information to video understanding. Dataset consists of 1,578 data entries and their corresponding videos, with the videos categorized into 14 distinct subcategories. For detailed classification, please refer to Figure 3 and Table 3 in Appendix.

Specifically, each data entry consists of 11 key-value pairs, as detailed in Table 6. These are categorized into four components: Edit dataset, Locality dataset, Generality dataset, and Robustness dataset.

3.2.1 Edit Dataset.

We begin with data selection, choosing videos and annotations from the influential video datasets MVBench (Li et al., 2024b) and Video-MME (Fu et al., 2024b) as our raw data. $\mathcal{M}_{edit}$ consists of four data components, where we concatenate the original questions and options for the videos as x_e , with the videos serving as v_e. We test MVBench and Video-MME using the LLaVA-NeXT-Video(7B) model and retain the incorrectly answered responses as y_o, which represents the original outputs requiring editing, while the correct answers for these entries are designated as y_e. While all entries with correct answers are discarded from the dataset.

3.2.2 Locality Dataset.

We must evaluate the model's normal performance after editing, therefore we decide to use commonsense question-answering from Wikipedia to assess textual locality. We randomly select 1,580 knowledge-based question-answer pairs from VLKEB (Huang et al., 2024a), such as ("Where does Disney's Hunchback of Notre Dame take place?", "Paris"), as the tuple pairs (x_l, y_l) in $\mathcal{D}^t_{loc}$.

Similarly, for video-level locality, we can query the model using existing videos in the dataset to evaluate the impact of editing methods on the model. For $\mathcal{D}^v_{loc}$, we re-shuffle the dataset using a shuffling algorithm and create one-to-one correspondences between the shuffled dataset and the original ordered dataset, specifically $(v_e, x_e, y_e) \mapsto (v_l, x_l, y_l)$.

3.2.3 Generality Dataset.

The Definition of Rephrased Questions. For the mapping $y_e = f(v_e, x_e; \theta)$, we consider it as the model's answer to the question x_e given the video input v_e. We can then define $argf(y_e, v_e; \theta) = x$, which represents a question x that would yield the answer y_e under the same parameters and video input, noting that this x is not unique. The process of constructing a rephrased question involves designing an x_r such that $argf(y_e, v_e; \theta) = x_r$.

Specific examples. For model editing, we aim to achieve excellent generalization capabilities. Taking text editing as an example, if we intend to edit ("Who is the president of USA now", "Joe Biden", "Donald Trump"), then similarly structured questions with grammatical and structural variations, such as ("Who currently holds the office of the U.S. President", "Joe Biden", "Donald Trump"), should

Model	Method	Rel.	Txt-Loc.	Vid-Loc.	Gen.	Rob.
LLaVA-NeXT-Video *Model Size: 7B*	Base Model	0.00	100.00	100.00	0.00	0.00
	FT (Language)	**99.96**	79.93	22.79	**99.92**	**99.96**
	FT (Vision)	99.95	**100.00**	13.48	98.46	99.74
	IKE	87.68	42.26	24.49	86.39	87.87
	MEND (Language)	96.31	99.02	85.11	96.12	96.21
	MEND (Vision)	91.44	**100.00**	57.71	90.69	89.29
	SERAC	80.23	99.87	96.39	79.51	78.98
	AlphaEdit	42.13	56.78	91.85	42.12	41.92
	MEMIT	43.00	84.16	**98.77**	42.93	42.77
Qwen2.5-VL *Model Size: 3B*	FT (Language)	**100.00**	94.03	55.90	**99.84**	**100.00**
	FT (Vision)	99.79	**100.00**	32.58	98.98	91.93
	IKE	77.90	71.06	21.41	79.36	77.74
	MEND (Language)	99.05	98.53	74.85	98.46	98.73
	MEND (Vision)	90.21	**100.00**	56.97	85.01	80.03
	SERAC	83.27	99.35	89.75	82.71	80.17
	AlphaEdit	40.91	68.46	86.39	41.10	40.91
	MEMIT	40.90	90.66	**99.02**	40.56	40.90
Qwen2.5-VL *Model Size: 7B*	FT (Language)	**100.00**	74.18	9.92	**99.81**	**99.95**
	FT (Vision)	99.56	**100.00**	30.77	97.50	89.36
	IKE	85.85	80.75	21.08	79.48	84.96
	MEND (Language)	96.97	97.43	71.80	96.18	96.43
	MEND (Vision)	91.21	**100.00**	54.70	87.50	81.99
	SERAC	75.46	99.77	88.09	75.57	75.06
	AlphaEdit	39.32	86.25	**97.39**	40.19	39.32
	MEMIT	39.71	78.86	94.62	40.10	39.71
Qwen3-VL *Model Size: 8B*	FT (Language)	**100.00**	61.82	10.96	**99.84**	**99.84**
	FT (Vision)	99.65	**100.00**	29.14	98.15	90.45
	IKE	99.15	73.49	22.24	98.99	99.04
	MEND (Language)	94.73	98.11	59.67	93.55	91.57
	MEND (Vision)	90.75	**100.00**	55.42	86.33	81.15
	SERAC	82.13	99.46	87.48	81.07	80.89
	AlphaEdit	39.29	83.11	95.16	39.50	39.29
	MEMIT	39.15	92.12	**96.15**	40.14	39.15

Table 1: Comparison of different models and editing methods across various metrics. *Rel.*, *Txt-Loc.*, *Vid-Loc.*, *Gen.* and *Rob.* denote Reliability, Text-Locality, Video-Locality, Generality and Robustness, respectively. Best results are highlighted in bold.

also be successfully edited. To evaluate this generalization capability, we utilize GPT-4o to generate high-quality question restatements, which are subsequently manually reviewed. This review process can ensure that the re-stated questions x_r maintain the same answers as the original questions x_e while preserving semantic similarity.

