EDITVERSE: UNIFYING IMAGE AND VIDEO EDITING AND GENERATION WITH IN-CONTEXT LEARNING

Anonymous authors
Paper under double-blind review

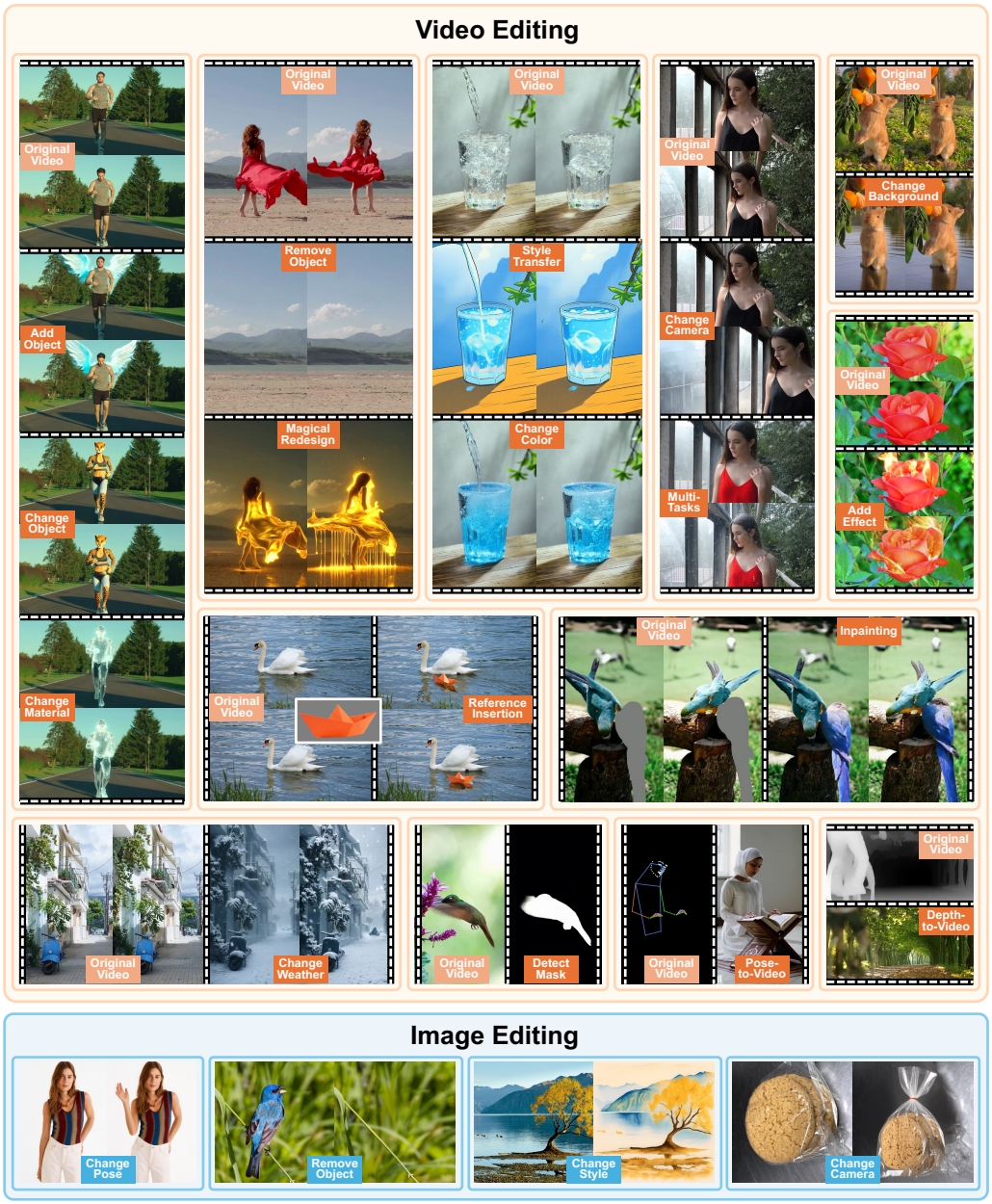


Figure 1: The strong video editing performance of EditVerse **emerges from a unified framework trained on a diverse set of mixed image and video data**. This teaser shows a representative subset of the supported image and video editing tasks, demonstrating the versatility and robustness of our approach. Original editing instructions and more results can be found in the Appendix.

ABSTRACT

Recent advances in foundation models highlight a clear trend toward unification and scaling, showing emergent capabilities across diverse domains. While image generation and editing have rapidly transitioned from task-specific to unified frameworks, video generation and editing remain fragmented due to architectural limitations and data scarcity. In this work, we introduce EditVerse, a unified framework for image and video generation and editing within a single model. By representing all modalities, *i.e.*, text, image, and video, as a unified token sequence, EditVerse leverages self-attention to achieve robust in-context learning, natural cross-modal knowledge transfer, and flexible handling of inputs and outputs with arbitrary resolutions and durations. To address the lack of video editing training data, we design a scalable data pipeline that curates 232K video editing samples and combines them with large-scale image and video datasets for joint training. Furthermore, we present EditVerseBench, the first benchmark for instruction-based video editing covering diverse tasks and resolutions. Extensive experiments and user studies demonstrate that EditVerse achieves state-of-the-art performance, surpassing existing open-source and commercial models, while exhibiting emergent editing and generation abilities across modalities.

1 INTRODUCTION

Recent advancements of foundation models in computer vision and large language models highlight a clear trend toward unification and scaling (Achiam et al., 2023; Zhou et al., 2024; Deng et al., 2025), showing that joint training on diverse datasets can unlock emergent intelligence. Specifically in image generation and editing, there is also a shift from domain-specific models (Zhang et al., 2023b; Ju et al., 2023b; Li et al., 2024) toward universal models (Labs et al., 2025; Chen et al., 2025c) that unify diverse generation and editing tasks under a generalized and scalable framework.

However, unlike the image domain, the exploration of unified video generation and editing remains limited (Jiang et al., 2025; Ye et al., 2025b). This stems from two primary challenges: **(1) Architectural Limitations**: Existing video generation models, mostly based on cross-attention (Polyak et al., 2025; Wan et al., 2025) or MMDiT (Yang et al., 2024c; Kong et al., 2024) architecture, are typically designed for specific tasks such as text-to-video generation. Adapting them to support various video generation and editing tasks introduces substantial design and scaling challenges. For example, VACE (Jiang et al., 2025) uses an additional branch that accepts unedited videos and masks as input, transforming a text-to-video model into a video inpainting model. However, it relies on masks to localize the editing regions and requires task-specific input configurations, making it less practical for real-world use. To unlock emergent abilities with in-context learning, a fully unified framework must be able to process diverse input modalities (e.g., text, image, video) and types (e.g., duration, resolution) with a consistent and flexible representation. **(2) Data Scarcity and Diversity**: Unlike the vast and varied datasets readily available for image editing (Yu et al., 2024; Ye et al., 2025a; Chen et al., 2025b), high-quality and diverse video editing datasets are significantly scarce.

To address this challenge, we propose **EditVerse**, a unified framework that enables image and video editing and generation within a single model, leveraging full self-attention to enable robust in-context learning and effective knowledge transfer between images and videos. Our design considers two aspects: **(1) In-Context Learning**: We represent all modalities (text, image, and video) as a unified one-dimensional token sequence, which is then concatenated and fed into the model as a long sequence. This design enables the use of full self-attention with strong in-context learning capabilities (Ju et al., 2025) to jointly model and align different modalities. As a result, EditVerse achieves enhanced text comprehension, improved image and video editing quality, and most importantly, natural cross-modal knowledge transfer between images and videos, which effectively alleviates the limitations caused by the scarcity of video editing data. **(2) Flexibility**: We use an interleaved design for text, image, and video, inspired by the native generation architecture of multimodal large language models (MLLM), which are well-suited for supporting diverse tasks and interactive generation. This design enables the model to process image and video inputs and outputs with arbitrary resolution, temporal duration, and sequential position, thereby providing enhanced flexibility. To further distinguish positional and modal information, we introduce a four-dimensional Rotary Positional Embedding (RoPE) that incorporates sequential, temporal, height, and width dimensions.

While careful model design is crucial, simply training it on image editing data is insufficient to enable the model to perform various video editing tasks. Based on the observation that open-source instruction-based video datasets (Zi et al., 2025) are inadequate in both volume and quality, we devise a data pipeline that first generates video editing samples with task-specific models, then filters high-quality samples from the generated samples. For our unified training, we mix such curated video editing data ($232K$) with $56K$ samples filtered from Señorita-2M as well as $2M$ image generation samples, $6M$ image editing samples, and $4M$ video generation samples.

