Editing the Moving World: Model Editing for Video LLMs

Anonymous ACL submission

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 four representative model editing methods (FT, IKE, MEND, and SERAC) 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. 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, among which MEND demonstrates superior performance, and identify new challenges and research directions for the future development of the model editing field.

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 concentrated on updating factual triples, with approaches such as ROME (Meng et al., 2022) 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, procedures, and free-form text (Peng et al., 2024; Liu et al., 2024; Deng et al., 2025; Jiang et al., 2025).

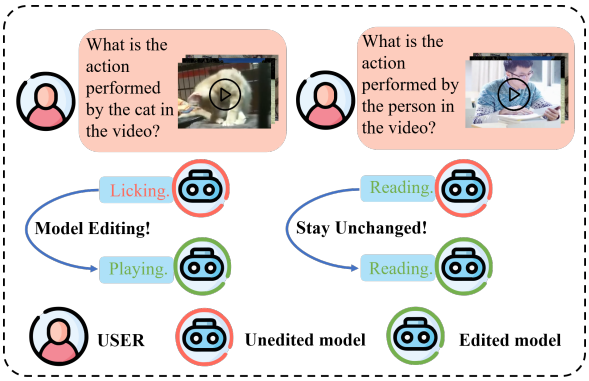

Figure 1: Overview of the Vid-LLMs editing task. The goal is to update the model's understanding of a specific video-text input. **Red-colored answers** indicate suboptimal outputs that require editing, while **green-colored answers** represent correct responses.

The paradigm was 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 Model Editing to Vid-LLMs is thus a timely yet non-trivial challenge: edits must account for complex motion patterns, higher-level abstractions, and broader temporal generalization.

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. **VMEB** systematically assesses editing performance in three widely used Vid-LLMs of different scales—LLaVA-NeXT-Video (7B) (Zhang et al., 2024) and Qwen2.5-VL (3B & 7B) (Bai et al., 2025). Evaluation covers the existing dimensions (Yao et al., 2023; Cheng et al., 2024) of **Reliability**, **Locality**, and **Generality**, while introducing video-specific axes that probe **Robustness**. We adapt representative editing methods (FT, IKE (Zheng et al., 2023), MEND (Mitchell et al., 2022a) and SERAC (Mitchell et al., 2022b)) to several Vid-LLMs, emphasizing edits that transcend conventional factual updates or static multimodel model editing. We hope **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 dynamic multimodal systems.

Experimental results demonstrate that the MEND method exhibits superior performance across all evaluation dimensions, achieving effective editing while preserving model integrity. Although SERAC shows relatively inferior editing performance, it causes minimal degradation to the model's original capabilities. In contrast, both FT and IKE methodologies inflict significant destructive impact on the models. Additionally, video-level metrics show a notable decline compared to text-level metrics, highlighting a notable gap in current mainstream model editing approaches when applied to Vid-LLMs.

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.

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) paired 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 shifted 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.

More recent efforts focus on unifying image/video understanding and handling long-duration videos. **LLaVA-Next-Video** (Zhang et al., 2024) treats sparse frames as a "long image," allowing a 7B vision–language backbone to inherit image skills and gain temporal reasoning with minimal extra cost. Concurrently, models like MovieChat (Song et al., 2024) address minute- to hour-long clips. **Qwen2.5-VL** (Bai et al., 2025), by scaling parameters and context with techniques like dynamic-resolution windowing and absolute-time encoding, achieves state-of-the-art grounding on long-video benchmarks while preserving conversational quality.

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 Model 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. Recently, growing interests extends the Model Editing to MLLMs.

