



OpenVE-3M: A Large-Scale High-Quality Dataset for Instruction-Guided Video Editing Supplementary Material

Haoyang He^{1,2*} Jie Wang^{2*} Jiangning Zhang^{1†} Zhucun Xue¹
Xingyuan Bu² Qiangpeng Yang^{2†} Shilei Wen² Lei Xie^{1†}

¹ Zhejiang University ² ByteDance

Abstract. The quality and diversity of instruction-based image editing datasets are continuously increasing, yet large-scale, high-quality datasets for instruction-based video editing remain scarce. To address this gap, we introduce OpenVE-3M, an open-source, large-scale, and high-quality dataset for instruction-based video editing. It comprises two primary categories: spatially-aligned edits (Global Style, Background Change, Local Change, Local Remove, Local Add, and Subtitles Edit) and non-spatially-aligned edits (Camera Multi-Shot Edit and Creative Edit). All edit types are generated via a meticulously designed data pipeline with rigorous quality filtering, a process consuming in excess of 10,000 GPU-days. OpenVE-3M surpasses existing open-source datasets in terms of scale, diversity of edit types, instruction length, and overall quality. Furthermore, to address the lack of a unified benchmark in the field, we construct OpenVE-Bench, containing 431 video-edit pairs that cover a diverse range of editing tasks with three key metrics highly aligned with human judgment. We present OpenVE-Edit, a 5B model trained on our dataset that demonstrates remarkable efficiency and effectiveness by setting a new state-of-the-art on OpenVE-Bench, outperforming all prior open-source models including a 14B baseline.

Keywords: Instruction-guided Video Editing · Dataset and Benchmark

1 Introduction

Recently, **I**nstruction-guided **I**mage **E**ding (IIE) models: FLUX-Kontext [14], Qwen-Image-Edit [29], and Nano Banana [8] have demonstrated powerful editing capabilities. The success of an increasing number of data-driven, open-source image editing models can be attributed to the growing availability of high-quality, open-source, IIE datasets [20, 35]. However, **I**nstruction-guided **V**ideo **E**ding (IVE) models and datasets remain *under-explored*. This is mainly due to the

* Equal Contribution.

† Corresponding authors.

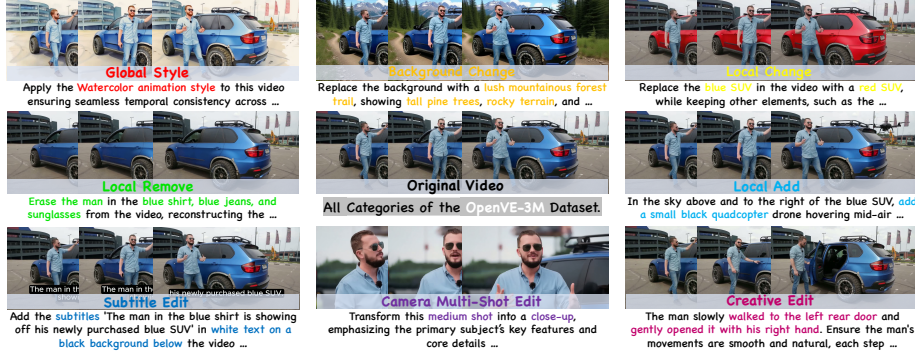


Fig. 1: Demonstration of **Eight different categories** on the same video from the proposed **OpenVE-3M** dataset.

significant difficulty in constructing high-quality IVE datasets. A key challenge is the need to maintain not only spatial consistency but also temporal consistency, which significantly increases the construction difficulty. Furthermore, the lack of effective data filtering strategies results in a high proportion of bad cases, making it difficult to completely eliminate poor-quality samples.

Existing IVE datasets, such as InsViE-1M [30], Seniorita-2M [40], and Ditto-1M [2], suffer from four main issues: **small scale, limited edit types, short instructions, and poor editing quality**. Tab. 1 presents a quantitative analysis of existing open-source IVE datasets. Although VIVID [11] contains 10M samples, it cannot be directly used for training as it only provides video masks instead of the edited videos. In contrast, InsViE-1M [30], Seniorita-2M [40], and Ditto-1M [2] contain only 1-2M samples and offer a limited variety of editing types. Fig. 2(a) shows the distribution of instruction lengths. The average instruction length in [30] and [40] is only about 4 words, which is insufficient to provide precise editing guidance and thus negatively impacts performance. To assess dataset quality, we input the original video, edited video, and instruction into Gemini 2.5 Pro [8] for evaluation on a 1-to-5 scale across three dimensions: *Instruction Compliance*, *Consistency & Detail Fidelity*, and *Visual Quality & Stability*. For each dataset, we randomly sample 50 edited pairs from each category. The final score distributions are shown in Fig. 2(b). Although [30] and [40] show a notable number of 5-point scores, they also have a high proportion of 1-point bad cases, resulting in a low average quality score. [2] demonstrates decent

Table 1: Comparison with current IVE datasets. Cat./Avg. Ins. Lgth refers to Categories/Average Instruction Lengths.

Datasets	Amount	Cat.	Avg. Ins.	Lgth	Frames	Resolution
EffiVED [38]	155K	4	N/A	8		512×512
InsV2V [7]	404K	4	N/A	16		256×256
VIVID-10M [11]	10M	3	N/A	30		1280×720
InsViE-1M [30]	1M	4	4.4	25		1024×576
Seniorita-2M [40]	2M	6	3.6	33-64		336×592 - 1120×1984
Ditto-1M [2]	1M	3	17.1	101		1280×720
ReCo [2]	1M	3	17.1	101		1280×720
OpenVE-3M(Ours)	3M	8	40.6	65-129		1280×720

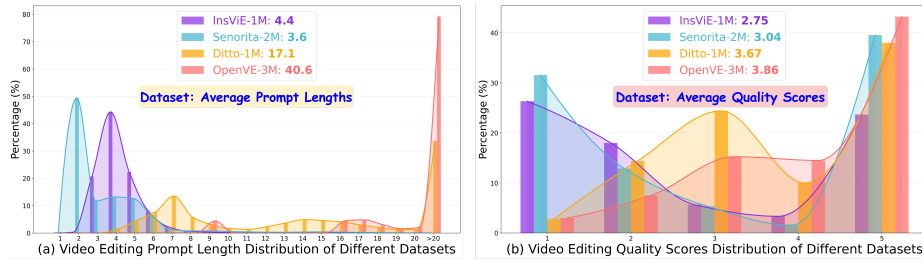


Fig. 2: Comparisons on video statistics between OpenVE-3M and current open-source video editing datasets.

quality, but its edits are primarily focused on style transfer, lacking diversity. In summary, there is a clear need for a large-scale, high-quality, and diverse IVE dataset.