3.2.4 Robustness Dataset.

Videos, as a type of signal, are frequently subject to noise interference such as disturbances and distortion. We aim for edits represented as $(v_e, x_e, y_e) \underset{v_r \sim \mathcal{N}(v_e)}{\mapsto} (v_r, x_e, y_e)$ to maintain equivalent efficacy. Therefore, we input v_r into the model that has undergone editing (v_e, x_e, y_e) to evaluate its robustness. We apply common random perturbations to the original video v_e, as shown in Table 4, thereby obtaining v_r. To conduct a more rigorous evaluation, we prepare video inputs with more severe perturbations, such as completely black or heavily noisy videos.

4 Experiments

We evaluate six editing methods (FT, IKE, MEND, SERAC, MEMIT and AlphaEdit) on the VMEB benchmark across LLaVA-NeXT-Video, Qwen2.5-VL, and Qwen3-VL. Our analysis proceeds as follows: §4.1 assesses performance across four key metrics; §4.2 investigates the "Localization Dilemma" in Locate-then-Edit paradigms; §4.3

compares vision versus language layer editing; and §4.4 examines visual context dependency and shortcut learning via perturbations.

4.1 Experimental Results

The experimental results are present in Table 1. From the table, we can find that: In general, MEND demonstrates the best overall balance. While FT achieves peak reliability, it severely compromises video locality. Conversely, AlphaEdit and MEMIT excel in locality but suffer in editing effectiveness.

Reliability. FT and MEND dominate with $> 90\%$ accuracy, significantly outperforming IKE and SERAC ($75\% - 85\%$). Notably, AlphaEdit and MEMIT exhibit the lowest reliability ($\sim 40\%$), indicating that the locate-then-edit paradigm restricts editing effectiveness.

Locality. AlphaEdit and MEMIT achieve state-of-the-art Video-Locality ($> 90\%$), far surpassing FT ($< 30\%$). SERAC also performs well in locality ($> 87\%$). MEND maintains a balanced profile with high Text-Locality ($> 97\%$), whereas IKE struggles to accurately localize changes despite being a non-parametric method.

Generality & Robustness. FT and MEND show superior knowledge comprehension and stability ($> 90\%$). IKE and SERAC yield mediocre results, primarily relying on the base model's inherent generalization capabilities. AlphaEdit and MEMIT perform poorly here ($\sim 40\%$), which is directly correlated with their lower reliability scores.

4.2 The Localization Dilemma in Vid-LLMs

As indicated in Table 1, Locate-then-Edit methods exhibit a "Localization Dilemma" (high locality but low reliability). Theoretically, these methods model knowledge update as satisfying $(W + \Delta W)k^* = v_{target}$ (Meng et al., 2023), premised on localizing knowledge to a specific discrete text key k^*. However, Vid-LLMs distribute visual semantics across continuous embedding sequences H_v via cross-attention mechanisms. Consequently, projecting the dynamic, distributed information from H_v onto a single static token k^* constitutes a mathematically ill-posed problem. The engineering compromise of fixing k^* to the final token (Wang et al., 2023) fails to resolve this structural incompatibility, confirming the failure of the discrete-token paradigm in multimodal settings and the necessity for Video-Native approaches.

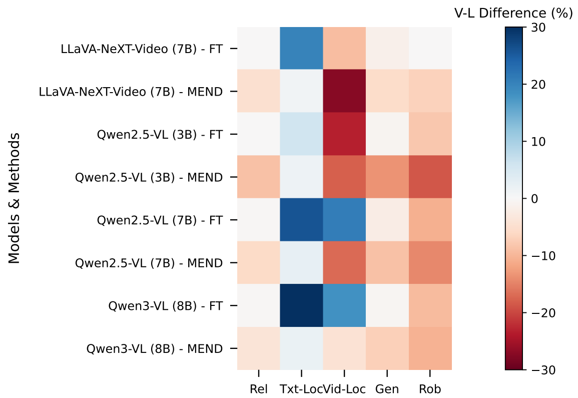

Figure 4: Heatmap of comparing editing vision layers methods with editing language layers methods, computed as $Metrics(Vision) - Metrics(Language)$. Red coloration indicates regions where vision layer editing yields inferior performance compared to language layer editing.

4.3 Editing Language Layers is More Efficient.

According to Figure 4, the heatmap visualizes the differential performance between vision layer editing and language layer editing for identical methods applied to the same model, computed as the results from vision layer editing minus those from language layer editing. It is worth noting that SERAC is an external method while IKE is prompt-based, so they do not distinguish between the modules being edited. Regarding AlphaEdit and MEMIT, we exclusively implement them on language layers as they are tailored for causal language modeling architectures. Editing vision layers demonstrates significantly reduced effectiveness compared to language layers, with poor performance in Vid-locality and robustness metrics. This indicates that in multimodal large models, language layers contain more implicit knowledge (Huang et al., 2024a; Cheng et al., 2024) and are directly correlated with LLM outputs. Therefore, we conclude that editing language layers is more efficient. However, this efficiency raises a critical concern: does the dominance of language layers imply that visual semantics—and by extension, the targeted editing of visual modules—are inconsequential? We investigate this paradox next.