At last, due to the absence of instruction-based video editing benchmarks encompassing diverse tasks and mixed resolutions, we introduce **EditVerseBench** to enable a more comprehensive evaluation. It contains 100 videos, evenly divided between 50 horizontal and 50 vertical formats, with each video paired with two editing prompts for different editing tasks. Each data instance includes an editing instruction, a source prompt, and a target prompt, spanning 20 distinct video editing categories. Comprehensive evaluations (both automated and user studies) demonstrate that EditVerse achieves state-of-the-art performance compared to existing open-source methods as well as commercial models. Moreover, experiment results show the model's capacity for knowledge transfer from image to video domain and reveal emergent abilities arising from our proposed design.

2 Related Work

Instruction-based Image and Video Editing Datasets. Recent years have seen a surge in large-scale, open-source datasets for instruction-based image editing. Early approaches relied on low-success-rate model-generated annotations (*e.g.*, InstructPix2Pix (Brooks et al., 2023), HQ-Edit (Hui et al., 2024)) or small-scale manual labeling (*e.g.*, MagicBrush (Zhang et al., 2023a)). Modern pipelines leverage task-specific models to generate high-quality data at scale (*e.g.*, UltraEdit (Zhao et al., 2024), OmniEdit (Wei et al., 2024), AnyEdit (Yu et al., 2024), SEED-Data-Edit (Ge et al., 2024), EditWorld (Yang et al., 2024b)), boosting model performance (OpenAI, 2024; Labs et al., 2025) and contributing to further datasets (*e.g.*, ShareGPT-4o-Image (Chen et al., 2025b)).

However, video editing datasets progress at a slower pace. InsV2V (Cheng et al., 2023) uses Prompt-to-Prompt (Hertz et al., 2022) and a large language model (LLM) to create its video editing datasets, but suffers from low-quality outputs. Although VIVID-10M (Hu et al., 2024) provides a collection of videos with corresponding prompt and mask annotations, it lacks paired ground-truth edited videos. Señorita-2M (Zi et al., 2025) uses task-specific diffusion models but remains limited in quality and diversity. In conclusion, instruction-based video editing datasets are less mature, highlighting the need for architectural innovation to transfer editing abilities from image to video.

Image and Video Editing. Diffusion models have driven rapid advances in image and video editing. Early work explored training-free techniques via attention or latent manipulation (Hertz et al., 2022; Cao et al., 2023; Ju et al., 2023a; Wang et al., 2023a; Qi et al., 2023; Liu et al., 2024b; Yoon et al., 2024), but such techniques frequently yield unsatisfactory results characterized by a lack of precise control and low quality. The field has thus shifted to training-based, data-driven methods. For image editing, models like InstructPix2Pix (Brooks et al., 2023) concatenate unedited and noisy latents for fine-tuning, with later work (Ju et al., 2025; Xiao et al., 2025; Cai et al., 2025a; Zhang et al., 2025b) showing sequential concatenation improves in-context learning, aligning with multimodal LLM architectures (*e.g.*, BAGEL (Deng et al., 2025), Transfusion (Zhou et al., 2024)).

While similar techniques can be employed for video editing, investigations into instruction-based video editing are relatively rare. EVE (Singer et al., 2024a) trains adapters on top of frozen text-to-image models to enable video editing ability. InsV2V (Cheng et al., 2023) extends InstructPix2Pix (Brooks et al., 2023) to a video version. GenProp (Liu et al., 2025a) propagates the edits in the given first frame to the following frames. Recent work UNIC (Ye et al., 2025b) concatenates conditions sequentially and supports six editing tasks with task-aware positional embeddings. However, these methods still fall short in supporting flexible instruction-based video editing tasks.

3 Method

As shown in Figure 2, EditVerse uses a transformer with full self-attention (Chen et al., 2025c; Ju et al., 2025). Input text and vision tokens are concatenated in an interleaved manner (Section 3.1). To support this, we introduce a four-dimensional Rotary Positional Embedding spanning spatial, sequential, and temporal dimensions (Section 3.2). During training and inference, EditVerse predicts visual velocity (Esser et al., 2024; Lipman et al., 2022) to guide image or video generation via denoising (Section 3.3). The following sections elaborate on our framework and design insights.

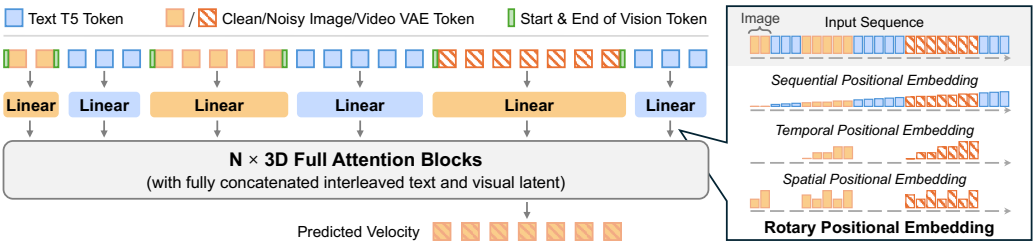


Figure 2: **Overview of EditVerse**. We design a unified framework for image and video editing and generation, which processes text and vision inputs into a unified sequence. The right part of the figure shows our positional embedding design. This framework leverages full self-attention to facilitate robust in-context learning and effective knowledge transfer among modalities.

3.1 Interleaved Text and Vision Input

Following prior works (Kingma & Welling, 2013), we encode the RGB pixel-space videos and images into a learned spatio-temporally compressed latent space by training a convolutional Variational Autoencoder (VAE) capable of both feature extraction and reconstruction. Specifically, given an input image or video $\mathbf{I}_{vision}$, the VAE compresses it into a continuous-valued latent representation with downsampling ratios r_T, r_H, r_W. Then, the vision features are patchified into a long token sequence with a $1 \times 2 \times 2$ kernel to get $\mathbf{X}_{vision} \in \mathbb{R}^{L_{vision} \times C_{vision}}$ (L_{vision} is the vision token number, C_{text} is the channel dimension of vision feature). For a given text input $\mathbf{I}_{text}$, we first generate text tokens using the Flan-T5-XXL (Chung et al., 2022) encoder. Then, we retain only the tokens that correspond directly to the input text, discarding the rest, yielding a final representation $\mathbf{X}_{text} \in \mathbb{R}^{L_{text} \times C_{text}}$ (L_{text} is the token count of $\mathbf{I}_{text}$, C_{text} is the channel dimension of Flan-T5-XXL), which saves computation while preserving the necessary information from text input.

To handle instructions composed of arbitrary combinations of text, images, and videos, we unify all modalities into a single interleaved sequence representation (shown in Figure 3). First, we project the tokens from each modality into a shared embedding space using separate single-layer linear projectors. This maps both text and visual inputs to the model's hidden dimension, C, yielding two distinct embedding matrices: $\mathbf{X}_{text\_align} \in \mathbb{R}^{L_{text} \times C}$ and $\mathbf{X}_{vision\_align} \in \mathbb{R}^{L_{vision} \times C}$. Then, we concatenate the projected embeddings to construct a unified input sequence, $\mathbf{X} \in \mathbb{R}^{L \times C}$, where L denotes the total number of text and vision tokens. The sequence preserves the original interleaved order of text and visual elements from the instruction. To explicitly indicate the location of vision tokens (images and videos) within an interleaved sequence, we add a learnable "start of vision" token and a learnable "end of vision" token at the beginning and the end of each vision token segment.