Therefore, we introduce **OpenVE-3M**, a large-scale, high-quality, and multi-category IVE dataset. It comprises **3M** samples categorized into two main types: **Spatially-Aligned (SA)** and **Non-Spatially-Aligned (NSA)**. SA edits ensure that the edited video maintains motion consistency with the original video in both space and time, including six sub-types: *Global Style*, *Background Change*, *Local Change*, *Local Remove*, *Local Add*, and *Subtitles Edit*. NSA edits maintain the primary subject’s consistency but introduce new, creative motion, including two sub-types: *Camera Multi-Shot Edit* and *Creative Edit*. Visual examples for all categories are presented in Fig. 1. Furthermore, OpenVE-3M features the longest average instruction length (40.6 words) and the highest average video editing quality score of 3.86 compared to existing IVE methods in Tab. 1.

The construction of the OpenVE-3M dataset includes 3 stages, each leveraging many models and APIs for high-quality generation: **1)Data Pre-processing**: Preparing data for category-specific generation by employing models such as **Multimodal Large Language Model (MLLM)** [3], detection and segmentation models [23], depth and edge estimation models [4, 6], and local descriptors [16]. **2)Taxonomy-guided Video Pair Generation**: Creating edited video pairs using a variety of models, including IIE model [14], image-to-video model [27] and so on. **3)High-Quality Pair Filtering**: Filtering the generated pairs to retain only high-quality examples using advanced MLLMs [3, 8, 25, 28]. The detailed construction pipeline is described in Sec. 2.

We also propose **OpenVE-Edit**, an IVE model composed of three main components: a MLLM, a **Mixture-of-Experts (MoE) Connector**, and a **Diffusion Transformer (DiT)**. The MLLM processes both the video and the text instruction simultaneously. Unlike approaches that only use a text encoder for the instruction, our MLLM is designed to extract high-level instruction representations while also capturing spatio-temporal relationships within the video. We introduce an MoE-Connector which leverages different expert networks to process features corresponding to various editing types. To enhance training efficiency, we initialize the final linear layer of the experts to zero. This strategy prevents

random features from disrupting the generation process in the early stages of training, thereby improving training stability and efficiency. The detailed model design is described in Sec. 3

Currently, there is no universal benchmark for IVE that is highly aligned with human evaluation. Therefore, we propose **OpenVE-Bench**, a meticulously curated benchmark comprising 431 video pairs across 8 categories. For each category, we design specific evaluation prompts targeting three key dimensions: *Instruction Compliance*, *Consistency & Detail Fidelity*, and *Visual Quality & Stability*. The final score is obtained by feeding the original video, the edited video, and the instruction into a MLLM. Sec. 4 details the benchmark design.

In summary, our contributions are as follows:

- We introduce **OpenVE-3M**, a large-scale, high-quality, and diverse IVE dataset. It contains 3M samples across two major categories (Spatially-Aligned and Non-Spatially-Aligned) with 8 sub-categories.
- We propose a **Robust and Scalable Pipeline** for constructing high-quality, multi-category IVE data, to facilitate further research in the community.
- We develop **OpenVE-Edit**, an efficient and effective IVE model. With only 5B parameters, it achieves SOTA performance, surpassing existing open-source 14B models.
- We establish **OpenVE-Bench**, a universal, multi-category, and challenging benchmark for IVE. It evaluates model performance along three key dimensions and demonstrates high alignment with human judgment.

2 OpenVE-3M: A High-Quality Instruction-Guided Video Editing Dataset

We introduce OpenVE-3M, a large-scale, high-quality, multi-category, and balanced dataset designed for IVE. It comprises eight categories, divided into six **Spatially-Aligned** (SA) and two **Non-Spatially-Aligned** (NSA) types. The full construction pipeline is illustrated in Fig.3.

2.1 Video Clip Pre-Processing

We establish a base video library of 1M videos sourced from three open-source high-quality datasets: Open-Sora-Plan [18], OpenViD-HD [21], and UltraVideo [31]. For each source video, we randomly extract a clip of *65 to 129 frames* and re-size it to *720P* resolution. Qwen2.5-VL-72B [3] is used to generate a long-form description for the video clip while simultaneously extracting the names of detectable objects for subsequent processing. Concurrently, we generate depth and Canny edge maps for the clip using Video DepthAnything [6] and OpenCV [4], respectively. For the extracted object names, we employ Grounded SAM2 [23] to perform detection and segmentation, obtaining bounding boxes and mask videos for all identified objects. In parallel, the bounding boxes and the video clip are fed into the DAM [16] to generate localized descriptions for each object. This process is applied to every source video, and the resulting information is used in the second stage to construct video editing pairs across different categories.

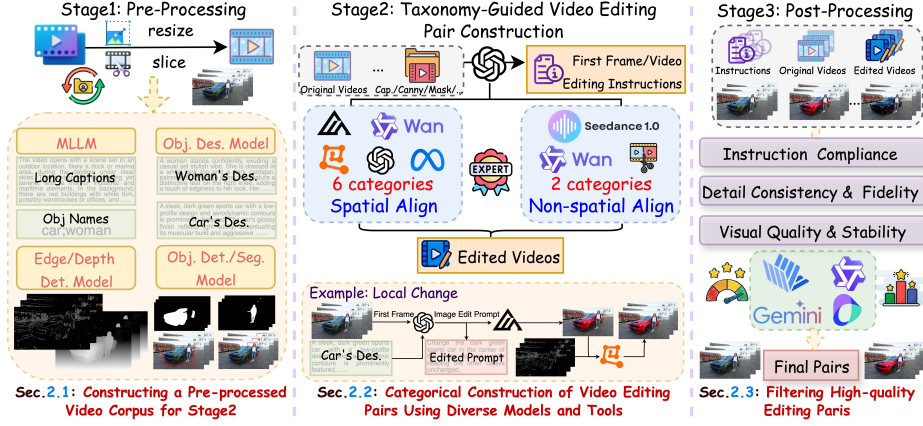


Fig. 3: Overview of Data Pipeline. Stage 1: Aims to construct a video corpus and perform various pre-processing steps in preparation for Stage 2 in Sec. 2.1. Stage 2: Focuses on generating editing pairs for each category by leveraging a diverse set of models and tools in Sec. 2.2. Stage 3: Involves the fine-grained filtering of all pairs generated in Stage 2 to retain only high-quality samples in Sec. 2.3.