Intrinsic Editing. Intrinsic methods embed new knowledge by modifying a model's internal

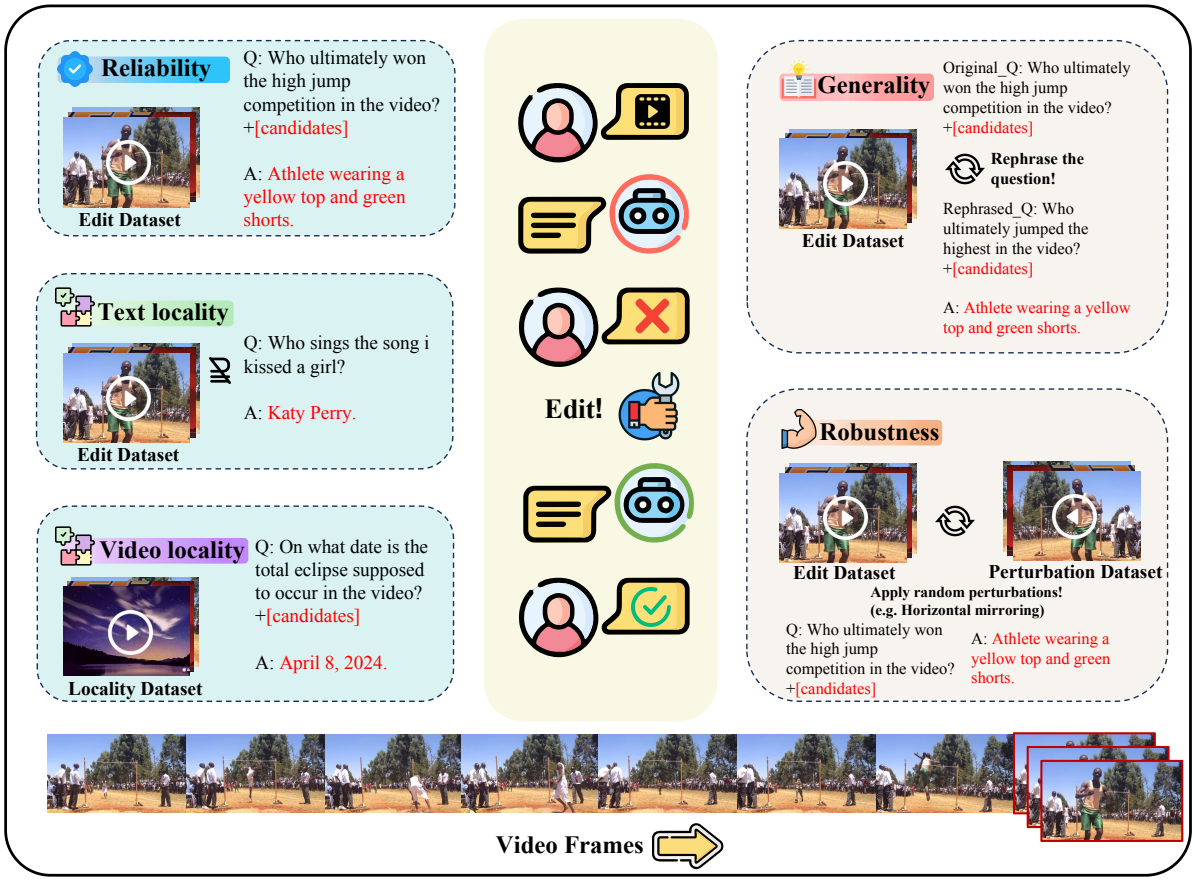

Figure 2: Framework of our Vid-LLMs Editing Tasks

weights or activations. Early fine–tuning delivers reliable edits but is computationally heavy and risks catastrophic forgetting. Meta-learning approaches such as MEND (Mitchell et al., 2022a) and MALMEN (Tan et al., 2024) introduce lightweight hypernetworks that predict parameter deltas conditioned on an edit request. Locate-then-edit techniques like ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) trace the causal pathway of the target fact, then overwrite only the salient components, achieving high locality with minimal collateral damage.

Extrinsic Editing. Extrinsic strategies inject knowledge via inputs or outputs without changing backbone weights. IKE (Zheng et al., 2023) uses few-shot prompts to edit via in-context learning. Memory-augmented systems, such as SERAC (Mitchell et al., 2022b) and MeLLo (Zhong et al., 2023), maintain a separate store of counterfactual knowledge: SERAC couples a gating classifier with an editable side network, whereas MeLLo refines facts through iterative prompting, making it suitable for multi-hop reasoning. These approaches are computationally light but face context-length and retrieval challenges.