3.2 Rotary Positional Embedding

To distinguish text, image, and video from each other and to indicate their sequential positions, we design a special Rotary Positional Embedding (RoPE) that incorporates sequential, temporal and spatial (height and width) dimensions (shown in Figure 2). For each of these four positional dimensions, we apply a separate RoPE computation. (1) Sequential dimension: This dimension

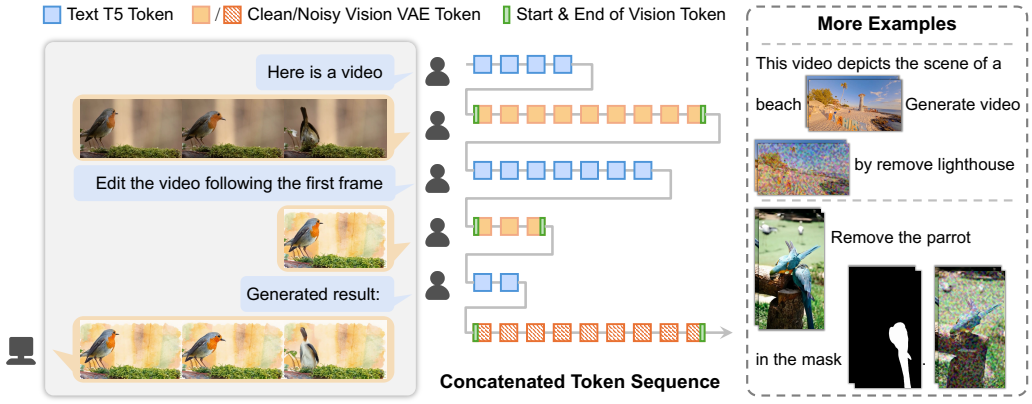


Figure 3: **Examples for the interleaved text and vision pattern**. EditVerse is capable of processing image and video inputs and outputs of arbitrary resolution, duration, and sequential positions.

captures the global position within the overall sequence, starting from 0. The value is incremented by 1 for each text token and image/video frame, up to the end of the sequence. (2) Temporal dimension: This dimension is used exclusively for video frames to encode their temporal order within a video clip. It begins at 0 and increases by 1 for each subsequent frame. For text and image inputs, this dimension remains 0. (3) Height and Width Dimensions: For images and video frames, the height and width dimensions correspond to the pixel coordinates, increasing incrementally from the top-left to the bottom-right corner (Polyak et al., 2025). The increment values reflect the number of pixels along the height and width axes. For text tokens, both dimensions are set to 0. The sequential, temporal, height, and width dimensions each compute a separate RoPE, which are assigned RoPE embedding dimensions of 12, 4, 56, 56 respectively. To better support variable-length input, we use the NTK-aware interpolation (Peng et al., 2023) in RoPE calculation for context window extension.

3.3 Training and Inference Paradigm

Given an interleaved sequence $\mathbf{X}_1 = \text{Concat}(\mathbf{X}_1^{(0)}, \mathbf{X}_1^{(1)}, \ldots, \mathbf{X}_1^{(n)})$, where each $\mathbf{X}_1^{(i)} \in \mathbb{R}^{L^{(i)} \times C}$ represents a clean image, video, or text segment. n is the total number of visual or textual segments, $L^{(i)}$ is the sequence length, C is the hidden dimension. We randomly select one image or video $\mathbf{X}_1^{(i)}$ as the generation target, optimizing with the Flow Matching (Lipman et al., 2022) training objective. In Flow Matching diffusion process, noise sample $\mathbf{X}_0^{(i)} \sim \mathcal{N}(0, 1)$ is progressively denoised into clean data $\mathbf{X}_1^{(i)}$ with $\mathbf{X}_t^{(i)} = t\mathbf{X}_1^{(i)} + (1-t)\mathbf{X}_0^{(i)}$, where timestep $t \in [0, 1]$. The learnable model u is trained to predict the velocity $\mathbf{V}_t = \frac{d\mathbf{X}_t^{(i)}}{dt}$, which can be further derived as: $\mathbf{V}_t = \frac{d\mathbf{X}_t^{(i)}}{dt} = \mathbf{X}_1^{(i)} - \mathbf{X}_0^{(i)}$. Thus, with an input sequence $\mathbf{X}_t = \text{Concat}(\mathbf{X}_1^{(0)}, \ldots, \mathbf{X}_t^{(i)}, \ldots, \mathbf{X}_1^{(n)})$, the model u with parameter Θ is optimized by minimizing the mean squared error loss $\mathcal{L}$ between the ground truth velocity and the model prediction, where $\mathbf{X}_0 = \text{Concat}(\mathbf{X}_1^{(0)}, \ldots, \mathbf{X}_0^{(i)}, \ldots, \mathbf{X}_1^{(n)})$:

$$\mathcal{L} = \mathbb{E}_{t, \mathbf{X}_0, \mathbf{X}_1} \left| u_\Theta(\mathbf{X}_t, t) - (\mathbf{X}_1 - \mathbf{X}_0)) \right|^2$$

During inference, the diffusion model first samples $\mathbf{X_0^{(i)}} \sim \mathcal{N}(0, 1)$, then uses an ordinary differential equations (ODE) solver with a discrete set of N timesteps to generate $\mathbf{X_1}$ from $\mathbf{X_0}$.

4 Data Pipeline

As shown in Table 1, EditVerse is trained on large-scale data composed of: $1.9M$ image generation samples (around $2.0B$ tokens), $3.9M$ video generation samples (around $68.8B$ tokens), $6.0M$ image editing samples (around $12.6B$ tokens), and $288K$ video editing samples (around $10.2B$ tokens).

Video Editing Data Pipeline. To address the scarcity of high-quality video editing datasets, we developed a pipeline to generate **EditVerse Editing Data**, which can be applied to obtain video editing pairs from any video input. (1) **Object Removal and Addition**: We extract object masks using Grounded-SAM-2 (Ravi et al., 2024; Ren et al., 2024), filter candidates by name, area, and confidence, and remove objects with DiffuEraser (Li et al., 2025), forming removal/addition pairs. (2) **Object Replacement**: Object masks are obtained with Grounded-SAM-2, then a Vision-Language Model (Wang et al., 2024) suggests transformations, and VACE (Jiang et al., 2025) inpaints the masked region with dynamic mask adjustments. (3) **Style Transfer**: We first apply image style transfer to the first frame, then generate full videos using VACE's depth-guided first-frame-to-video feature. (4) **Camera Change**. We use ReCamMaster (Bai et al., 2025) to generate videos with 10 different camera movements to obtain data pairs with camera change. (5) **Mask Detection**. We convert object removal, object addition, and object replacement data from (1) and (2) using the prompt template: "I want to [edit prompt]. Detect the region that needs to be edited". (6) **Propagation**. We extract the

Image Datasets		
	Dataset	**#Samples**
Edit	MagicBrush	$9K$
	ShareGPT-4o-Image	$46K$
	Object Removal & Addition‡	$119K$
	OmniEdit	$186K(1.2M^*)$
	ImgEdit	$246K(1.2M^*)$
	NHR-Edit	$358K$
	UltraEdit	$500K$
	AnyEdit	$1.2M(2.5M^*)$
	GPT-Image-Edit-1.5M	$1.5M$
	Instruction-based Editing‡	$1.8M$
Gen	BLIP3-o 60K	$60K$
	LLaVA-pretrain	$500K$
	Text-to-Image‡	$610K$
	LLaVA-next fine-tuning	$700K$
Video Datasets		
	Dataset	**#Samples**
Edit	Señorita-2M	$56K(2M^*)$
	EditVerse Editing Data	$232K(1.3M^*)$
Gen	Text-to-Video‡	$223K$
	Customization	$740K$
	EditVerse Gen Data	$3.0M$

*Dataset volume before filtering. ‡ Internal dataset.

Table 1: **Statistics of the training datasets.** Detailed information in Table 10.

first edited frame from object removal, object addition, object replacement, and style transfer data from (1), (2), and (3) to serve as propagation input.

In addition, we incorporate data from the open-source dataset Señorita-2M (Zi et al., 2025). However, we observe a relatively low success rate in this dataset, which necessitated extensive filtering.

Video Generation Data Pipeline. Since we start from a pretrained model capable of text-to-image and text-to-video tasks, we only use a small scale of pure text-based generation data ($223K$ samples for text-to-video) to preserve the model's inherent generative capabilities. For controllable generation, we create control-to-video and video-to-control pairs with depth, sketch, and pose as control, annotated with Depth Anything v2 (Yang et al., 2024a), RTMPose (Jiang et al., 2023), and OpenCV Canny Edge Detection (Itseez, 2015). We also include image-to-video and video inpainting data annotated with Grounded-SAM-2 (Ravi et al., 2024; Ren et al., 2024). Together, these datasets form **EditVerse Gen Data**. We supplement this with a video customization dataset (Cai et al., 2025b).

Image Editing. After reviewing the data quality of existing image editing datasets, we incorporate 8 high-quality open-source datasets: MagicBrush (Zhang et al., 2023a), ShareGPT-4o-Image (Chen et al., 2025b), OmniEdit (Wei et al., 2024), ImgEdit (Ye et al., 2025a), NHR-Edit (Kuprashevich et al., 2025), UltraEdit (Zhao et al., 2024), AnyEdit (Yu et al., 2024), and GPT-Image-Edit-1.5M (Wang et al., 2025). In addition, we incorporate two internal image editing datasets: one focused on image addition and removal, and the other on free-form instruction-based image editing.

Image Generation. For text-to-image, we include $610K$ internal text-to-image samples as well as several open-source image understanding datasets, BLIP3-o 60K (Chen et al., 2025a), LLaVA-pretrain (Liu et al., 2023), and LLaVA-next fine-tuning (Liu et al., 2024a), that contain high-quality text annotations, which can improve the understanding ability of editing instructions.