2.2 Taxonomy-Guided Generation of Video Edits and Instructions

The OpenVE-3M dataset includes eight major video editing categories: six SA (*Global Style Transfer*, *Background Change*, *Local Change*, *Local Remove*, *Local Add*, *Subtitles Edit*) and two NSA (*Multi-Shot Camera Edit*, *Creative Edit*). SA edits maintain perfect consistency in motion and detail between the original and edited videos. In contrast, NSA edits maintain the primary subject’s consistency but introduce new, creative motion. The construction process for each category is detailed below. The respective flowcharts can be found in **Appendix**.

Global Style. This category involves transforming the global style of a video while preserving the original motion and details. It includes *18 common styles* (e.g., Ghibli, oil painting), *four times of day* (e.g., Morning, Blue Hour), and *three weather conditions* (e.g., Sunny, Rainy, Snowy). For the data generation pipeline, the initial frame of the video is input to GPT-4o with a carefully designed prompt to generate an image editing instruction and corresponding forward/backward prompts for video editing. The initial frame and the image editing instruction are then used with the FLUX-Kontext-dev [14] model to perform stylistic image editing. The edited initial frame, along with the Canny or depth video of the original clip, is then input to the Wan2.1-Fun-V1.1-14B-Control [27] to generate the final stylized video under the structural guidance of the Canny maps.

Background Change. For videos with a clear foreground-background distinction, this task involves *changing the background to various scenes*. Since robust video foreground detection algorithms are not readily available, we adapt an image-based approach. First, an image foreground detection algorithm is applied to the initial frame to generate a foreground mask. We then compute the Intersection over Union (IoU) between this mask and all possible combinations of object

masks (obtained via Grounded SAM2 [23]). If the IoU for a combination exceeds 0.95, that combined mask is considered the definitive foreground mask for the video. The pipeline then proceeds by using GPT-4o to generate image and video-level prompts for background editing. The FLUX-Kontext-dev [14] model edits the background of the initial frame. We create a foreground-only Canny/depth video by masking out the background. Finally, this masked Canny/depth video and the edited initial frame are input to the Wan2.1-Fun-V1.1-14B-Control [27] model to generate the final video with the new background.

Local Change. This includes a range of edits such as *object transformation, style modification, color alteration, and age progression*. The initial frame and its localized object descriptions (from DAM [16]) are input to GPT-4o with a carefully designed prompt. This generates a variety of rich, localized image and video editing prompts. The FLUX-Kontext-dev [14] model then performs the local edit on the initial frame. Finally, the edited frame, along with the original Canny or depth video, is fed into the Wan2.1-Fun-V1.1-14B-Control [27] model to produce the edited video clip.

Local Remove/Add. For local object removal and addition, we devise two distinct data generation pipelines. To create training data for local addition, we first leverage video inpainting models [15, 39] to remove an object from an original video clip based on a given mask. This resulting inpainted video, where the object is absent, serves as the source video for the addition task. We then use GPT-4o to generate a corresponding ‘local add’ instruction from an object description [16], while the original, unedited video acts as the ground-truth target. Conversely, to generate data for local removal, we synthesize a video with a new object. This process begins by using GPT-4o to generate an image-level ‘add’ instruction for the first frame of a video, which is then edited by the FLUX-Kontext-dev [14] model. Subsequently, the Wan2.2-I2V-A14B [27] model generates a full video from this single edited frame, and we employ Grounded SAM2 [23] to segment the newly added object. The masked region containing the object is then pasted onto the original video frames to produce the final edited clip. This synthetic video with the added object then becomes the source video for the removal task. A ‘local remove’ instruction is generated by GPT-4o, and the original video without object serves as the corresponding target video.

Subtitles Edit. This category includes tasks for *adding, removing, and modifying subtitles*, featuring *nine variations (three positions: top, middle, bottom)*. The pipeline uses GPT-4o to generate appropriate video text. A rendering tool then renders subtitles of different styles at various positions. Finally, GPT-4o generates the editing instructions for the add, remove, and modify tasks.

Camera Multi-Shot Edit. This NSA task involves editing a video to switch between *close-up, medium, and wide shots of the same subject*, comprising a total of *six transition types*. We leverage the “Native Multi-Shot” feature of Seedance [10]. GPT-4o generates prompts for a three-shot sequence, which are then used for I2V generation. After generating a multi-shot video with a consistent subject, visual style, and atmosphere, a shot detection model segments the

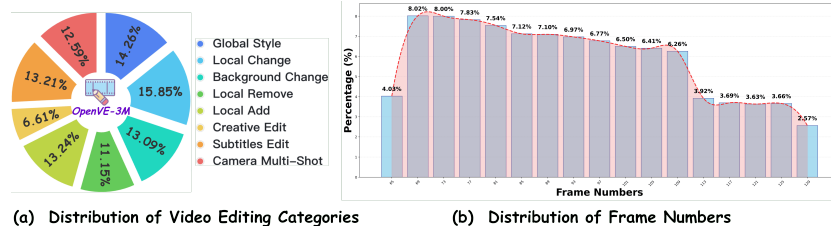


Fig. 4: Category and Frame Count Statistics for OpenVE-3M.

different shots. These segments are then used to create shot-switching edit pairs and corresponding instructions.

Creative Edit. This NSA involves editing an object to follow a creative instruction, where the *subject’s actions may change significantly*. For a given source video, we use GPT-4o to generate multiple creative I2V instructions based on its initial frame. The Seedance [10] model then generates high-quality videos for each instruction. Finally, any two of these generated videos can form an editing pair, with GPT-4o generating a creative editing instruction to describe the transformation between them.