Multimodal Editing. The editing paradigm was 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). 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., 2025), 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), dataset construction details (§3.2), the Vid-LLMs we chose (§3.3), and the editing methods (§3.4) we used in the experiments.

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:

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

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., 2022; Yao et al., 2023).They are calculated as follows:

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

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

where $\mathcal{D}^{\text{t}}_{\text{loc}}$ and $\mathcal{D}^{\text{v}}_{\text{loc}}$ 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:

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

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. Studies have shown that robustness to visual variations is particularly challenging in edited multimodal models, as minor visual perturbations can significantly affect model outputs despite maintaining semantic equivalence (Elsayed et al., 2018). In the domain of model editing, we aim for edits to remain effective when the model processes videos perturbed by random noise compared to those used during editing.

To quantify this capability, we introduce the $\mathcal{M}_{rob}$ metric, which evaluates both the robustness

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

of the editing method and the generalization capability of the edited model. This metric is calculated 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, while v_r represents the new video obtained by applying random perturbation processing to the original video.

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 1580 data entries and their corresponding videos, with the videos categorized into 14 distinct subcategories. For detailed classification, please refer to figure 3.

3.2.1 Edit Dataset.

We began 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 concatenated the original questions and options for the videos as x_e , with the videos serving as v_e. We tested MVBench and Video-MME using the LLaVA-NeXT-Video(7B) model and retained the incorrectly answered responses as y_o, which represent the original outputs requiring editing, while the correct answers for these entries were designated as y_e. While all entries with correct answers were discarded from the dataset.

3.2.2 Locality Dataset.

We must evaluate the model's normal performance after editing, therefore we decided to use commonsense question-answering from Wikipedia to assess textual locality. We randomly selected 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 reshuffled the dataset using a shuffling algorithm and created 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 also be successfully edited. To evaluate this generalization capability, we utilized GPT-4O to generate high-quality question restatements, which were subsequently manually reviewed. This review process ensured that the restated questions x_r maintained 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 2, thereby obtaining v_r.

3.3 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., 2025) is a flagship multimodal LLM series particularly strong in long-video

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 1: 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 2: All processing is based on FFmpeg, without loss of semantics.

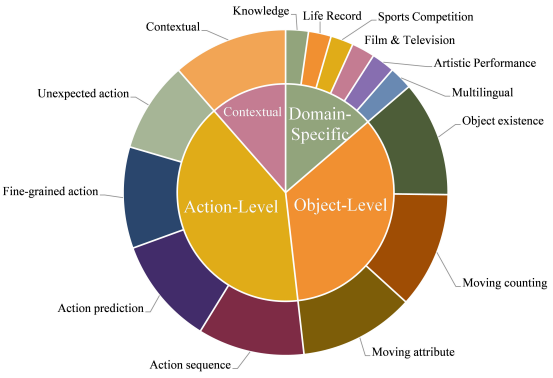

Figure 3: Dataset Composition with Primary and Secondary Categories

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.

3.4 Editing Methods

3.4.1 Fine tuning

Fine tuning is the most fundamental method in intrinsic editing methods. It mainly updates the parameters of the model through gradient descent to achieve the purpose of changing certain characteristics of the model.

3.4.2 In-Context Knowledge Editing(IKE)

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

3.4.3 MEND

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

3.4.4 SERAC

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

4 Experiments

In this section, we conducted experiments of **FT**, **IKE**, **MEND**, and **SERAC** on **VMEB**. With an equal number of training steps across methods, the test results are presented in Table 3, where we have highlighted both the superior and anomalous data points for ease of reference. In §4.1, we provide a comparative analysis of how different editing methods perform in terms of **Reliability**, **Locality**, **Generality**, and **Robustness**, as well as the variations between different models. §4.2 focuses on analyzing the implications of extending model editing to video applications. Figure illustrates the differences between textual indicators and video-level indicators, which forms the basis for our discussion on future challenges for model editing techniques.

4.1 Results

From a macro perspective, MEND demonstrates the best overall performance, while FT and IKE both show tendencies to damage the model, and SERAC exhibits poorer editing effectiveness than

Table 3: 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 could be highlighted in bold, while second-best results could be underlined if criteria were defined. The data highlighted in red indicates that the method performs poorly on that metric, which would degrade model performance.