4.4 Does Visual Context Really Matter in Vid-LLM Editing?

Definitely, yes. But based on the experimental results, we observe an intriguing phenomenon: performance does not decline as anticipated. In the evolution of Knowledge Editing, the degradation

Method	Setting	Rel.	Grounded	Txt-Loc.	Vid-Loc.	Gen.	Rob.
FT	Normal Perturbation	99.90	**0.00**	61.91	10.66	99.75	100.00
	v_e Ablation	100.00	-	60.31	10.35	100.00	99.86
	v_r Ablation	99.72	-	61.78	10.77	99.66	99.39
IKE	Normal Perturbation	98.87	**0.00**	73.72	22.37	99.02	98.90
	v_e Ablation	99.83	-	73.50	22.26	99.51	99.15
	v_r Ablation	99.32	-	73.13	21.99	99.38	99.52
MEND	Normal Perturbation	94.35	**10.81**	98.20	60.05	93.77	91.27
	v_e Ablation	93.11	-	93.56	33.04	91.85	90.09
	v_r Ablation	94.48	-	98.49	59.74	93.72	86.84
SERAC	Normal Perturbation	81.88	**20.55**	99.71	87.82	81.25	80.89
	v_e Ablation	76.72	-	99.14	85.24	75.47	80.83
	v_r Ablation	81.96	-	99.41	87.34	80.76	74.39

Table 2: Results of the four editing methods after visual context removal. **Grounded** denotes the Visually Grounded Score. This metric penalizes models that ignore visual context (Blindness) and amplifies visual dependency signals.

from text to image (Huang et al., 2024a; Cheng et al., 2024) and subsequently to video is not substantial. This raises a fundamental question: Can model editing truly teach models to "understand" changes in visual semantics and spatial logical relationships?

To probe visual criticality, we test the top-performing methods (FT, IKE, MEND, SERAC) by replacing inputs v_e/v_r with black videos, and we define Visually Grounded Score as:

$$\mathcal{M}_{\text{grounded}} = \text{Rel}_{\text{normal}} \times \left(1 - \frac{\text{Rel}_{\text{ablation}}}{\text{Rel}_{\text{normal}}}\right)^{\alpha}$$

Here we choose $\alpha = 0.5$. Lower values indicate reduced visual reliance and incompatibility with the video paradigm. Counter-intuitively, experimental resuls in Table 2 shows negligible impact from removing visual context, with success rates remaining high even on black inputs.

For FT and IKE, the results remain virtually unchanged. We attribute this to the inherent blindness of FT and the fact that IKE prioritizes in-context textual cues over visual semantics. Conversely, MEND exhibits significant declines in video-locality and robustness. We analyze that this occurs because completely black noise videos constitute out-of-distribution (OOD) data, causing the hypernetwork to compute erroneous and destructive parameter updates. Similarly, SERAC shows a decrease in reliability and robustness, which is likely due to misjudgments by the Scope Classifier caused by the lack of visual context.

These results validate our robustness metric's effectiveness in capturing visual semantic changes and suggest that model editing operates in a probabilistic manner. This reflects the distinct complexity of knowledge storage in MLLMs compared to LLMs. Since video understanding depends heavily on dynamic visual context rather than just parametric knowledge utilized in text generation, existing approaches that solely modify static weights fail to address the high-dimensional dependency between visual inputs and semantic reasoning.

Alternatively, it is possible that these models do not strictly differentiate between modalities at higher representational levels, potentially forming a shared semantic space where concepts transcend their original modalities. This resonates with our findings in Section §4.3: the superior efficiency of editing language layers likely stems from the fact that complex multimodal knowledge is aggregated within this shared, high-level semantic space. However, we ultimately believe that targeted model editing methods under multimodal and video paradigms are necessary.

5 Conclusion

This paper introduces VMEB as the first diagnostic benchmark for model editing in Vid-LLMs. Rather than proposing an incremental improvement to existing editing pipelines, we take a step back to ask: do current methods actually work in video paradigms, and if not, why? In detail, we provide pioneering implementations of FT, IKE, MEND, SERAC, AlphaEdit and MEMIT for Qwen2.5-VL, LLaVA-NeXT-Video, and Qwen3-VL. Through extensive experiments, our work identifies the core incompatibilities in existing approaches and demonstrates that current research faces significant limitations due to the inherent complexity of the video paradigm, which demands the handling of intricate motion patterns, spatial relationships, and multi-layered abstract knowledge. This establishes that multimodal model editing requires fundamentally new frameworks rather than adaptations of text-only methods, charting a critical direction for advancing knowledge editing in Vid-LLMs. Furthermore, the editing task itself requires a more general definition, moving beyond traditional triple-format constraints to dynamic multimodal datas.

Limitations

Model Architectural Diversity. While we select representative Vid-LLMs for our experiments, we do not exhaustively cover the full spectrum of vision-language fusion architectures, such as diverse temporal aggregation strategies or specialized cross-attention variants. However, given the high consistency of our experimental results—particularly the phenomenon of visual blindness—across the tested models, we posit that our findings hold strong generalizability across mainstream Vid-LLM architectures, suggesting that specific architectural differences have a limited impact on the core conclusions.

Rigidity of Localization Strategies. For Locate-then-Edit methods (e.g., AlphaEdit and MEMIT), we adhere to the standard paradigm within the EasyEdit framework (Wang et al., 2023), which strictly designates a specific token in the text sequence (typically the final token) as the editing target. We do not investigate alternative strategies tailored to multimodal characteristics, such as targeting visual encoder output tokens or specific fusion layers. We believe this constraint is intrinsically linked to the design paradigms of these methods.

Synthetic Nature of Perturbations. Our evaluation of Robustness primarily relies on algorithmically generated, signal-level perturbations (e.g., noise, black frames, and speed variations). Due to the challenges inherent in data generation, we are unable to include real-world "semantic-level" perturbations, such as variations of the same event under different lighting conditions or camera angles. Consequently, our current metrics largely measure stability against low-level visual feature shifts and may not fully capture the model's semantic robustness within complex, real-world dynamic environments.