2.3 Data Filtering and Curation

To ensure video quality, we first filter the synthesized videos to mitigate distribution and temporal drifts that may arise from discrepancies between I2I and I2V generation. Specifically, we employ Temporal Flickering (TF) to quantify inter-frame flicker via mean absolute difference and CLIP-F to measure temporal consistency through average CLIP [22] similarity. Videos exhibiting high flicker or low consistency are subsequently discarded. Secondly, we designed a meticulous data filtering pipeline for all synthesized data pairs (excluding Subtitles Edit) based on VLMs. This process is based on three main evaluation metrics, each rated on a scale of 1 to 5: 1) *Instruction Compliance*, 2) *Consistency & Detail Fidelity*, and 3) *Visual Quality & Stability*. Crucially, the score for Instruction Compliance serves as an upper bound for the other two metrics. We prioritize instruction compliance as the primary evaluation criterion because many generated videos, despite high visual quality, may fail to follow the edit instruction. Finally, we input the editing instruction, original video, and edited video into various VLMs for automated scoring. To benchmark the VLMs, we first manually annotated a set of 300 video pairs, defining pairs with an average score > 3 as positive samples and those ≤ 3 as negative samples. The resulting accuracies were: Qwen3-VL-30B-A3B [3] (61%), InternVL3.5-38B [28] (66%), Seed1.6-VL [25] (70%), and Gemini2.5-Pro [8] (69%). Due to API rate limits (TPM) for Seed1.6-VL and Gemini2.5-Pro, we ultimately selected the Intern3.5-VL-38B [28] to eval the entire dataset, retaining pairs with score ≥ 3 .

Our final OpenVE-3M dataset contains 3 million video editing pairs across the eight categories. As shown in Tab. 1, OpenVE-3M surpasses existing IVE

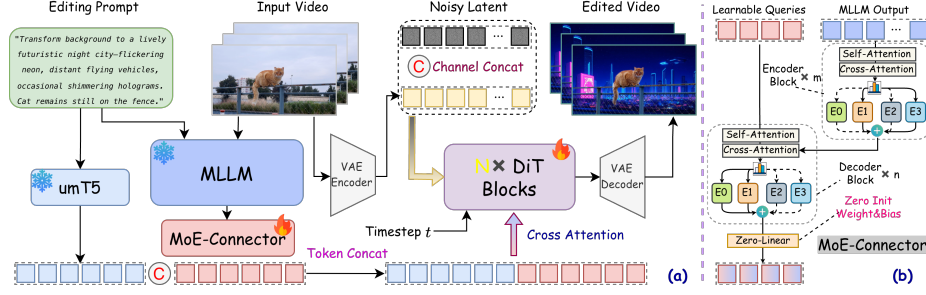


Fig. 5: Overall of the OpenVE-Edit. (a) Architecture of OpenVE-Edit. (b) The detailed structure of the MoE-Connector module.

datasets in terms of task diversity, data scale, resolution, frame count, and prompt detail. Specifically, our dataset features more categories and a higher average instruction length than its counterparts. Its video frame count and resolution are also among the highest quality available. Furthermore, as illustrated in Fig. 4, the distribution across different editing types is well-balanced, making OpenVE-3M a comprehensive, high-quality, and large-scale IVE dataset.

3 OpenVE-Edit: An Effective Instruction-Guided Video Editing Method

In Fig. 5, OpenVE-Edit, consists of three main modules: a MLLM, a MoE-Connector, and a DiT. The input editing instruction and video are jointly fed into the MLLM to capture the semantic relationships between the instruction and the visual content. Subsequently, a task-aware MoE-Connector processes the hidden features from the MLLM, decoupling them through multiple expert networks. These processed features are then concatenated along the token dimension with the instruction features encoded by umT5. Concurrently, the latent features of the original video, derived from a VAE, are concatenated with noise along channel dimension. This composite latent representation subsequently interacts with the combined semantic editing features through the Cross-Attention mechanism in the DiT model.

3.1 Editing Knowledge Injection via MLLM

Baseline video generation models, such as Wan [27], rely solely on umT5-encoded text features, which interact with noisy latent features through cross-attention. However, using only umT5 features provides a literal representation of the editing instruction and fails to capture higher-level semantic and spatial relationships between the instruction and the visual content. Therefore, we jointly input the original video and the editing instruction into an MLLM. This enables the model to capture these higher-level semantic and spatial relationships. Furthermore,

to emphasize edit-relevant semantics, we discard token embeddings associated with prefixes, such as system prompts. This filtering process ensures that only tokens corresponding directly to the editing information are retained, focusing the subsequent processing on the specific editing task.

3.2 Decoupling Task Cues from Expert Gating

Addressing the diverse requirements of editing tasks with a single model presents a challenge of task heterogeneity, which can lead to parameter inefficiency. Shared parameters may internalize conflicting representations from different tasks, resulting in suboptimal specialization and an increased parameter count. Therefore, we design a task-aware MoE-Connector module to handle different editing types for both images and videos. Since videos of varying frame counts and resolutions produce different numbers of tokens from the MLLM, we employ learnable queries to extract key information from the hidden states for different editing tasks. This is achieved by activating specific sparse expert sub-networks, allowing for precise capacity allocation for each task. Given input $\mathbf{X}_{in} \in \mathbb{R}^{b \times s \times d}$ where b is batch size, s is sequence length, and d is the MLLM hidden dimension, the forward process of the overall model is:

$$\mathbf{Y} = \mathbf{W}_o \left(\text{MoE-Dec}(\mathbf{Q}_l, \text{MoE-Enc}(\mathbf{F}_i(\mathbf{X}_{in}))) \right), \quad (1)$$

where $\mathbf{Q}_l \in \mathbb{R}^{b \times L_q \times D_h}$ is the learnable decoder query with L_q being the learnable query length and D_h being the hidden dimension, $\mathbf{W}_o$ is the output linear transformation matrix, $\mathbf{F}_i$ is the Feed-Forward Network (FFN) for dimension mapping, MoE-Enc and MoE-Dec are the MoE encoder and decoder modules respectively, each containing a self-attention layer, a cross-attention layer, and an MoE-FFN. For input $\mathbf{x} \in \mathbb{R}^{D_h}$, the MoE-FFN computation process is:

$$\mathbf{y} = \sum_{i \in S} w_i \cdot \left(\mathbf{W}_{i,2} \cdot \text{GELU}(\mathbf{W}_{i,1}\mathbf{x} + \mathbf{b}_{i,1}) + \mathbf{b}_{i,2} \right), \quad (2)$$

where $S = \text{topk}(\text{softmax}(\mathbf{W}_g\mathbf{x}), k)$ is the set of top- k selected expert indices, $\mathbf{W}_g$ is the gate weight matrix, $\mathbf{W}_{i,j}$ and $\mathbf{b}_{i,j}$ are the weights and biases of the i -th expert network, w_i is the normalized weight for expert i .