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	**99.96**	79.93	22.79	**99.92**	**99.96**
	IKE	87.68	42.26	24.49	86.39	87.87
	MEND	96.31	99.02	85.11	96.12	96.21
	SERAC	80.23	**99.87**	**96.39**	79.51	78.98
Qwen2.5-VL *Model Size: 3B*	Base Model	0.00	100.00	100.00	0.00	0.00
	FT	**100.00**	94.03	55.90	**99.84**	**100.00**
	IKE	77.90	71.06	21.41	79.36	77.74
	MEND	99.05	98.53	74.85	98.46	98.73
	SERAC	83.27	**99.35**	**89.75**	82.71	80.17
Qwen2.5-VL *Model Size: 7B*	Base Model	0.00	100.00	100.00	0.00	0.00
	FT	**100.00**	74.18	9.92	**99.81**	**99.95**
	IKE	85.85	80.75	21.08	79.48	84.96
	MEND	96.97	97.43	71.80	96.18	96.43
	SERAC	75.46	**99.77**	**88.09**	75.57	75.06

MEND. After applying random perturbations to the videos, all metrics showed varying degrees of decline.

Reliability. Based on the results analysis, all methods demonstrated significant improvement in accuracy, indicating effective performance across existing model editing approaches. Among these methods, FT and MEND achieved notably higher accuracy rates, approaching 100%, while IKE and SERAC methods performed relatively poorly with accuracy rates ranging between 70% and 80%. However, it is important to note that compared to other model editing methods, the destructive impact of FT on the model is evident; therefore, high accuracy alone cannot be considered indicative of FT's superiority. Simultaneously, IKE, as a method that does not modify model parameters, exhibited the lowest accuracy in most scenarios, highlighting the limitations of prompt-injection-based approaches. Furthermore, SERAC, despite requiring training, did not demonstrate significant improvement over IKE in video-level editing, indicating SERAC's relative insufficiency in this domain. In contrast, MEND performed exceptionally well, showing no significant decline in performance when extended to video-level editing while ensuring the model remained undamaged, thus demonstrating excellence across multiple evaluation metrics.

Locality. Under this metric, all methods exhibited varying degrees of performance decline compared to the Base model, demonstrating that existing model editing approaches cannot yet achieve modification without affecting the model's performance. However, SERAC and MEND demonstrated excellent performance at the text-level, indicating that the foundational language model's capabilities remained largely unaffected by the editing process. In contrast, both FT and IKE methodologies showed metrics predominantly below 90% at both text-level and video-level, suggesting these approaches introduce slight degradation to the model. Notably, IKE, despite being a method that does not alter model parameters, still demonstrated significant impact on the model's performance. This finding indicates that models can potentially be "confused" by prompts, resulting in outputs that deviate from expected outcomes.

Generality. Comparing different methods, both MEND and FT methodologies demonstrate accuracy rates approaching 100%; however, FT's performance comes at the expense of compromising the

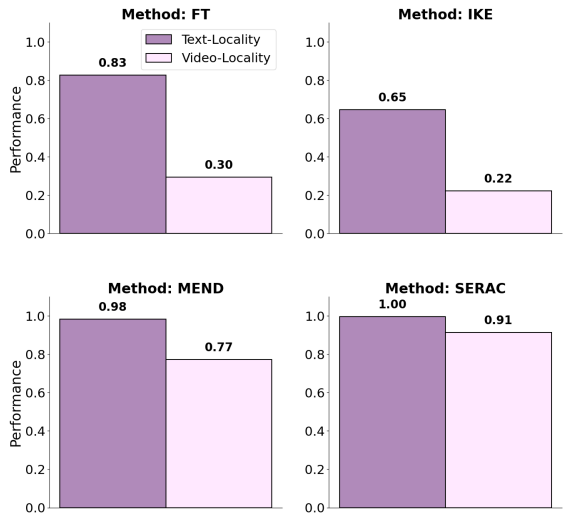

Figure 4: Video-level metrics by average demonstrate a significant degradation compared to the text-level metrics by average.

model's standard functionality. In contrast, MEND substantially preserves the model's inherent capabilities while maintaining excellent generality, ensuring that edits remain effective across similar queries. This finding suggests that MEND potentially enables the model to "comprehend" the edited knowledge at a deeper level. In contrast, IKE and SERAC exhibit relatively poor performance, primarily because both methods rely on the model's inherent generalization ability (IKE depends on Vid-LLMs while SERAC relies on the binary classification model)—specifically, the ability to recognize different expressions of the same problem.