Declaration of LLM Usage

Large language models are employed only for minor editorial assistance, such as language refinement and clarity enhancement. All scientific ideas, experimental results, interpretations, and conclusions are entirely the authors' own. The authors independently perform the literature review and curate all references from published, verifiable sources, and take full responsibility for the final manuscript.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (Grant No. 2023YFB3307500), the National Natural Science Foundation of China (Grant No. 62306087 and 62472121), the Natural Science Foundation of Shandong Province (Grant No. ZR2023QF154), the Key Research and Development Program of Shandong Province (Grant No. 2025CXPT077), the Research on Cognitive Processing Technologies for Multimodal Big Data in Policing Information Project (Grant No. 2024DXZD0004), the Special Funding Program of Shandong Taishan Scholars Project and CCF-Tencent Rhino-Bird Open Research Fund (Grant No. CCF-Tencent RAGR20250105).

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a Visual Language Model for Few-Shot Learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2425–2433.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

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, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Canyu Chen, Baixiang Huang, Zekun Li, Zhaorun Chen, Shiyang Lai, Xiongxiao Xu, Jia-Chen Gu, Jindong Gu, Huaxiu Yao, Chaowei Xiao, Xifeng Yan, William Yang Wang, Philip Torr, Dawn Song, and Kai Shu. 2024. Can editing llms inject harm? *Preprint*, arXiv:2407.20224.

Siyuan Cheng, Bozhong Tian, Qingbin Liu, Xi Chen, Yongheng Wang, Huajun Chen, and Ningyu Zhang. 2024. Can we edit multimodal large language models? *Preprint*, arXiv:2310.08475.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jingcheng Deng, Zihao Wei, Liang Pang, Hanxing Ding, Huawei Shen, and Xueqi Cheng. 2025. Everything is editable: Extend knowledge editing to unstructured data in large language models. In *The Thirteenth International Conference on Learning Representations*.

Yuntao Du, Kailin Jiang, Zhi Gao, Chenrui Shi, Zilong Zheng, Siyuan Qi, and Qing Li. 2024. Mmke-bench: A multimodal editing benchmark for diverse visual knowledge.

Gamaleldin F. Elsayed, Shreya Shankar, Brian Cheung, Nicolas Papernot, Alexey Kurakin, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial examples that fool both computer vision and time-limited humans. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3914–3924, Red Hook, NY, USA. Curran Associates Inc.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Rongrong Ji, and Xing Sun. 2024a. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *Preprint*, arXiv:2405.21075.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2024b. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. *arXiv preprint arXiv:2405.21075*.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. *Preprint*, arXiv:2304.00740.

Han Huang, Haitian Zhong, Tao Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2024a. Vlkeb: A large vision-language model knowledge editing benchmark. *Preprint*, arXiv:2403.07350.

Xiusheng Huang, Yequan Wang, Jun Zhao, and Kang Liu. 2024b. Commonsense knowledge editing based on free-text in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14870–14880, Miami, Florida, USA. Association for Computational Linguistics.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. 2023. Vbench: Comprehensive benchmark suite for video generative models. *Preprint*, arXiv:2311.17982.

Houcheng Jiang, Junfeng Fang, Ningyu Zhang, Guojun Ma, Mingyang Wan, Xiang Wang, Xiangnan He, and Tat seng Chua. 2025. Anyedit: Edit any knowledge encoded in language models. *Preprint*, arXiv:2502.05628.

KunChang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2024a. Videochat: Chat-centric video understanding. *Preprint*, arXiv:2305.06355.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024b. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Xinye Li, Zunwen Zheng, Qian Zhang, Dekai Zhuang, Jiabao Kang, Liyan Xu, Qingbin Liu, Xi Chen, Zhiying Tu, Dianhui Chu, and Dianbo Sui. 2025. ScEdit: Script-based assessment of knowledge editing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 2032–2052, Vienna, Austria. Association for Computational Linguistics.

Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. 2024c. Badedit: Backdooring large language models by model editing. In *The Twelfth International Conference on Learning Representations*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024. Video-llava: Learning united visual representation by alignment before projection. *Preprint*, arXiv:2311.10122.

Jiateng Liu, Pengfei Yu, Yuji Zhang, Sha Li, Zixuan Zhang, Ruhi Sarikaya, Kevin Small, and Heng Ji. 2024a. EVEDIT: Event-based knowledge editing for deterministic knowledge propagation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4907–4926, Miami, Florida, USA. Association for Computational Linguistics.

Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. 2024b. Sora: A review on background, technology, limitations, and opportunities of large vision models. *Preprint*, arXiv:2402.17177.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2024. Video-chatgpt: Towards detailed video understanding via large vision and language models. *Preprint*, arXiv:2306.05424.

Shengyu Mao, Ningyu Zhang, Xiaohan Wang, Mengru Wang, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2023. Editing personality for llms.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 35.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022b. Locating and editing factual knowledge in GPT. In *NeurIPS*.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *ICLR*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831. PMLR.

Hao Peng, Xiaozhi Wang, Chunyang Li, Kaisheng Zeng, Jiangshan Duo, Yixin Cao, Lei Hou, and Juanzi Li. 2024. Event-level knowledge editing. *Preprint*, arXiv:2402.13093.

Anton Sinitsin, Vsevolod Plokhotnyuk, Dmitriy Pyrkin, Sergei Popov, and Artem Babenko. 2020. Editable neural networks. *Preprint*, arXiv:2004.00345.

Enxin Song, Wenhao Chai, Tian Ye, Jenq-Neng Hwang, Xi Li, and Gaoang Wang. 2024. Moviechat+: Question-aware sparse memory for long video question answering. *arXiv preprint arXiv:2404.17176*.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. 2024. Video understanding with large language models: A survey. *Preprint*, arXiv:2312.17432.

Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, Ali Vosoughi, Chao Huang, Zeliang Zhang, Pinxin Liu, Mingqian Feng, Feng Zheng, Jianguo Zhang, Ping Luo, Jiebo Luo, and Chenliang Xu. 2025. Video understanding with large language models: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, PP:1–1.

Peng Wang, Ningyu Zhang, Xin Xie, Yunzhi Yao, Bozhong Tian, Mengru Wang, Zekun Xi, Siyuan Cheng, Kangwei Liu, Guozhou Zheng, and Huajun Chen. 2023. Easyedit: An easy-to-use knowledge editing framework for large language models. *CoRR*, abs/2308.07269.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024a. Knowledge editing for large language models: A survey. *ACM Comput. Surv.*, 57(3).

Xiaohan Wang, Shengyu Mao, Ningyu Zhang, Shumin Deng, Yunzhi Yao, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024b. Editing conceptual knowledge for large language models. *Preprint*, arXiv:2403.06259.

Zecheng Wang, Xinye Li, Zhanyue Qin, Chunshan Li, Zhiying Tu, Dianhui Chu, and Dianbo Sui. 2024c. Can we debias multimodal large language models via model editing? In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 3219–3228, New York, NY, USA. Association for Computing Machinery.

Zecheng Wang, Deyuan Liu, Chunshan Li, Dianhui Chu, Weipeng Chen, Bingning Wang, and Dianbo Sui. 2026. Towards enhanced llm pretraining: Dynamic checkpoint merging via generation quality. *Information Fusion*, 125:103415.

Yuetian Weng, Mingfei Han, Haoyu He, Xiaojun Chang, and Bohan Zhuang. 2024. Longvlm: Efficient long video understanding via large language models. *Preprint*, arXiv:2404.03384.

Zhen Xing, Qijun Feng, Haoran Chen, Qi Dai, Han Hu, Hang Xu, Zuxuan Wu, and Yu-Gang Jiang. 2024. A survey on video diffusion models. *Preprint*, arXiv:2310.10647.

Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. 2022. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu

Zhang. 2023. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10222–10240, Singapore. Association for Computational Linguistics.

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

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024. Llava-next: A strong zero-shot video understanding model.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. A survey of large language models. *CoRR*, abs/2303.18223.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? *CoRR*, abs/2305.12740.

A Models and Editing Methods

A.1 Vid-LLMs

LLaVA-NeXT-Video (Zhang et al., 2024) is a state-of-the-art Vid-LLM excelling in video understanding via extensive instruction tuning on a synthetic dataset. It processes video by treating sparse frames as a "long image," enabling it to combine robust image understanding with acquired temporal reasoning skills.

Qwen2.5-VL (Bai et al., 2025b) is a flagship multimodal LLM series particularly strong in long-video comprehension (up to hours). It features dynamic resolution processing and absolute time encoding for precise event localization, using a redesigned Vision Transformer and time-aligned Multimodal Rotary Position Embedding (MROPE) to understand temporal dynamics.

Qwen3-VL (Bai et al., 2025a) represents the latest iteration in the Qwen-VL series, building upon the architectural strengths of its predecessors. It further enhances visual-language alignment and reasoning capabilities, delivering superior performance in processing complex video inputs and handling long-context multimodal tasks with higher efficiency.

A.2 Editing Methods

A.2.1 Fine-tuning

Fine-tuning is the most fundamental method among intrinsic editing approaches. It primarily updates the model's parameters through gradient descent to achieve the objective of altering specific model characteristics.

A.2.2 In-Context Knowledge Editing (IKE)

In-Context Knowledge Editing (IKE) is a model editing method based on In-Context Learning (ICL). It directly modifies the factual knowledge within the LLM by designing specific demonstration samples, achieving efficient knowledge updates with minimal side effects and without adjusting model parameters (Zheng et al., 2023).

A.2.3 MEND

MEND is an efficient model editing method based on gradient decomposition. By training a lightweight auxiliary editing network, it transforms traditional fine-tuning gradients into low-rank parameter updates, thereby enabling fast and accurate editing of LLMs (Mitchell et al., 2022a).

A.2.4 SERAC

SERAC is a semi-parametric editing approach based on a retrieval-augmented counterfactual model. It stores edits in an explicit memory and learns to reason over them to adjust the underlying model's predictions as needed (Mitchell et al., 2022b).

A.2.5 MEMIT

Mass-Editing Memory in a Transformer (MEMIT) is a scalable method designed for injecting large batches of knowledge into LLMs. It distributes the parameter updates across multiple layers of the model's Multi-Layer Perceptrons (MLPs), allowing for stable mass-editing while minimizing the degradation of general model performance (Meng et al., 2023).

A.2.6 AlphaEdit

AlphaEdit is a projection-based editing method that constrains parameter updates to the null space of the model's existing covariance matrix. By doing so, it minimizes interference with previously learned knowledge while effectively injecting new facts, thereby achieving a superior balance between editing reliability and locality (Fang et al., 2025).

B Dataset Construction Details

The dataset is partitioned according to the following specifications:

- **Train-Evaluation Split**: 7:3 ratio
- **Training Set** (train.json): 1,106 data instances
- **Evaluation Set** (eval.json): 472 data instances
- **Video Files** (VMEB.zip) 3976 videos
- **Additional File**: eval_multihop.json (identical to eval.json for compatibility with VLKEB. For detailed explanations, please refer to §B.4.2.)