Data Filtering. To select high-quality training data from model-generated videos, we used a VLM (Wang et al., 2024) to score 0 to 10 for generated data across instruction adherence, context preservation, sharpness, temporal consistency, artifacts, object integrity, aesthetics, and physical plausibility. Manual inspection of score–quality relationships guided threshold selection for the final dataset. Table 1 shows that our pipeline achieves a sixfold higher retention rate than Señorita-2M.

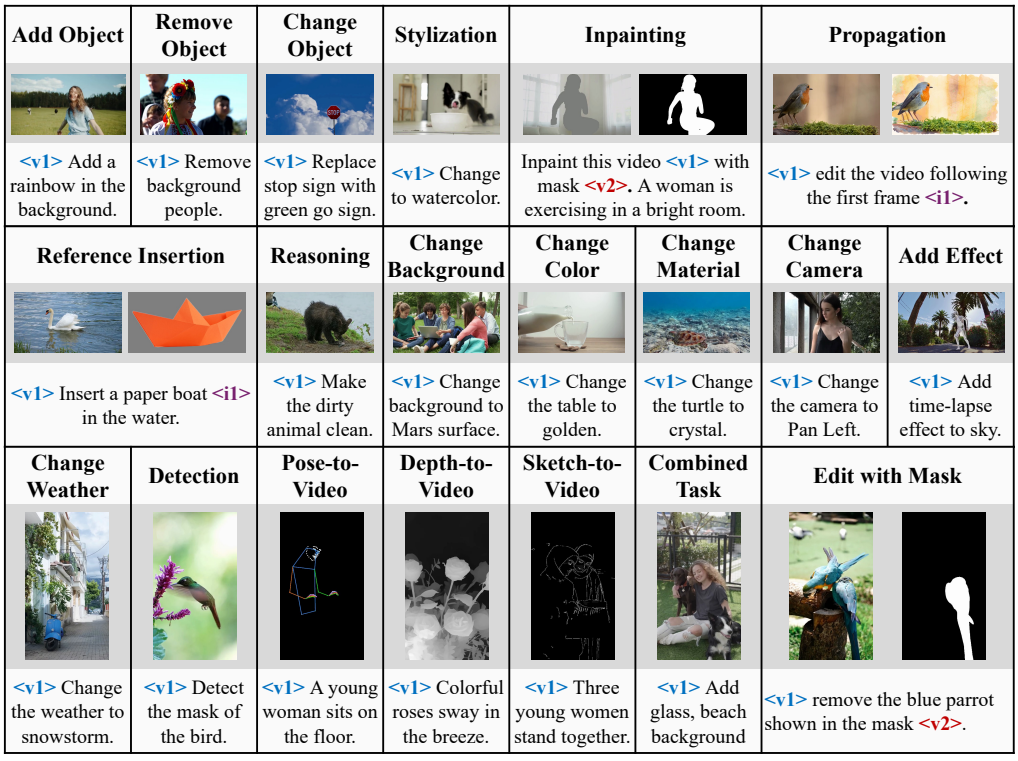


Figure 4: **Examples from the proposed EditVerseBench**. EditVerseBench includes 200 editing pairs, evenly distributed across 20 editing categories as well as horizontal and vertical orientations.

5 EXPERIMENTS

5.1 EDITVERSEBENCH

Commonly used video editing benchmarks (*e.g.*, V2VBench (Sun et al., 2024), TGVE (Wu et al., 2023b; Singer et al., 2024b)) only consist of square videos and are primarily designed for training-free editing (Geyer et al., 2023; Yatim et al., 2024) rather than instruction-based editing. Moreover, such benchmarks do not adequately cover the diverse editing tasks commonly encountered in real-world video editing scenarios. To address these limitations, we propose EditVerseBench, a comprehensive instruction-based video editing benchmark composed of 20 distinct instruction-based video editing tasks. We manually selected 100 videos from a free stock website (Pixabay, 2025) that cover a variety of scenes, including 50 horizontal and 50 vertical videos. For each video, we randomly select two editing instructions from the 20 editing tasks. This results in a total of 200 editing pairs (5 horizontal and 5 vertical videos per editing task). We show one example from each editing category in Figure 4. To evaluate the editing performance on our proposed EditVerseBench, we use 6 metrics covering four aspects: VLM evaluation, video quality (frame-wise Pick Score (Kirstain et al., 2023)), text alignment (CLIP (Radford et al., 2021) text-image and ViCLIP (Wang et al., 2023b) text-video alignment), and temporal consistency (frame-wise CLIP (Radford et al., 2021) and DINO (Caron et al., 2021) consistency). Details can be found in the Appendix.

5.2 COMPARISON TO PREVIOUS METHODS

We show comparisons of EditVerseBench and TGVE+ (Singer et al., 2024b) in this section. More comparisons (image generation and editing, video generation) are provided in the Appendix.

Comparison on EditVerseBench. Since InsV2V (Cheng et al., 2023) and Lucy Edit (Team, 2025) are the only open-source instruction-based video editing methods that exactly match our setting, we further selected two well-known training-free methods, TokenFlow (Geyer et al., 2023) and STDF (Yatim et al., 2024), as well as a first-frame propagation method, Señorita-2M (Zi et al., 2025), for comparison on EditVerseBench. We use the first frame of our results as input to Señorita-2M. Moreover, we also compare to a commercial model, Runway Aleph (Runway, 2025). As shown in Table 2, EditVerse outperforms previous research models on all metrics, demonstrating the effectiveness of our proposed method. Figure 6 shows visual comparisons on EditVerseBench. We further conduct a user study to assess human judgments of editing performance. The evaluation criteria include (i) instruction alignment, (ii) preservation of unedited regions, and (iii) overall video quality. We collected $3,000$ pairwise ratings comparing EditVerse against each of the other methods, with the results summarized in Figure 5. We find the user study result is more aligned with the VLM metric in automatic evaluation.

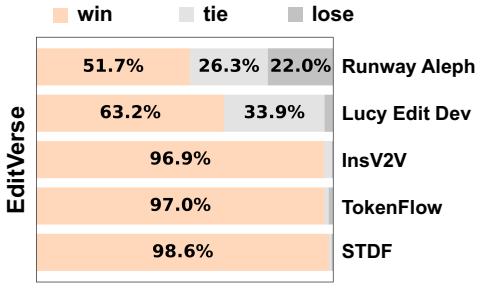


Figure 5: **User study on EditVerseBench.**

Method	VLM evaluation	Video Quality	Text Alignment		Temporal Consistency		Human
	Editing Quality ↑	Pick Score ↑	Frame ↑	Video ↑	CLIP ↑	DINO ↑	Rating ↓
Attention Manipulation (Training-free)							
TokenFlow	5.26	19.73	25.57	22.70	98.36	98.09	5
STDF	4.41	19.45	25.24	22.26	96.04	95.22	6
First-Frame Propagation (w/ End-to-End Training)							
Señorita-2M	6.97	19.71	26.34	23.24	98.05	97.99	-
Instruction-Guided (w/ End-to-End Training)							
InsV2V	5.21	19.39	24.99	22.54	97.15	96.57	4
Lucy Edit	5.89	19.67	26.00	23.11	98.49	98.38	3
EditVerse	**7.65**	**20.07**	**26.73**	**23.93**	**98.56**	**98.42**	**1**
Closed-Source Commercial Models							
Runway Aleph	7.44	20.42	27.70	24.27	98.94	98.60	2

Table 2: **Quantitative comparison on EditVerseBench**. We compare two training-free methods, one first-frame propagation method, one instruction-guided video editing method, and additionally one commercial model as reference. Best non-commercial results are in **bold**.

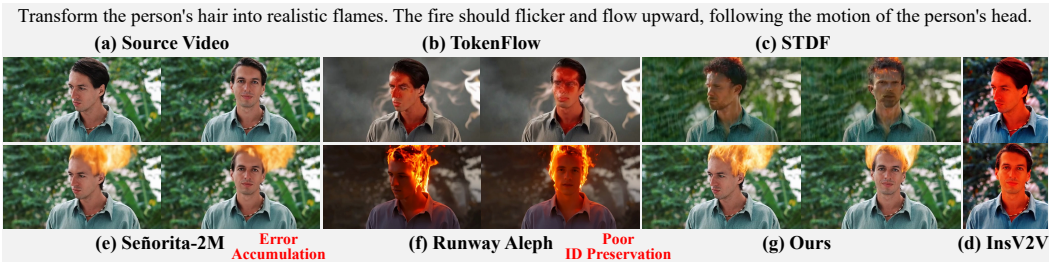


Figure 6: **Visualization of EditVerse and other video editing methods**. EditVerse shows stronger context preservation and edit faithfulness. Complete comparisons are in the Appendix.