3.3 Accelerating Convergence in A Unified Model

Unlike large pre-trained models, our MoE-Connector is randomly initialized, which risks introducing noise and destabilizing training. To mitigate this, we draw inspiration from ControlNet [37] by *zero-initializing* the weights of the connector’s final MLP layer $\mathbf{W}_o$. We then concatenate its output with the original umT5 instruction features. This design effectively renders the module "invisible" at the start of training, as its zero-output preserves the integrity of the umT5 features. As training progresses, the module gradually learns to contribute useful information via gradient descent, ensuring stable and efficient convergence.

4 OpenVE-Bench: A Robust Instruction-Guided Video Editing Benchmark

4.1 Benchmark Construction

Our OpenVE-Bench is constructed with two primary categories: SA and NSA edits. These are further divided into eight fine-grained subcategories, totaling 431 IVE pairs, with each subcategory containing over 50 video clips on average. For *Global Style*, we carefully select 58 video clips and design instructions covering 18 distinct styles, 3 different times of day, and 4 weather conditions. For *Background Change*, we manually select 59 clips suitable for background changes and design diverse instructions with styles ranging from harmonious to highly stylized. For *Local Change*, we select 65 clips featuring prominent subjects such as humans, animals, or vehicles, with instructions for object transformation, style modification, and age manipulation. For *Local Add*, we select 67 clips of varying difficulty and design instructions for adding objects of different sizes, from small items like a kite to large ones like a car. For *Local Remove*, we select 59 videos featuring subjects of varying sizes and types and design corresponding removal instructions. For *Subtitles Edit*, we select 50 clips from diverse scenes and styles, with instructions to add, delete, or replace text at top, middle, or bottom positions using various styles. For *Camera Multi-Shot Edit*, we select 43 clips with prominent subjects and design instructions for transitioning between long, medium, and close-up shots. For *Creative Edit*, we select 30 distinctive video clips and use GPT-4o to generate diverse and imaginative instructions.

4.2 Evaluation Metrics

Following established criteria [19, 32, 34], we evaluate IVE on a 1-5 scale for *Instruction Compliance*, *Consistency & Fidelity*, and *Visual Quality & Stability*. Crucially, the *Instruction Compliance* score upper-bounds the other two, prioritizing adherence to the edit instruction. We also assess video quality by quantifying temporal flicker with *TF* (Temporal Flickering (mean absolute difference)) and consistency with *CLIP-F* (average inter-frame CLIP similarity).

5 Experiments

5.1 Implementation Details

We use Wan2.2-TI2V-5B as the video generation base model. The Qwen2.5VL-3B [3] processes the input video and editing instructions. For the MoE-Connector, both the encoder and decoder blocks have 2 layers. Each layer contains 6 experts, with 2 experts activated per forward pass. The sequence length of the learnable queries is 512. We employ a two-stage training strategy. First, we train the model for one epoch at a 480P resolution with a global batch size of 512 and a learning rate of $1e-5$. Subsequently, we fine-tune the model for an additional epoch at a 720P resolution with a reduced learning rate of $1e-6$.

Table 2: Quantitative Comparison on **OpenVE-Bench** with Seed1.6-VL [25]. White, yellow, gray, and blue backgrounds indicate Closed-source, Open-source and Ours, respectively. #Params. and #Reso. refers to Parameters and Resolutions, respectively. * Indicates that Extra Image Editing Model is required for video editing.

Methods	Model Details		OpenVE-Bench results on Seed1.6-VL									Gemini	Video Quality	
	#Params.	#Reso.	Overall	GS	BC	LC	LR	LA	SE	CE	CME	Overall	TF	CLIP-F
Runway [1]	-	1280×720	3.50	3.47	2.84	3.88	3.88	2.79	3.50	3.23	4.48	3.65	0.9851	0.9659
VACE [12]	14B	1280×720	1.17	1.41	1.16	1.43	1.00	1.05	1.02	1.13	1.16	1.57	0.9852	0.9685
OmniVideo [24]	1.3B	640×352	1.02	1.02	1.00	1.00	1.00	1.00	1.16	1.00	1.00	1.31	0.9916	0.9785
InsViE [30]	2B	720×480	1.40	2.25	1.23	1.60	1.00	1.23	1.22	1.68	1.02	1.53	0.9768	0.9739
AnyV2V [13]*	3.7B	512×512	1.41	1.84	1.31	1.94	1.09	1.27	1.02	1.57	1.11	1.51	0.9614	0.9589
Señorita [40]*	5B	800×448	1.90	2.29	1.47	2.69	1.93	1.64	2.49	1.91	1.02	1.93	0.9802	0.9658
Lucy-Edit [26]	5B	1280×704	1.95	2.17	2.20	3.30	1.03	2.37	1.06	2.35	1.14	2.15	0.9836	0.9692
ICVE [17]	13B	384×240	2.25	2.35	1.86	2.91	2.68	2.27	2.04	1.94	1.38	2.07	0.9877	0.9661
DITTO [2]	14B	832×480	2.06	3.70	2.23	2.28	1.00	2.08	1.01	2.61	1.51	1.98	0.9848	0.9635
VideoCoF [33]	14B	1280×704	2.24	1.07	1.33	3.52	3.26	2.67	2.57	1.64	1.05	2.24	0.9768	0.9651
OpenVE-Edit	5B	1280×704	2.41	3.11	2.72	3.19	1.42	2.41	2.56	2.01	1.24	2.49	0.9889	0.9698

5.2 Comparison With State-of-The-Art Models

We compare our model with existing state-of-the-art open-source models, including VACE [12], OmniVideo [24], InsViE [30], AnyV2V [13], Señorita [40], ICVE [17], Lucy-Edit [26], and DITTO [2], as well as the closed-source model Runway Aleph [1]. During our reproduction of the open-source models on a single GPU with 80GB of VRAM, we observe several limitations. OmniVideo [24] can only generate videos at a 640x352 resolution with 17 frames; other settings result in video artifacts. The ICVE [17] model can generate a maximum of 41 frames at a 480x768 resolution; generating more frames leads to OOM. Therefore, we use a resolution of 384x240 for ICVE to ensure all frames can be edited. Unlike end-to-end approaches, AnyV2V [13] and Señorita [40] rely on external I2I models to edit the first frame before guiding the final video generation. AnyV2V employs [5] while Señorita-2M uses [14] for this first-frame editing. The other models are evaluated using their respective official training resolutions and input frame counts. Additionally, due to the cost constraints of Runway Aleph, we only select 30 samples from each evaluation category for benchmarking.