B.1 Construction Process of Rephrased Questions

B.1.1 Specific Construction Method

We utilize the GPT-4o model to construct rephrased questions, with the prompt design format illustrated in Figure 5. During the construction process, for each data point, we generate three rephrased questions and manually select the one that best meets our requirements. The selected question transforms the relevant sentence structure, grammar, or syntactic patterns without altering the subject matter of the original question. This selected question serves as our rephrased question.

B.1.2 Data Example

We randomly select a data point from our dataset as an example. The "src" key of this data point (for a detailed definition of this key, please refer to §B.4) is "According to the video, which country will host this live stage event? America., Australia., Canada., England." This key contains the question and answer options. In the actual dataset, we remove the quotation marks from the answers. This decision is implemented because fewer punctuation marks ensure more stable model output and facilitate more efficient data processing. During the rephrasing process, we ensure that the answer options remained unchanged.

We input the prompt and the "src" key into GPT-4o, which generates the following three rephrased questions:

1. "Based on the video, where is this live stage event going to take place? America., Australia., Canada., England."

2. "From the video, can you tell which country is the host of this live stage event? America., Australia., Canada., England."

3. "In the video, which country is the live stage event set to take place? America., Australia., Canada., England."

We select the third option as the rephrased question for this data point. Thus, for this particular data point, the "src" key is "According to the video, which country will host this live stage event? America., Australia., Canada., England." and the "rephrase" key (i.e., the rephrased question) is "In the video, which country is the live stage event set to take place? America., Australia., Canada., England."

B.2 Construction Process of Pred key

We utilize the LLaVA-NeXT-Video (7B) model to generate responses for the original dataset, and subsequently select incorrect answers to serve as the "pred" key in the dataset. It is worth noting that when we input questions to the model, what we actually input is [prompt] + src, which is used to generate the model's response and presented in Figure 6.

B.3 Detailed Description of Video Perturbation

B.3.1 Hardware and Software Environment

Our video perturbation operations are executed on a high-performance computing platform with the

Type	Level	Number of Instances
Object-Level Understanding	Basic	685
Action-Level Understanding	Intermediate	802
Domain-Specific Understanding	Specialized	273
Contextual Understanding	Advanced	228
Total	–	**1988**

Table 3: Composition of the **VMEB** Dataset. The dataset is categorized into four levels according to the depth of understanding required: Basic, Intermediate, Specialized and Advanced.

Perturbations
`Horizontal mirroring`
`2x playback speed`
`4x playback speed`
`90-degree rotation`
`180-degree rotation`
`270-degree rotation`
`Grayscale conversion`

Table 4: All processing is based on FFmpeg, without loss of semantics.

[SYSTEM]

- You are an assistant that strictly adheres to formatting requirements

[USER]

- Strict Instructions:
  1. Original question: {src}
  2. Correct answer: {pred}
- Generation Requirements:
- {rephrase}: Restate the question using different grammar while keeping the answer unchanged. Must be in English, and the options after the question must remain unchanged.
- Example input:
  "What is the action performed by the person in the video?['rocking', 'playing fun', 'child speaking', 'performing']"
- Example output:
  "What kind of the action did the person do?['rocking', 'playing fun', 'child speaking', 'performing']"
- Output Format:
  Return strictly JSON format: {{"rephrase": “rephrased_question"}}

Figure 5: Prompt for generating rephrased question, where {src} and {pred} is replaced by the actual data in the editing dataset.

following specifications:

- CPU: 14 cores
- Memory: 100 GB
- GPU: NVIDIA A800 80GB PCIe (1 unit)

The implementation leverages a Python-based processing framework with the environment specifications in Table 5.

The entire video processing pipeline is fundamentally built upon FFmpeg as the core processing engine. FFmpeg is utilized through Python's subprocess module to execute various transformation operations on the video files. This architecture allows us to leverage FFmpeg's powerful video processing capabilities while maintaining precise control over the perturbation parameters through our Python framework. The system incorporates robust error handling mechanisms, including automatic retries with exponential backoff, ensuring reliable processing even when handling large video datasets.

Component	**Version/Details**
Python	Python 3.x
Core Libraries	multiprocessing, subprocess, os, glob
Progress Visualization	tqdm
Error Handling	logging
Parallel Processing	Dynamic allocation

Table 5: Python Environment Configuration

[SYSTEM]

- You are a professional assistant.

[USER]

- Now you need to answer a question. The question contains options which you should choose from, for example:
  Question: "What is the action performed by the person in the video?" bathing, watering, washing, bubbling.
- ASSISTANT: bathing
- Please strictly answer according to the example format, only output the answer, do not add explanations.Answer without quotes.
- Question: {src}

Figure 6: Prompt for generating the answer(pred), where {src} is replaced by the actual datas in the editing dataset.

B.3.2 Perturbation Design Principles

Our video perturbation methodology is carefully designed to preserve semantic integrity while introducing controlled variations. We implement a content-aware approach with the following constraints:

- For videos containing questions about color attributes, grayscale processing is explicitly excluded to maintain critical color information necessary for accurate question answering.
- For videos related to spatial understanding, transformations such as mirror flipping and rotations (90°, 180°, 270°) are excluded to preserve essential spatial relationships.
- For all remaining videos, we employ a randomized perturbation strategy by selecting one processing method from a pool of seven distinct techniques: horizontal mirror flipping, 2x speed acceleration, 4x speed acceleration, grayscale conversion, and three rotation angles (90°, 180°, 270°).

This selective approach ensures that perturbations challenged model robustness without compromising the fundamental semantic content necessary for accurate comprehension. The implementation employs an efficient multiprocessing architecture that can be dynamically allocated computational resources based on system capabilities, with built-in error handling and recovery mechanisms to ensure processing reliability.