Robustness. When considering all four methods collectively, they exhibit similar patterns for both reliability and robustness metrics. FT and MEND demonstrate consistently higher scores, whereas IKE and SERAC exhibit comparatively lower performance metrics. Similar performance patterns between robustness and reliability are expected, as the video perturbations applied are not overly severe and preserve the majority of visual semantics. Consequently, methods that achieve better editing performance also excel in the robustness metric.

4.2 Video-Level Editing: Great Performance, Yet Requires Further Investigation

As shown in Figure 4, the video-level metrics demonstrate varying degrees of decline compared to text-level metrics, highlighting the inherent complexity of multimodal editing. However, the performance degradation from text to image (Huang et al., 2024a; Cheng et al., 2024) and subsequently to video is not substantial. In our experiment, we only edited the MLP layers in the last three layers, without editing the corresponding visual layers or projection layers. This means that our edits did not directly target visual semantics, yet they achieved the effect of editing visually. This raises a fundamental question: Can model editing truly teach models to "understand" visual semantics and transformations in spatial logical relationships? Based on our analysis results and the progression of model editing work from text to image (Huang et al., 2024a; Cheng et al., 2024) to video, we conclude that model editing tends to operate more in a probabilistic sense. During the editing process, models likely learn statistical patterns present in the training data rather than developing true understanding. Another possibility is that these models may not strictly differentiate between modalities at higher representational levels, potentially forming a shared semantic space where concepts transcend their original modalities. Therefore, we conclude that these effects cannot be attributed to the model's comprehension of changes in visual semantics, but rather to complex interactions within the model's representation space. This presents an entirely new challenge for the future of model editing: developing editing methods capable of distinguishing genuine cross-modal understanding in multimodal representations, as well as creating novel benchmarks to evaluate such editing methods.

5 Conclusion

In this paper, we propose a novel benchmark called **VMEB**. We have adapted model editing methods, such as **IKE**, **MEND**, and **SERAC**, to Vid-LLMs and have implemented them for the **Qwen2.5-VL** and **LLaVA-NeXT-Video** models, representing the first attempt at model editing in the video domain. Then, we analyzed the experimental results and formulated the following conclusion: Among the four editing methodologies examined—FT, IKE, SERAC, and MEND—MEND demonstrated superior performance across all evaluation dimensions. While SERAC exhibited suboptimal editing effectiveness, it caused minimal degradation to the model's integrity. In contrast, both FT and IKE methodologies resulted in demonstrably destructive impacts on the model's performance. Finally we offer some perspectives on the future development of the model editing field.

Limitations

Model Selection. In our experiment, we selected only two types of Vid-LLMs, without considering the majority of other models. Additionally, we only selected open-source large models, while editing closed-source large models remains an unresolved challenge. The models we selected were all below 10B parameters, and we did not test larger models, such as the 72B version of LLaVA-NeXT-Video (Zhang et al., 2024) or Qwen2.5-VL (Bai et al., 2025).

Editing Layers. For model editing, we only preliminarily attempted application editing at the language layer, while ignoring other model structures. Editing only the language layer may not enable the model to "understand" the edited content, potentially resulting in poor actual editing effectiveness.

Perturbation Methods. During the video processing, our disturbance method preserved the complete semantics of the videos without attempting to use stricter perturbations such as black line masking or blurring. This approach resulted in robustness metrics not declining significantly compared to reliability metrics.

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, 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. 2025. 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.

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.

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.

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. 2024. 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.

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. 2022. 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*.