Quantitative Comparison. Tab. 2 has the evaluation results of all instruction-guided video editing models on OpenVE-Bench. The closed-source model, Runway Aleph, achieves the best performance on both the Seed-1.6VL [25] and Gemini 2.5 Pro [8] evaluators, significantly outperforming existing open-source models. The open-source models VACE [12], OmniVideo [24], and InsViE [30] exhibit limited performance due to constraints in their model size or training data. With 5B parameters, Lucy-Edit [26] achieves moderate performance. ICVE [17], with 13B parameters, obtains respectable results, but its high-resolution editing capability is limited to a smaller number of frames. Ditto [2] scores highly on the global style metric, as its training dataset primarily consists of this edit type. Our model, OpenVE-Edit, with only 5B parameters, achieves an overall score of **2.41**, surpassing all existing open-source models with a smaller parameters. While OmniVideo [24] excels on temporal metrics (TF and CLIP-F), it entirely



Fig. 6: Qualitative comparison with SoTA methods with Background Change(left) and Local Change(right) examples.

disregards the edit instruction. Our method, however, achieves both strong temporal quality and superior instruction-following results.

Qualitative Results. Fig. 6 presents a qualitative comparison between our method and existing open-source SOTA methods. We select the top three performing open-source models for this comparison. In the “Background Change” example on the left, although Lucy-Edit [26] successfully changes the background, it fails to maintain the consistency of the dog from the original video. ICVE [17] incorrectly erases the woman and alters the man’s appearance. Ditto [2] mistakenly treats the painting on the wall as a foreground object and darkens the dog’s fur. In contrast, our method alters the background according to the instruction while preserving the consistency of all foreground subjects. In the “Local Change” example on the right, Lucy-Edit [26] incorrectly edits the clothing of all three individuals. ICVE [17] erroneously modifies the two people on the left and also changes their identities. Ditto [2] not only edits the wrong subject but also incorrectly alters the background. Our method, however, precisely follows the instruction by changing only the specified woman’s clothing while maintaining the consistency of the other individuals and the background.

5.3 Extra Ablation and Explanatory Analysis

Ablation on Control Signals for Data Construction. We use depth maps and Canny edge maps derived from the original video as separate control conditions using Wan2.1-Fun-V1.1-14B-Control [27]. In Fig. 7, when there is a significant difference between the first and last frames of a video, using depth control results in poor detail generation. In contrast, Canny edge control better preserves the details of the original video. Furthermore, if the depth maps are consistent

Table 3: Ablations of the **Model Design** and the **Training Dataset**. **Overall** indicates the average value of all metrics in OpenVE-Bench.

Index	Structure		Training Dataset		Overall
	MLLM	MoE-Connector	Senorita-2M	OpenVE-3M	
①				✓	2.12
②	✓			✓	2.31
③	✓	✓	✓		1.54
④	✓	✓		✓	2.41

across consecutive frames, the background in the generated video often remains static. Therefore, we choose Canny maps as the control condition for our pipeline. **Acceleration of Data Construction.** In the data construction phase, three tasks require inference with the Wan2.1-Fun-V1.1-14B-Control model. The average inference time for a single 81-frame, 720P video on one GPU is *50 minutes*. This speed is too slow for large-scale data generation. Consequently, we employ two strategies to accelerate this process. First, we replace FlashAttention-2 [9] with the faster SageAttention-2 [36]. Second, we reduce the number of inference steps from *50 to 10*. These optimizations reduce the average generation time to *6 minutes*, an $\times 8 \uparrow$ speedup. Manual inspection of a large number of samples confirms that the quality of the accelerated data generation meets our requirements and shows no degradation. Additionally, for image-to-video (I2V) generation, we replace the original Wan2.2-I2V-A14B model with its *4-step distilled* version, Wan2.2-I2V-A14B-NFE4-V1, which also significantly improves efficiency. **Ablations on Model Design.** Compared to using only umT5 instructions, jointly inputting the video and instruction into the MLLM enables the model to better comprehend the instruction and execute the corresponding edit. In Fig. 8, ① using only T5 features can lead to a mismatch between the editing result and the instruction, whereas features from the MLLM ② mitigate this issue. This is corroborated by the quantitative metrics, with the score improving

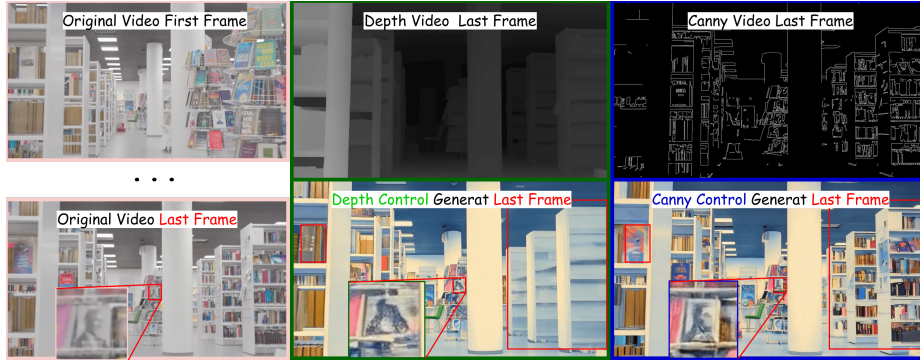


Fig. 7: Comparison with different control videos for Stage2.

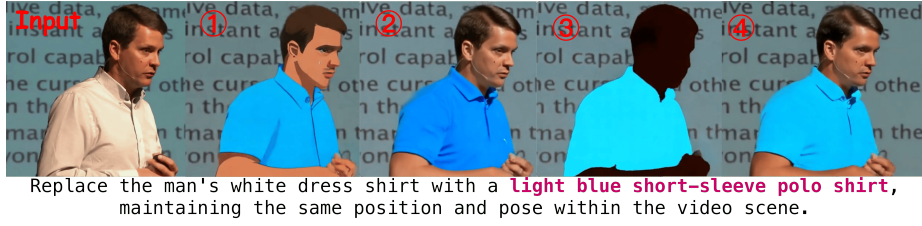


Fig. 8: Qualitative ablations of the **Model Design** and the **Training Dataset**.

from *2.12* to *2.31* (+0.19 $\uparrow$) in Tab. 3. Furthermore, compared to a simple MLP connector with shared parameters, the MoE-Connector is more effective at processing features for different edit types by routing them to specialized experts. This is reflected in the average score, which increases from *2.31* to *2.41* (+0.1 $\uparrow$), demonstrating the effectiveness of our approach in handling diverse and complex editing tasks. ④ enhances ② by incorporating an MoE-Connector, resulting in a better representation of semantic details.