B.3.3 Perturbation Examples

The complete range of our perturbation techniques is visually documented in Figure 7, which presents examples of all perturbation methods applied to sample video frames. Each example illustrates the visual transformation introduced by the corresponding perturbation technique, providing a comprehensive visualization of the modifications applied throughout our experimental process.

The processing pipeline incorporates quality control measures, including verification of output file integrity and size optimization through adaptive compression, ensuring consistent quality across the processed dataset while maintaining reasonable file sizes.

B.4 Dataset Composition Examples

Our dataset is designed for video-centric model editing tasks and comprises three main components: train.json, eval.json, and associated video files. The dataset is built upon the VLKEB engineering framework, maintaining compatibility with its structure while introducing our specific modifications.

B.4.1 Data Format

Each entry in both train.json and eval.json follows an identical structure. Table 6 outlines the keys and their corresponding meanings.

B.4.2 Implementation Note

While our codebase is derived from VLKEB's source code, we have maintained the requirement for an eval_multihop.json file to ensure compatibility. This file contains identical content to eval.json but is not utilized in our experimental procedures. We opt not to modify VLKEB's relevant source code in the interest of development efficiency.

B.4.3 Quantity Correspondence

As shown in Table 6, each data entry contains three video addresses. The m_loc video in a given entry

Key	Description
src	Question with accompanying options
rephrase	Rephrased question with accompanying options
pred	Model-generated original answer (y_o), representing the incorrect answer
alt	Correct answer (y_e)
video	Relative path to the video (v_e), stored as a string
video_rephrase	Relative path to the perturbed video (v_r), stored as a string
loc	Common-sense question unrelated to editing, used to evaluate model's text-locality
loc_ans	Answer to the common-sense question
m_loc	Path to an additional video (v_l), stored as a string
m_loc_q	Question corresponding to video v_l, used to evaluate model's video-locality
m_loc_a	Answer to the question about video v_l

Table 6: Dataset Schema and Key Descriptions

may or may not be included among the video or video_rephrase keys of other entries in the dataset. Specifically, an m_loc video might appear exclusively in its own entry, or it might also appear as a video or video_rephrase in other data entries. Consequently, the total number of unique videos in the dataset slightly exceeds twice the total number of data entries.

C Detailed Experimental Steps

C.1 Experimental Platform and Environment

In this section, we provide a comprehensive overview of our experimental setup to ensure reproducibility of our results.

C.1.1 Hardware Configuration

Our experiments are conducted on a high-performance computing platform with the following specifications:

Component	Specification
CPU	14 cores
Memory	100 GB RAM
GPU	NVIDIA A800 80GB PCIe $\times$ 1

C.1.2 Software Environment

All experiments are implemented using Python 3.9.7 in a Conda environment. The major software components and their versions are listed below:

Software	Version
PyTorch	2.0.1
Transformers	4.49.0
CUDA Libraries	11.7
PEFT	0.7.1
Flash Attention	2.6.1
Accelerate	1.5.2
Datasets	1.18.3
NumPy	1.22.1
OpenCV-Python	4.8.0.76
AV (PyAV)	14.2.0

Additional dependencies include scikit-learn (1.0.2), pandas (1.4.0), and various utilities for data processing and model optimization. Our environment utilizes optimized CUDA libraries with cuBLAS, cuDNN, and other NVIDIA performance libraries to accelerate computations on the GPU. The complete environment configuration is available in our repository for comprehensive reproducibility.

C.2 Code Declaration

This research implementation builds upon the VLKEB framework (Huang et al., 2024a). Our codebase extends the original implementation with modifications to support the methodologies described in this paper. The original VLKEB project is distributed under the Apache 2.0 license, which permits adaptation and modification with appropriate attribution. All our modifications maintain compliance with the terms of this license.

The primary adaptations to the original framework include enhancements to support our video-language model editing methodology, dataset processing components, and evaluation procedures. The architecture of our implementation preserves the core mechanisms of VLKEB while introducing the novel components necessary for Vid-LLMs editing described in our work.

C.3 Parameters for Model Editing

This section contains the detailed configuration parameters used in our experiments. We present detailed tables corresponding to different model editing methods and their training/evaluation settings. The calculation formula for loss is as follows:

$$\begin{aligned} loss_{total} = & \ c_{edit} \times loss_{edit} \\ & + c_{loc} \times \left(loss_{loc}^{text} + loss_{loc}^{video}\right) \\ & + i_{edit} \times loss_{edit}^{video} \end{aligned} \tag{6}$$

where c_{edit}, c_{loc}, and i_{edit} respectively adjust the weights of different metrics.

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
Base Learning Rate	1e-5	1e-5	5e-7
Edit Learning Rate	1e-2	1e-2	1e-5
Optimizer	Adam	Adam	Adam
Gradient Clip	100.0	100.0	1.0
Batch Size	1	1	1
Sentence Encoder	all-mpnet-base-v2	all-mpnet-base-v2	all-mpnet-base-v2
c_{edit}	0.1	0.1	0.1
i_{edit}	0.1	0.1	0.1
c_{loc}	1.0	1.0	1.0

Table 7: SERAC training parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
Base Learning Rate	1e-5	1e-5	5e-7
Edit Learning Rate	1e-2	1e-2	1e-5
Batch Size	1	1	1
Evaluation Only	True	True	True

Table 8: SERAC evaluation parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
Editing Layers	25-27	25-27	29-31
Base Learning Rate	1e-6	1e-6	5e-7
Edit Learning Rate	1e-4	1e-4	1e-5
Optimizer	Adam	Adam	Adam
Gradient Clip	50.0	50.0	1.0
Batch Size	1	1	1
c_{edit}	0.1	0.1	0.1
i_{edit}	0.1	0.1	0.1
c_{loc}	1.0	1.0	1.0