Comparison on TGVE+. Following Movie Gen (Polyak et al., 2025), we evaluate EditVerse on TGVE+ (Singer et al., 2024b). Specifically, we follow previous works and measure (i) ViCLIP$_{dir}$: text-video direction similarity, which evaluates the alignment between changes in captions and corresponding changes in the videos, and (ii) ViCLIP$_{out}$: output similarity, which measures the similarity between the edited video and the output caption. As shown in Table 3, EditVerse surpasses previous methods on both metrics. It is worth noting that all TGVE+ videos are square, whereas our training data does not include any square video editing samples.

Method	ViCLIP$_{dir}$ ↑	ViCLIP$_{out}$ ↑
Tune-A-Video (Wu et al., 2023a)	0.131	0.242
TokenFlow (Geyer et al., 2023)	0.128	0.237
STDF (Yatim et al., 2024)	0.093	0.227
Fairy (Wu et al., 2024)	0.140	0.197
InsV2V (Cheng et al., 2023)	0.174	0.236
SDEdit (Meng et al., 2021)	0.131	0.241
EVE (Singer et al., 2024a)	0.198	0.251
Movie Gen Edit (Polyak et al., 2025)	**0.225**	0.248
EditVerse (Ours)	**0.225**	**0.252**

Table 3: **Quantitative comparison on TGVE+.** Results show superior performance of EditVerse.

5.3 Analysis of Emergent Ability

Emergent ability is one of the most exciting phenomena observed in large-scale model training, arising as data and model capacity increase. In this section, we specifically analyze this phenomenon.

Demonstration of emergent ability. We show the emergent ability of video editing in two aspects: (1) the model can perform editing tasks that were not present in the training data, and (2) for tasks included in the training data, the model's performance can even surpass the ground-truth quality.

The video editing training data covers a limited set of tasks mentioned in Section 4, including object modification (addition, removal, or replacement), style transfer, camera change, mask detection, and propagation. However, as shown in Figure 1, our model is capable of performing tasks beyond the training distribution (*e.g.*, change material, change weather, add effects). Furthermore, it can also handle multiple tasks (*e.g.*, reference insertion by combining customization with inpainting).

We also find that EditVerse can surpass the ground-truth training data in both quality and success rate by leveraging knowledge from the image generation/editing and video generation domains. We show two examples for object replacement and style transfer in Figure 7.

The source of emergent ability. We further analyze the source of emergent ability by performing ablations on the training data. We find that removing either image generation/editing data or video generation data negatively impacts video editing quality. Specifically, image generation/editing data helps the model better understand editing instructions and perform more diverse edits, while video

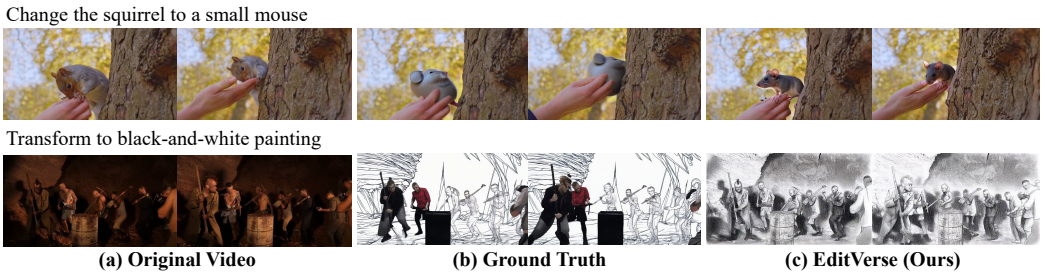


Figure 7: **Compare EditVerse generated results with ground truth.** Results show EditVerse can surpass ground-truth data quality by extracting knowledge from image and video generation data.

Figure 8: **Visualization of ablation on training data.** Image data plays a critical role.

Training Datasets			VLM evaluation	Video Quality	Text Alignment		Temporal Consistency	
Image	Video Gen	Video Edit	Editing Quality	Frame	Video	Pick Score	CLIP	DINO
✓	✓	✗	3.62	18.64	22.31	20.44	93.48	90.27
✗	✗	✓	5.76	19.41	25.22	22.37	98.26	97.83
✓	✗	✓	6.52	19.81	25.78	22.63	98.24	97.97
✗	✓	✓	6.40	19.72	25.37	22.51	**98.77**	**98.60**
✓	✓	✓	**6.95**	**19.99**	**26.26**	**23.81**	98.68	98.44

Table 4: **Ablation study on training data**. We run $20K$ steps with the same setup as in Section A.1. Results indicate that both image and video generation data are crucial to video editing performance.

generation data improves temporal consistency and motion modeling. Figure 8 and Table 4 illustrate the differences with and without image generation/editing and video data. Interestingly, EditVerse is able to perform some video editing tasks even without training on video editing data (as shown at the top row in Table 4), though quality and success rate are limited.

5.4 Ablation Study on Model Design

Compared with previous approaches (Chen et al., 2025c), our model contains two key designs: the interleaved formulation and the special positional embedding. Therefore, as shown in Table 5, we perform an ablation study by (i) removing the interleaved formulation (placing all images and videos at the end of the sequence) and (ii) removing the sequential dimension RoPE. Results show that both designs have a large influence on the model's performance, especially for the text alignment and editing quality. This is because the temporal consistency and video quality are partly inherited from the base model, while text alignment and editing quality largely depend on the in-context learning ability coming from the model design. Only the interleaved input format combined with sequential positional embedding can best enable the model to be aware of the relationships among different modalities (*e.g.*, knowledge transfer of image and video), thereby achieving optimal performance.

Model Design		VLM Evaluation	Video Quality	Text Alignment		Temporal Consistency	
Interleave	Sequential PE	Editing Quality	Pick Score	Frame	Video	CLIP	DINO
✓	✗	6.42	19.89	25.77	22.74	98.62	98.43
✗	✓	6.84	19.92	26.19	23.51	**98.69**	98.39
✓	✓	**6.95**	**19.99**	**26.26**	**23.81**	98.68	**98.44**

Table 5: **Ablation study on interleaved formation and sequential RoPE**. For each model variant, we run $20K$ steps with the same experimental setting detailed in Section A.1.

6 Conclusion

This paper introduced EditVerse, a unified framework designed to address the architectural and data-scarcity challenges in universal video generation and editing. By representing text, images, and videos as a single interleaved token sequence, our model leverages full self-attention for robust in-context learning, enabling flexible inputs/outputs of arbitrary resolution and duration, while facilitating knowledge transfer from the data-abundant image domain to the video domain.

We further developed a data pipeline for obtaining high-quality video editing samples and proposed EditVerseBench, a benchmark covering diverse editing tasks. Results show that EditVerseachieves state-of-the-art performance. These findings validate that a unified architecture can mitigate video data limitations via cross-modal learning, revealing emergent abilities and paving the way for more general multimodal foundation models. Limitations and future work are discussed in the Appendix.

REFERENCES

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

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. *arXiv preprint arXiv:2503.11647*, 2025.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18392–18402, 2023.

Qi Cai, Jingwen Chen, Yang Chen, Yehao Li, Fuchen Long, Yingwei Pan, Zhaofan Qiu, Yiheng Zhang, Fengbin Gao, Peihan Xu, et al. Hidream-i1: A high-efficient image generative foundation model with sparse diffusion transformer. *arXiv preprint arXiv:2505.22705*, 2025a.

Yuanhao Cai, He Zhang, Xi Chen, Jinbo Xing, Yiwei Hu, Yuqian Zhou, Kai Zhang, Zhifei Zhang, Soo Ye Kim, Tianyu Wang, et al. Omnivcus: Feedforward subject-driven video customization with multimodal control conditions. *arXiv preprint arXiv:2506.23361*, 2025b.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 22560–22570, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9650–9660, 2021.

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

Junying Chen, Zhenyang Cai, Pengcheng Chen, Shunian Chen, Ke Ji, Xidong Wang, Yunjin Yang, and Benyou Wang. Sharegpt-4o-image: Aligning multimodal models with gpt-4o-level image generation, 2025b. URL https://arxiv.org/abs/2506.18095.

Xi Chen, Zhifei Zhang, He Zhang, Yuqian Zhou, Soo Ye Kim, Qing Liu, Yijun Li, Jianming Zhang, Nanxuan Zhao, Yilin Wang, et al. Unireal: Universal image generation and editing via learning real-world dynamics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12501–12511, 2025c.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. URL https://arxiv.org/abs/2210.11416.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. Seed-data-edit technical report: A hybrid dataset for instructional image editing. *arXiv preprint arXiv:2405.04007*, 2024.