Ablations on Training Datasets. We ablate the training data using Senorita-2M [40] as a baseline. The resulting average score of *1.54* is significantly lower than the *2.41* achieved when training on our OpenVE-3M dataset in Tab. 3. This indicates potential quality issues in the Senorita-2M dataset, a conclusion further supported by the high proportion of samples with a score of 1, as illustrated in Fig. 2. Also, in Fig. 8, the model in ④ is outperformed by ③, a performance drop caused by training on the low-level data abundant in Senorita-2M.

Ablations on Video Removal Methods. While MiniMax-Remover [39] offers smoother temporal consistency and less flicker than DiffuEraser [15], it suffers from twice the inference time and poorer background reconstruction in Fig. 9. Consequently, we leverage a mix of both models to generate our Remove and Add data, balancing their respective strengths and weaknesses.

Ablations on Human alignment with VLMs. The 300 pairs were stratified across all 8 edit categories and cover outputs from multiple editing models/quality levels. Each pair was independently scored by 3 annotators using the same three 1–5 criteria. The final human score is the average of the three annotations. The inter-rater agreement is strong, with two-way random-effects ICC(2,k)=0.948 over the annotators’ overall scores. We computed PLCC/ SROCC between VLM overall scores and mean human overall scores in Tab. 4. Seed1.6-VL shows the strongest agreement with humans (PLCC/SROCC 0.833/0.842),



Fig. 9: Qualitative ablations of the **Video Removal Methods**.

Table 4: Ablations of the Human alignment with VLMs on OpenVE-Bench.

Judge	Seed1.6-VL	Gemini2.5-Pro	Intern3.5-VL-38B	Qwen3-VL-30B-A3B
PLCC	0.833	0.731	0.670	0.583
SROCC	0.842	0.756	0.643	0.555

and Gemini2.5-Pro is also clearly better than InternVL3.5/ Qwen3-VL. This supports our benchmark design: *stronger VLMs are used for final model ranking*, while *open-source VLMs are only auxiliary references or coarse filters*. To avoid overstatement, we will revise the claim to “*reasonably aligned with human judgment*”.

6 Conclusion

We introduce OpenVE-3M, a large-scale, high-quality dataset for instruction-following video editing. It comprises two main categories, spatially-aligned and non-spatially-aligned, encompassing a total of eight sub-categories. The detailed data construction and filtering pipelines are open-sourced to facilitate further research in the community. We also propose the OpenVE-Edit model, trained on OpenVE-3M. With only 5B parameters, our model outperforms a 14B parameter baseline. For standardized evaluation, we introduce OpenVE-Bench, which demonstrates high correlation with human judgment. On this benchmark, our method achieves SOTA performance among open-source models.

Limitations and Future Work. While ours OpenVE-3M dataset is diverse, it omits certain categories like reference-based style transfer. Also, computational constraints limited our experiments to smaller-scale model architectures. We plan to explore architectures that better unify spatially-aligned and non-spatially-aligned edits, potentially by adapting computationally-demanding image editing paradigms like noise concatenation along token dimensions, which remains under-explored in video. We will also investigate unified models, focusing on the bidirectional synergy where model understanding, generation and editing capabilities mutually enhance each other.

Acknowledgements

This work was supported by Jianbing Lingyan Foundation of Zhejiang Province, P.R. China (Grant No. 2023C01022) and Major Project of Science and Technology of Yunnan Province, China under Grant 202402AD080001.

References

1. Creating anything. <https://runwayml.com/> (2025) **11**
2. Bai, Q., Wang, Q., Ouyang, H., Yu, Y., Wang, H., Wang, W., Cheng, K.L., Ma, S., Zeng, Y., Liu, Z., Xu, Y., Shen, Y., Chen, Q.: Scaling instruction-based video editing with a high-quality synthetic dataset (2025), <https://arxiv.org/abs/2510.15742> **2, 11, 12**
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025) **3, 4, 7, 10**
4. Bradski, G.: The opencv library. Dr. Dobb’s Journal: Software Tools for the Professional Programmer **25**(11), 120–123 (2000) **3, 4**
5. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 18392–18402 (2023) **11**
6. Chen, S., Guo, H., Zhu, S., Zhang, F., Huang, Z., Feng, J., Kang, B.: Video depth anything: Consistent depth estimation for super-long videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22831–22840 (2025) **3, 4**
7. Cheng, J., Xiao, T., He, T.: Consistent video-to-video transfer using synthetic dataset. arXiv preprint arXiv:2311.00213 (2023) **2**
8. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025) **1, 2, 3, 7, 11**
9. Dao, T.: Flashattention-2: Faster attention with better parallelism and work partitioning. arXiv preprint arXiv:2307.08691 (2023) **13**
10. Gao, Y., Guo, H., Hoang, T., Huang, W., Jiang, L., Kong, F., Li, H., Li, J., Li, L., Li, X., et al.: Seedance 1.0: Exploring the boundaries of video generation models. arXiv preprint arXiv:2506.09113 (2025) **6, 7**
11. Hu, J., Zhong, T., Wang, X., Jiang, B., Tian, X., Yang, F., Wan, P., Zhang, D.: Vivid-10m: A dataset and baseline for versatile and interactive video local editing. arXiv preprint arXiv:2411.15260 (2024) **2**
12. Jiang, Z., Han, Z., Mao, C., Zhang, J., Pan, Y., Liu, Y.: Vace: All-in-one video creation and editing. arXiv preprint arXiv:2503.07598 (2025) **11**
13. Ku, M., Wei, C., Ren, W., Yang, H., Chen, W.: Anyv2v: A tuning-free framework for any video-to-video editing tasks. arXiv preprint arXiv:2403.14468 (2024) **11**
14. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., et al.: Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. arXiv preprint arXiv:2506.15742 (2025) **1, 3, 5, 6, 11**
15. Li, X., Xue, H., Ren, P., Bo, L.: DiffuEraser: A diffusion model for video inpainting. arXiv preprint arXiv:2501.10018 (2025) **6, 14**
16. Lian, L., Ding, Y., Ge, Y., Liu, S., Mao, H., Li, B., Pavone, M., Liu, M.Y., Darrell, T., Yala, A., et al.: Describe anything: Detailed localized image and video captioning. arXiv preprint arXiv:2504.16072 (2025) **3, 4, 6**
17. Liao, X., Zeng, X., Song, Z., Fu, Z., Yu, G., Lin, G.: In-context learning with unpaired clips for instruction-based video editing (2025), <https://arxiv.org/abs/2510.14648> **11, 12**