Chenmien Tan, Ge Zhang, and Jie Fu. 2024. Massive editing for large language models via meta learning. In *International Conference on Learning Representations*.

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.

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.

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.

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.

Zexuan Zhong, Zhengxuan Wu, Christopher D. Manning, Christopher Potts, and Danqi Chen. 2023. Mquake: Assessing knowledge editing in language models via multi-hop questions. *CoRR*, abs/2305.14795.

A Dataset Construction Details

The dataset is partitioned according to the following specifications:

- **Train-Evaluation Split**: 7:3 ratio
- **Training Set** (train.json): 1,108 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 the §A.4.2.)

A.1 Construction Process of Rephrased Questions

A.1.1 Specific Construction Method

We utilize the GPT-4o model to construct rephrased questions, with the prompt design format illustrated in the 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.

A.1.2 Data Example

We randomly selected 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 §A.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 removed the quotation marks from the answers. This decision was implemented because fewer punctuation marks ensure more stable model output and facilitate more efficient data processing. During the rephrasing process, we ensured that the answer options remained unchanged.

We input the prompt and the "src" key into GPT-4o, which generated 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.,

[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 datas in the editing dataset.

Canada., England."

We selected 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."

A.2 Construction Process of Pred key

We utilized the LLaVA-NeXT-Video (7B) model to generate responses for the original dataset, and subsequently selected 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.

A.3 Detailed Description of Video Perturbation

A.3.1 Hardware and Software Environment

Our video perturbation operations were executed on a high-performance computing platform with the following specifications:

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

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

Table 4: Python Environment Configuration

The implementation leveraged a Python-based processing framework with the environment specifications in Table 4.

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

[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 rephrased question, where {src} and {pred} is replaced by the actual datas in the editing dataset.

A.3.2 Perturbation Design Principles

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

- For videos containing questions about color attributes, grayscale processing was 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°) were excluded to preserve essential spatial relationships.
- For all remaining videos, we employed 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 ensured that perturbations challenged model robustness without compromising the fundamental semantic content necessary for accurate comprehension. The implementation employed an efficient multiprocessing architecture that dynamically allocated computational resources based on system capabilities, with built-in error handling and recovery mechanisms to ensure processing reliability.

A.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 incorporated 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.

A.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.

A.4.1 Data Format

Each entry in both train.json and eval.json follows an identical structure. Table 5 outlines the keys and their corresponding meanings.

A.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 opted not to modify VLKEB's relevant source code in the interest of development efficiency.

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 5: Dataset Schema and Key Descriptions

A.4.3 Quantity Correspondence

As shown in Table 5, each data entry contains three video addresses. The m_loc video in a given entry 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.

B Detailed Experimental Steps

B.1 Experimental Platform and Environment

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

B.1.1 Hardware Configuration

Our experiments were 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

B.1.2 Software Environment

All experiments were 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 utilized 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.

B.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 frame-

work 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.

Our code will be made publicly available on GitHub for research purposes.

B.3 Parameters for Model Editing

This section contains the detailed configuration parameters used in our experiments. We present six 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	LLaVA-NeXT-Video
Model Size	7B/3B	7B
Editing Layers	25-27	29-31
Base Learning Rate	1e-5	5e-7
Edit Learning Rate	1e-2	1e-5
Optimizer	Adam	Adam
Gradient Clip	100.0	1.0
Batch Size	1	1
Sentence Encoder	all-mpnet-base-v2	all-mpnet-base-v2
c_{edit}	0.1	0.1
i_{edit}	0.1	0.1
c_{loc}	1.0	1.0

Table 6: SERAC training parameters for Qwen2.5-VL and LLaVA-NeXT-Video models

Parameter	Qwen2.5-VL	LLaVA-NeXT-Video
Model Size	7B/3B	7B
Editing Layers	25-27	29-31
Base Learning Rate	1e-5	5e-7
Edit Learning Rate	1e-2	1e-5
Batch Size	1	1
Evaluation Only	True	True