Table 9: MEND training parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
Editing Layers	25-27	25-27	29-31
Base Learning Rate	1e-6	1e-6	5e-7
Edit Learning Rate	1e-4	1e-4	1e-5
Batch Size	1	1	1
Evaluation Only	True	True	True

Table 10: MEND evaluation parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
Editing Layers	27	27	31
Base Learning Rate	1e-6	1e-6	1e-6
Edit Learning Rate	1e-4	1e-4	1e-4
Optimizer	Adam	Adam	Adam
Gradient Clip	100.0	100.0	100.0
Batch Size	1	1	1

Table 11: Fine-Tuning parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B/3B	8B	7B
k (Context Size)	27	27	27
Sentence Model	all-MiniLM-L6-v2	all-MiniLM-L6-v2	all-MiniLM-L6-v2

Table 12: IKE parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B / 3B	8B	7B
Optimization Steps	25 (7B) / 50 (3B)	25	25
Learning Rate	0.1	0.1	0.1
Weight Decay	0.5	0.5	0.5
KL Factor	0.05	0.05	0.05
Clamp Norm Factor	0.75	0.75	0.75

Table 13: AlphaEdit parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models. Note that layers are selected with a stride of 2.

Parameter	Qwen2.5-VL	Qwen3-VL	LLaVA-NeXT-Video
Model Size	7B / 3B	8B	7B
Optimization Steps	25	25	25
Learning Rate	0.1	0.1	0.1
Weight Decay	0.5	0.5	0.5
KL Factor	0.05	0.05	0.05
Clamp Norm Factor	0.75	0.75	0.75

Table 14: MEMIT parameters for Qwen2.5-VL, Qwen3-VL and LLaVA-NeXT-Video models. Note that layers are selected with a stride of 2.

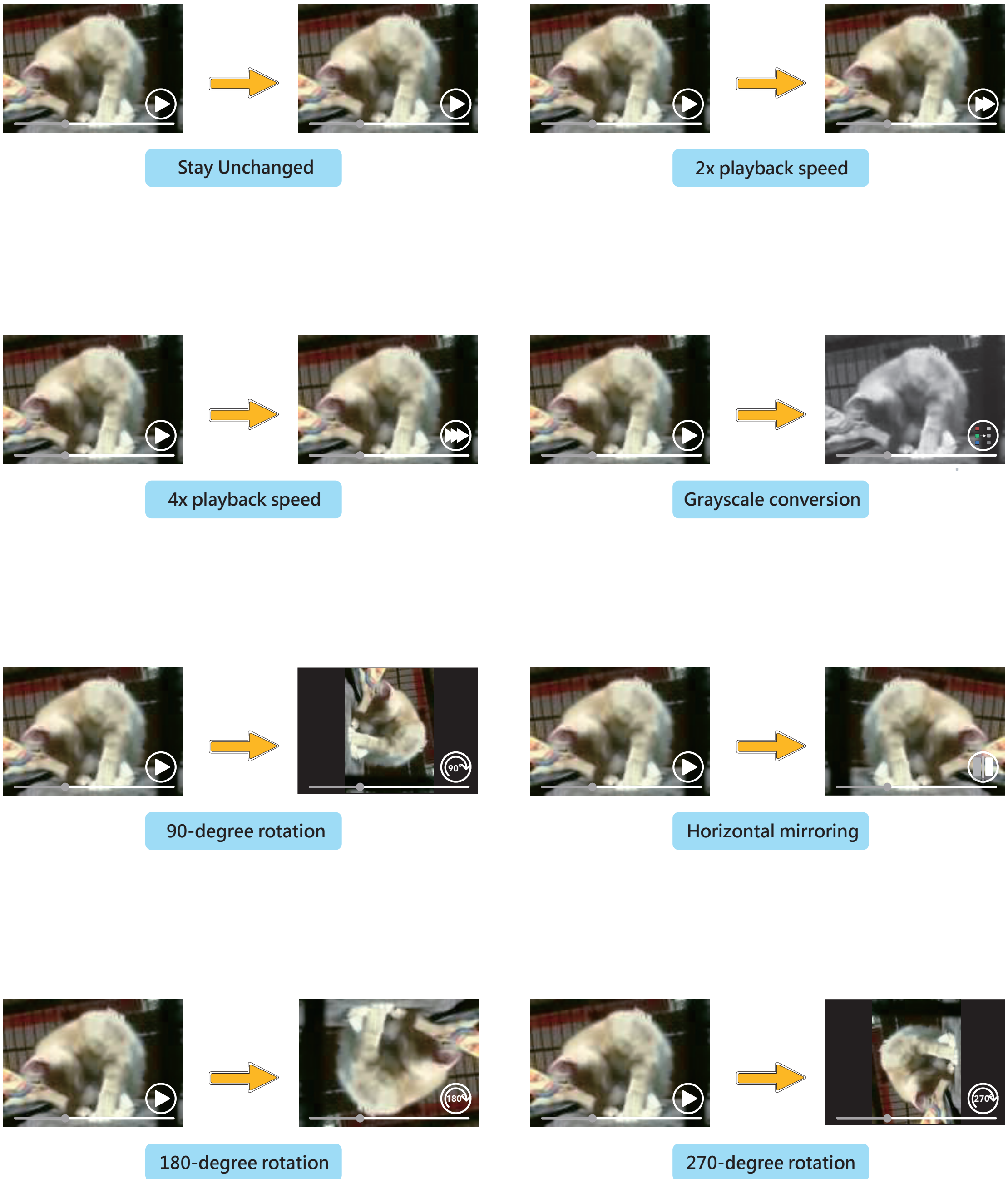

Figure 7: Eight ways of normal perturbation to videos