18. Lin, B., Ge, Y., Cheng, X., Li, Z., Zhu, B., Wang, S., He, X., Ye, Y., Yuan, S., Chen, L., et al.: Open-sora plan: Open-source large video generation model. arXiv preprint arXiv:2412.00131 (2024) [4](#)
19. Liu, S., Han, Y., Xing, P., Yin, F., Wang, R., Cheng, W., Liao, J., Wang, Y., Fu, H., Han, C., et al.: Step1x-edit: A practical framework for general image editing. arXiv preprint arXiv:2504.17761 (2025) [10](#)
20. Ma, J., Zhu, X., Pan, Z., Peng, Q., Guo, X., Chen, C., Lu, H.: X2edit: Revisiting arbitrary-instruction image editing through self-constructed data and task-aware representation learning. arXiv preprint arXiv:2508.07607 (2025) [1](#)
21. Nan, K., Xie, R., Zhou, P., Fan, T., Yang, Z., Chen, Z., Li, X., Yang, J., Tai, Y.: Openvid-1m: A large-scale high-quality dataset for text-to-video generation. arXiv preprint arXiv:2407.02371 (2024) [4](#)
22. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021) [7](#)
23. Ravi, N., Gabeur, V., Hu, Y.T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L., Mintun, E., Pan, J., Alwala, K.V., Carion, N., Wu, C.Y., Girshick, R., Dollár, P., Feichtenhofer, C.: Sam 2: Segment anything in images and videos (2024), <https://arxiv.org/abs/2408.00714> [3](#), [4](#), [6](#)
24. Tan, Z., Yang, H., Qin, L., Gong, J., Yang, M., Li, H.: Omni-video: Democratizing unified video understanding and generation. arXiv preprint arXiv:2507.06119 (2025) [11](#)
25. Team, B.S.: Seed1.5-vl technical report. arXiv preprint arXiv:2505.07062 (2025) [3](#), [7](#), [11](#)
26. Team, D.: Lucy edit: Open-weight text-guided video editing (2025), https://d2drjpuinn461b.cloudfront.net/Lucy_Edit__High_Fidelity_Text_Guided_Video_Editing.pdf [11](#), [12](#)
27. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., Wang, J., Zhang, J., Zhou, J., Wang, J., Chen, J., Zhu, K., Zhao, K., Yan, K., Huang, L., Feng, M., Zhang, N., Li, P., Wu, P., Chu, R., Feng, R., Zhang, S., Sun, S., Fang, T., Wang, T., Gui, T., Weng, T., Shen, T., Lin, W., Wang, W., Wang, W., Zhou, W., Wang, W., Shen, W., Yu, W., Shi, X., Huang, X., Xu, X., Kou, Y., Lv, Y., Li, Y., Liu, Y., Wang, Y., Zhang, Y., Huang, Y., Li, Y., Wu, Y., Liu, Y., Pan, Y., Zheng, Y., Hong, Y., Shi, Y., Feng, Y., Jiang, Z., Han, Z., Wu, Z.F., Liu, Z.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) [3](#), [5](#), [6](#), [8](#), [12](#)
28. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025) [3](#), [7](#)
29. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-image technical report. arXiv preprint arXiv:2508.02324 (2025) [1](#)
30. Wu, Y., Chen, L., Li, R., Wang, S., Xie, C., Zhang, L.: Insvie-1m: Effective instruction-based video editing with elaborate dataset construction. arXiv preprint arXiv:2503.20287 (2025) [2](#), [11](#)
31. Xue, Z., Zhang, J., Hu, T., He, H., Chen, Y., Cai, Y., Wang, Y., Wang, C., Liu, Y., Li, X., et al.: Ultravideo: High-quality uhd video dataset with comprehensive captions. arXiv preprint arXiv:2506.13691 (2025) [4](#)
32. Yang, S., Hui, M., Zhao, B., Zhou, Y., Ruiz, N., Xie, C.: Complex-edit: Cot-like instruction generation for complexity-controllable image editing benchmark. arXiv preprint arXiv:2504.13143 (2025) [10](#)

33. Yang, X., Xie, J., Yang, Y., Ma, Y., Huang, Y., Xu, M., Wu, Q.: Videocof: Unified video editing with temporal reasoner. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 37940–37949 (2026) [11](#)
34. Ye, Y., He, X., Li, Z., Lin, B., Yuan, S., Yan, Z., Hou, B., Yuan, L.: Imgedit: A unified image editing dataset and benchmark. arXiv preprint arXiv:2505.20275 (2025) [10](#)
35. Yu, Q., Chow, W., Yue, Z., Pan, K., Wu, Y., Wan, X., Li, J., Tang, S., Zhang, H., Zhuang, Y.: Anyedit: Mastering unified high-quality image editing for any idea. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 26125–26135 (2025) [1](#)
36. Zhang, J., Huang, H., Zhang, P., Wei, J., Zhu, J., Chen, J.: Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. arXiv preprint arXiv:2411.10958 (2024) [13](#)
37. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023) [9](#)
38. Zhang, Z., Dai, Z., Qin, L., Wang, W.: Effived: Efficient video editing via text-instruction diffusion models. arXiv preprint arXiv:2403.11568 (2024) [2](#)
39. Zi, B., Peng, W., Qi, X., Wang, J., Zhao, S., Xiao, R., Wong, K.F.: Minimax-remover: Taming bad noise helps video object removal. arXiv preprint arXiv:2505.24873 (2025) [6](#), [14](#)
40. Zi, B., Ruan, P., Chen, M., Qi, X., Hao, S., Zhao, S., Huang, Y., Liang, B., Xiao, R., Wong, K.F.: Se\~norita-2m: A high-quality instruction-based dataset for general video editing by video specialists. arXiv preprint arXiv:2502.06734 (2025) [2](#), [11](#), [14](#)