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arxiv:2307.10373*, 2023.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Xuanhua He, Quande Liu, Zixuan Ye, Weicai Ye, Qiulin Wang, Xintao Wang, Qifeng Chen, Pengfei Wan, Di Zhang, and Kun Gai. Fulldit2: Efficient in-context conditioning for video diffusion transformers. *arXiv preprint arXiv:2506.04213*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jiahao Hu, Tianxiong Zhong, Xuebo Wang, Boyuan Jiang, Xingye Tian, Fei Yang, Pengfei Wan, and Di Zhang. Vivid-10m: A dataset and baseline for versatile and interactive video local editing. *arXiv preprint arXiv:2411.15260*, 2024.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.

Itseez. Open source computer vision library. `https://github.com/itseez/opencv`, 2015.

Tao Jiang, Peng Lu, Li Zhang, Ningsheng Ma, Rui Han, Chengqi Lyu, Yining Li, and Kai Chen. Rtmpose: Real-time multi-person pose estimation based on mmpose, 2023. URL `https://arxiv.org/abs/2303.07399`.

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

Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. Direct inversion: Boosting diffusion-based editing with 3 lines of code. *arXiv preprint arXiv:2310.01506*, 2023a.

Xuan Ju, Ailing Zeng, Chenchen Zhao, Jianan Wang, Lei Zhang, and Qiang Xu. Humansd: A native skeleton-guided diffusion model for human image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15988–15998, 2023b.

Xuan Ju, Weicai Ye, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, and Qiang Xu. Fulldit: Multi-task video generative foundation model with full attention. *arXiv preprint arXiv:2503.19907*, 2025.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. *Advances in neural information processing systems*, 36:36652–36663, 2023.

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

Maksim Kuprashevich, Grigorii Alekseenko, Irina Tolstykh, Georgii Fedorov, Bulat Suleimanov, Vladimir Dokholyan, and Aleksandr Gordeev. Nohumansrequired: Autonomous high-quality image editing triplet mining. *arXiv preprint arXiv:2507.14119*, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Xiaowen Li, Haolan Xue, Peiran Ren, and Liefeng Bo. Diffueraser: A diffusion model for video inpainting. *arXiv preprint arXiv:2501.10018*, 2025.

Zhen Li, Mingdeng Cao, Xintao Wang, Zhongang Qi, Ming-Ming Cheng, and Ying Shan. Photomaker: Customizing realistic human photos via stacked id embedding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8640–8650, 2024.

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

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8599–8608, 2024b.

Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, et al. Generative video propagation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 17712–17722, 2025a.

Shiyu Liu, Yucheng Han, Peng Xing, Fukun Yin, Rui Wang, Wei Cheng, Jiaqi Liao, Yingming Wang, Honghao Fu, Chunrui Han, et al. Step1x-edit: A practical framework for general image editing. *arXiv preprint arXiv:2504.17761*, 2025b.

Ilya Loshchilov, Frank Hutter, et al. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5(5):5, 2017.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

OpenAI. Hello gpt-4o. Blog post, May 2024. URL https://openai.com/index/hello-gpt-4o/.

Bowen Peng, Jeffrey Quesnelle, Honglu Fan, and Enrico Shippole. Yarn: Efficient context window extension of large language models. *arXiv preprint arXiv:2309.00071*, 2023.

Pixabay. Pixabay: Free images, videos, music, and more. https://pixabay.com/, 2025. Accessed: 2025-09-11.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix

Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2025. URL https://arxiv.org/abs/2410.13720.

Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15932–15942, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos, 2024. URL https://arxiv.org/abs/2408.00714.

Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, Zhaoyang Zeng, Hao Zhang, Feng Li, Jie Yang, Hongyang Li, Qing Jiang, and Lei Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

Runway. Introducing runway aleph. urlhttps://runwayml.com/research/introducing-runway-aleph, July 25 2025. Accessed: 2025-09-10.

Uriel Singer, Amit Zohar, Yuval Kirstain, Shelly Sheynin, Adam Polyak, Devi Parikh, and Yaniv Taigman. Video editing via factorized diffusion distillation. In *European Conference on Computer Vision*, pp. 450–466. Springer, 2024a.

Uriel Singer, Amit Zohar, Yuval Kirstain, Shelly Sheynin, Adam Polyak, Devi Parikh, and Yaniv Taigman. Video editing via factorized diffusion distillation. In *European Conference on Computer Vision*, pp. 450–466. Springer, 2024b.

Wenhao Sun, Rong-Cheng Tu, Jingyi Liao, and Dacheng Tao. Diffusion model-based video editing: A survey. *arXiv preprint arXiv:2407.07111*, 2024.

DecartAI Team. Lucy edit: Open-weight text-guided video editing. 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Wen Wang, Yan Jiang, Kangyang Xie, Zide Liu, Hao Chen, Yue Cao, Xinlong Wang, and Chunhua Shen. Zero-shot video editing using off-the-shelf image diffusion models. *arXiv preprint arXiv:2303.17599*, 2023a.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023b.

Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. Gpt-image-edit-1.5 m: A million-scale, gpt-generated image dataset. *arXiv preprint arXiv:2507.21033*, 2025.

Cong Wei, Zheyang Xiong, Weiming Ren, Xeron Du, Ge Zhang, and Wenhu Chen. Omniedit: Building image editing generalist models through specialist supervision. In *The Thirteenth International Conference on Learning Representations*, 2024.

Bichen Wu, Ching-Yao Chuang, Xiaoyan Wang, Yichen Jia, Kapil Krishnakumar, Tong Xiao, Feng Liang, Licheng Yu, and Peter Vajda. Fairy: Fast parallelized instruction-guided video-to-video synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8261–8270, 2024.

Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7623–7633, 2023a.

Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, et al. Cvpr 2023 text guided video editing competition. *arXiv preprint arXiv:2310.16003*, 2023b.

Shitao Xiao, Yueze Wang, Junjie Zhou, Huaying Yuan, Xingrun Xing, Ruiran Yan, Chaofan Li, Shuting Wang, Tiejun Huang, and Zheng Liu. Omnigen: Unified image generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 13294–13304, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024a.

Ling Yang, Bohan Zeng, Jiaming Liu, Hong Li, Minghao Xu, Wentao Zhang, and Shuicheng Yan. Editworld: Simulating world dynamics for instruction-following image editing. *arXiv preprint arXiv:2405.14785*, 2024b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024c.

Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8466–8476, 2024.

Yang Ye, Xianyi He, Zongjian Li, Bin Lin, Shenghai Yuan, Zhiyuan Yan, Bohan Hou, and Li Yuan. Imgedit: A unified image editing dataset and benchmark. *arXiv preprint arXiv:2505.20275*, 2025a.

Zixuan Ye, Xuanhua He, Quande Liu, Qiulin Wang, Xintao Wang, Pengfei Wan, Di Zhang, Kun Gai, Qifeng Chen, and Wenhan Luo. Unic: Unified in-context video editing. *arXiv preprint arXiv:2506.04216*, 2025b.

Jaehong Yoon, Shoubin Yu, and Mohit Bansal. Raccoon: A versatile instructional video editing framework with auto-generated narratives. *arXiv preprint arXiv:2405.18406*, 2024.

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

Fan Zhang, Shulin Tian, Ziqi Huang, Yu Qiao, and Ziwei Liu. Evaluation agent: Efficient and promptable evaluation framework for visual generative models. *arXiv preprint arXiv:2412.09645*, 2024.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in Neural Information Processing Systems*, 36:31428–31449, 2023a.

Kai Zhang, Peng Wang, Sai Bi, Jianming Zhang, and Yuanjun Xiong. Knapformer: An online load balancer for efficient diffusion transformers training. *arXiv preprint arXiv:2508.06001*, 2025a.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 3836–3847, 2023b.

Zechuan Zhang, Ji Xie, Yu Lu, Zongxin Yang, and Yi Yang. In-context edit: Enabling instructional image editing with in-context generation in large scale diffusion transformer. *arXiv preprint arXiv:2504.20690*, 2025b.

Haozhe Zhao, Xiaojian Shawn Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

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

Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Se\~ norita-2m: A high-quality instruction-based dataset for general video editing by video specialists. *arXiv preprint arXiv:2502.06734*, 2025.

A APPENDIX

The appendix includes the following content:

Full Comparison Results Webpage

The full comparison results of EditVerse and existing approaches are on this anonymous webpage: http://editverse-anonymous.s3-website-us-east-1.amazonaws.com/. We provide a side-by-side visualization of EditVerse comparing to Runway Aleph (Runway, 2025), Lucy Edit (Team, 2025), InsV2V (Cheng et al., 2023), STDF (Yatim et al., 2024), TokenFlow (Geyer et al., 2023), and Señorita-2M (Zi et al., 2025) in this webpage. Each visualization webpage contains 20 editing categories.