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

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

Table 8: MEND training parameters for Qwen2.5-VL and LLaVA-NeXT-Video models

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

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

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

Table 10: Fine-Tuning parameters for Qwen2.5-VL and LLaVA-NeXT-Video models

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

Table 11: IKE parameters for Qwen2.5-VL and LLaVA-NeXT-Video models

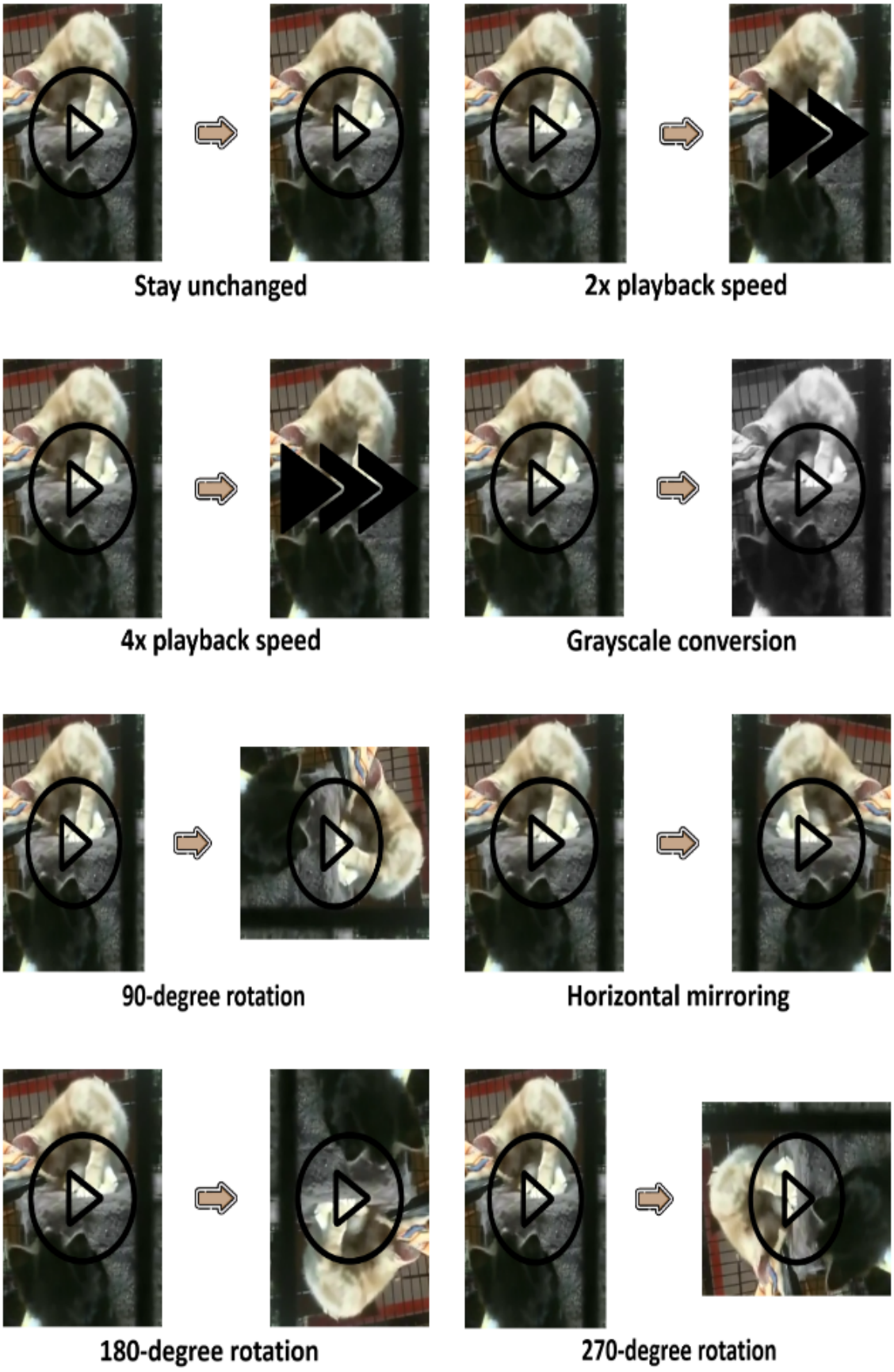

Figure 7: Perturbation Examples