Supplemental ZIP file

- **Demos/Demo.mp4 File**: A demo video showcasing EditVerse.
- **Demos/Demo_WebPage Folder**: A webpage showing results on image generation, image editing, video generation, and video editing. The entry point is *click_me.html*.
- **Evaluation/EditVerseBench.json File**: The complete EditVerseBenchcontent.
- **Evaluation/Quantitative_Evaluation_Results Folder**: This folder provides detailed quantitative evaluation results of each method in CSV format.
- **Evaluation/User_Study.xlsx**: Raw user study results compared to Runway Aleph.

Appendix PDF

- **Section A.1**: Implementation and Evaluation Details
- **Section A.2**: Additional Experiments (evaluation of image editing, video editing, image generation, and video generation on existing benchmarks)
- **Section A.3**: Detailed Training Data (more detailed descriptions of training data)
- **Section A.4**: Limitation and Future Works

A.1 IMPLEMENTATION AND EVALUATION DETAILS

Implementation Details. EditVerse is trained on a $2B$ dense transformer architecture similar to LLaMA 3 (Dubey et al., 2024). It is initially pretrained on text-to-image and text-to-video data to get basic generative capabilities at a resolution of 360p. Then, we train the model on our dataset as listed in Section 4. For each image/video, we resize it according to its original aspect ratio so that its area falls between 256×256 and 512×512. During training, we use a global batch size of 256 and train for around $56K$ steps. We use AdamW optimizer (Loshchilov et al., 2017) with hyperparameters set to $\beta_1 = 0.9, \beta_2 = 0.95$, a peak learning rate of $8e^{-6}$, and weight decay of 0.01. We use a warm-up of $2K$ steps and a cosine decay learning schedule, decreasing the learning rate to the minimum of $1e^{-6}$. We set the gradient clipping norm to 1.0 and disable gradient clipping during the warm-up stage. Since the training data consist of token sequences with variable lengths, making it difficult to form batches, we adopt the packing strategy introduced in KnapFormer (Zhang et al., 2025a). During inference, we use a classifier-free guidance scale of 5.0, applying it only to text conditions. The inference timestep is set to 50 for the balance of performance and inference speed. It takes around $30GB$ memory and 118 seconds to edit one $360p$ video on NVIDIA A100 $80GB$.

Automatic Evaluation. To provide a comprehensive and robust evaluation of instruction-based video editing, we employ a suite of six metrics spanning four aspects: overall editing quality evaluated by a Vision-Language Model (VLM), video quality, text alignment, and temporal consistency.

- **Overall Editing Quality Evaluated by VLM**: We employ a state-of-the-art Vision-Language Model (VLM), GPT-4o OpenAI (2024), to serve as an automated judge. We uniformly sample three frames from each source and edited video pair. For each sample, the VLM receives the source frame, the edited frame, and the text instruction. It is prompted to score the edit from 0 (worst) to 3 (best) across three key criteria: Prompt Following, Edit Quality, and Background Consistency, which are summed to get the overall score for the current frame. The VLM score for the entire video is the average of the three frame scores.
- **Video Quality**: We employ PickScore (Kirstain et al., 2023), which shows a strong correlation with human judgment on image quality and prompt alignment. We calculate the PickScore for each frame and average these scores across the entire video.

- **Text Alignment**: Text alignment evaluates how well the edited video reflects the given text instruction. We measure this at both the frame level and the video level.

  CLIP Text-Image Alignment: This metric assesses the semantic alignment between the editing instruction and each frame of the output video. We encode the text instruction using the CLIP text encoder and each frame using the CLIP vision encoder to get the respective feature vectors. The final score is the average cosine similarity across all frames.

  ViCLIP Text-Video Alignment: Frame-wise alignment does not capture the temporal aspects of the instruction. Therefore, we use ViCLIP (Wang et al., 2023b) to compute an embedding for the entire video clip and measure its cosine similarity with the text instruction's embedding. This measures how well the video as a whole corresponds to the prompt.

- **Temporal Consistency**: Temporal consistency measures the smoothness and coherence of the edited video, penalizing flickering, jarring transitions, and inconsistent object appearances between frames. We assess this using feature similarity between adjacent frames.

  Frame-wise CLIP Consistency: We use the ViT-L/14 vision encoder from CLIP (Radford et al., 2021) to extract features of each frame in the edited video. The consistency score is calculated as the average cosine similarity between the features of all adjacent frames.

  Frame-wise DINO Consistency: To capture more fine-grained structural and textural consistency, we extract features from a pre-trained DINOv2 model (Caron et al., 2021). Similarly, the consistency score is calculated as the average cosine similarity.

User Study. To validate our automated metrics and directly measure human perceptual preferences, we conducted a comprehensive user study. The user study was outsourced to a professional external vendor, who recruited 20 annotators coming from diverse non-expert backgrounds. Each comparison pair was independently evaluated by 3 different annotators, and each annotator labeled 150 comparison pairs in total. To faithfully capture end-user preferences, we intentionally kept the instruction minimal and user-centric, providing only the brief prompt shown below, without additional technical guidance. This setup was chosen to reflect how typical users would judge the outputs rather than imposing task-specific expertise. Although we did not compute a formal inter-annotator agreement statistic, the redundancy of three independent judgments per pair helps mitigate noise and increases the robustness of the aggregated preferences. Concrete visualization examples of the annotation interface are provided in Demos/DemoWebPageFolder, and the raw annotation results for RunwayAleph are available in Evaluation/UserStudy.xlsx, which together further support the credibility and transparency of our human study. Using a web-based interface, participants were shown pairs of edited videos, labeled "Result 1" and "Result 2", each generated by different models using the same source video and text instruction. Their task was to compare the two videos and choose among "Result 1 is better," "Result 2 is better," or "They are about the same" across three evaluation criteria: (1) *Text-Instruction Alignment*: Which video better follows the provided instruction? (2) *Preservation of Unedited Regions*: Are unmodified parts of the video accurately preserved, with minimal distortion or artifacts? Ideally, edits should only affect the intended object or region. Select the one that preserves better. (3) *Aesthetic Quality*: Which video is more visually appealing in terms of realism, smoothness, and overall perceptual quality? A video is considered the winner of a comparison if it achieves a majority of wins across these three criteria. We find the user study shows a Pearson Correlation of 0.84 with automatic VLM evaluation, indicating a very strong positive correlation between the user study and VLM rankings.

Example of Data Filtering Prompt. We provide a template for the data filtering prompt, which serves as a guide for evaluating the quality of AI-generated video edits:

You are an expert in AI-generated video quality assessment. Your task is to evaluate video editing performance based on the provided frames and the instructions. Original and edited frames are provided. The editing instruction is [editing instruction]. Evaluation criteria is instruction adherence (0-10): 10 - Instruction is perfectly and completely executed. 5 - Instruction is partially followed, with some elements missing or misinterpreted. 0 - Instruction is completely ignored.

Because different filtering dimensions require slightly different emphases, the text prompts are adjusted to reflect the specific type of filtering being carried out.

Text Instructions of Figure 1. We list the editing instructions that were used in Figure 1 from top-to-bottom, and left-to-right: (1) Add a pair of sparkling feathered wings to the person who is running. (2) Turn the man into a running cartoon leopard. (3) Turn the person into a translucent,

crystal-glass-like form. (4) Remove the woman. (5) Transform the woman's dress into a golden, fluid-like form with flames. (6) Turn into cartoon form. (7) Change the water to blue. (8) Change the camera pose to Pan Left. (9) Change the woman's slip dress to red and add a gentle snowfall effect. (10) Turn the grass into a reflective water surface. (11) Dramatically transform the scene by adding animated fiery embers and gentle flame wisps subtly dancing along the edges of the rose petals, giving the impression that the flower is being ignited by magical fire without harm, creating a surreal and striking contrast of beauty and intensity. (12) Insert a paper boat in the water [source image] A graceful white swan glides silently across the still surface of a clear lake, its long neck curved in a gentle arch and its feathers shining with a soft pearly sheen in the sunlight. Beside it, an orange paper boat drifts lightly, its sharp folds and pointed bow creating small ripples as it floats. (13) Two vibrant blue parrots are perched closely together on a tree stump. They appear to be pecking or searching for food in the crevice of the wood. The background shows a sunlit, green outdoor area with other birds visible in the distance, giving the scene a lively and natural atmosphere. (14) Change the weather to a heavy snowfall. (15) Detect the mask of the bird. (16) A young beautiful woman wearing a white hijab and a long white top sits quietly on the floor. She is reading from an open book, which rests on an intricately carved wooden stand. Her expression is calm and focused as she moves her finger along the lines of text, absorbed in her reading. The peaceful setting, with soft light and a tiled background, suggests a moment of reflection or prayer. (17) A quiet tree-lined path stretches into the distance, bathed in soft sunlight. Green leaves form a canopy overhead, while brown and yellow leaves are scattered across the ground. The scene feels calm and peaceful, inviting a slow walk or a moment of reflection in nature.

A.2 Additional Experiments

Video Editing. We provide a quantitative comparison on V2VBench (Sun et al., 2024) in Table 6. Note that all V2VBench videos are square, whereas our training data does not include any square video editing samples. Our method achieves the best or competitive results across most metrics.

Method	Frames Quality ↑	Semantic Consistency ↑	Object Consistency ↑	Frames Text Alignment ↑	Frames Pick Score ↑	Video Text Alignment ↑	Motion Alignment ↑
Network and Training Paradigm							
Tune-A-Video	**5.001**	0.934	0.917	27.513	20.701	0.254	-5.599
SimDA	4.988	0.940	0.929	26.773	20.512	0.248	-4.756
VidToMe	4.988	0.949	0.945	26.813	20.546	0.240	-3.203
VideoComposer	4.429	0.914	0.905	28.001	20.272	0.262	-8.095
MotionDirector	4.984	0.940	0.951	27.845	20.923	0.262	-3.088
EditVerse (Ours)	4.957	**0.959**	**0.960**	**28.587**	**21.117**	**0.273**	**-3.015**
Attention Feature Injection							
Video-P2P	4.907	0.943	0.926	23.550	19.751	0.193	-5.974
Vid2Vid-Zero	5.103	0.919	0.912	**28.789**	**20.950**	**0.270**	-4.175
Fate-Zero	5.036	**0.951**	**0.952**	25.065	20.707	0.225	**-1.439**
TokenFlow	5.068	0.947	0.943	27.522	20.757	0.254	-1.572
FLATTEN	4.965	0.943	0.949	27.156	20.745	0.251	-1.446
FRESCO	**5.127**	0.908	0.896	25.639	20.239	0.223	-5.241
Diffusion Latent Manipulation							
Text2Video-Zero	5.097	0.899	0.894	**29.124**	20.568	0.265	-17.226
Pix2Video	5.075	0.946	0.944	28.731	**21.054**	**0.271**	-2.889
ControlVideo	**5.404**	**0.959**	**0.948**	28.551	20.961	0.261	-9.396
Rerender	5.002	0.872	0.863	27.379	20.460	0.261	-4.959
RAVE	5.077	0.926	0.936	28.190	20.865	0.255	**-2.398**

Table 6: **Quantitative comparison on V2VBench (Sun et al., 2024).** Methods are grouped into three categories: (i) Network and Training Paradigm, (ii) Attention Feature Injection, and (iii) Diffusion Latent Manipulation. Local best are in **bold**. Global best are underlined.

Image Editing. We present a quantitative evaluation of EditVerse for the task of image editing using the ImgEdit-Bench, as summarized in Table 7. The results demonstrate that EditVerse achieves highly competitive performance in image editing, surpassing a wide range of existing approaches (Deng et al., 2025; Liu et al., 2025b). This highlights the effectiveness of our method.

Video Generation. We evaluate the video generation capability of EditVerse on the VBench benchmark (Zhang et al., 2024), shown in Table 8. As shown, EditVerse achieves highly competitive

Method	Add	Adjust	Extract	Replace	Remove	Background	Style	Hybrid	Action	Overall↑
MagicBrush	2.84	1.58	1.51	1.97	1.58	1.75	2.38	1.62	1.22	1.83
Instruct-P2P	2.45	1.83	1.44	2.01	1.50	1.44	3.55	1.20	1.46	1.88
AnyEdit	3.18	2.95	1.88	2.47	2.23	2.24	2.85	1.56	2.65	2.45
UltraEdit	3.44	2.81	2.13	2.96	1.45	2.83	3.76	1.91	2.98	2.70
ICEdit	3.58	3.39	1.73	3.15	2.93	3.08	3.84	2.04	3.68	3.05
Step1X-Edit	3.88	3.14	1.76	3.40	2.41	3.16	4.63	2.64	2.52	3.06
UniWorld-V1	3.82	3.64	2.27	3.47	3.24	2.99	4.21	2.96	2.74	3.26
BAGEL	3.81	3.59	1.58	3.85	3.16	3.39	4.51	2.67	4.25	3.42
EditVerse (Ours)	3.81	3.62	1.44	3.95	3.14	3.58	4.71	2.72	3.80	3.42
OmniGen2	3.57	3.06	1.77	3.74	3.20	3.57	4.81	2.52	4.68	3.44
Kontext-dev	3.83	3.65	2.27	4.45	3.17	3.98	4.55	3.35	4.29	3.71
Ovis-U1	3.99	3.73	2.66	4.38	4.15	4.05	4.86	3.43	4.68	3.97
GPT-4o-Image	4.61	4.33	2.90	4.35	3.66	4.57	4.93	3.96	4.89	4.20

Table 7: **Quantitative comparison on ImgEdit-Bench** (Ye et al., 2025a).

performance compared with a wide range of both open-source and commercial models. Notably, even though EditVerse is trained on diverse tasks beyond video generation and is built with a relatively small model size, it can still match or surpass the performance of several larger-scale models.

Models	# Params.	Total	Quality Score	Semantic Score
ModelScope	1.7B	75.75	78.05	66.54
LaVie	3B	77.08	78.78	70.31
OpenSoraPlan V1.3	-	77.23	80.14	65.62
Show-1	6B	78.93	80.42	72.98
AnimateDiff-V2	-	80.27	82.90	69.75
Gen-2	-	80.58	82.47	73.03
Pika-1.0	-	80.69	82.92	71.77
VideoCrafter-2.0	-	80.44	82.20	73.42
EditVerse (Ours)	2B	80.97	83.47	70.97
CogVideoX	5B	81.61	82.75	77.04
Kling	-	81.85	83.39	75.68
Step-Video-T2V	30B	81.83	84.46	71.28
Gen-3	-	82.32	84.11	75.17

Table 8: **Comparison with text-to-video models on the VBench** (Zhang et al., 2024). # Params. is the number of total parameters. EditVerse shows competitive performance with a small model size.

Image Generation. We evaluate the image generation capability of EditVerse using the GenEval benchmark (Ghosh et al., 2023) shown in Table 9, which is designed to comprehensively assess text-to-image models across multiple aspects of visual reasoning and compositional fidelity. Our method achieves state-of-the-art performance when compared against a wide range of both open-source and commercial systems, highlighting better semantically aligned generation.

Method	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall
LlamaGen	0.71	0.34	0.21	0.58	0.07	0.04	0.32
LDM	0.92	0.29	0.23	0.70	0.02	0.05	0.37
SDv1.5	0.97	0.38	0.35	0.76	0.04	0.06	0.43
PixArt-Alpha	0.98	0.50	0.44	0.80	0.08	0.07	0.48
SDv2.1	0.98	0.51	0.44	0.85	0.07	0.17	0.50
DALL-E 2	0.94	0.66	0.49	0.77	0.10	0.19	0.52
Emu3-Gen	0.98	0.71	0.34	0.81	0.17	0.21	0.54
SDXL	0.98	0.74	0.39	0.85	0.15	0.23	0.55
DALL-E 3	0.96	0.87	0.47	0.83	0.43	0.45	0.67
Infinity†	-	0.85	-	-	0.49	0.57	0.73
SD3-Medium	0.99	0.94	0.72	0.89	0.33	0.60	0.74
FLUX.1-dev†	0.98	0.93	0.75	0.93	0.68	0.65	0.82
EditVerse (Ours)	0.99	0.95	0.81	0.82	0.68	0.64	0.82

† uses LLM-rewritten prompts.

Table 9: **Comparison with text-to-image models on the GenEval** (Zhang et al., 2024).

A.3 Detailed Training Data

Table 10 provides a detailed overview of the entire training datasets that are used in our work, along with their respective ratio in the training process. The table is organized by task type: image editing, image generation, video editing, and video generation. For each dataset, we report the total number of samples, the ratio applied when constructing the training mixture, and a brief description highlighting the data quality, coverage, and characteristics. The training data comprises a mixture of high-quality open-source data, curated internal datasets, and filtered synthetic datasets. This combination allows us to balance scale, quality, and diversity, ultimately supporting unified training across both editing and generation tasks for images and videos.

A.4 Limitation and Future Work

While EditVerse presents a significant step toward unified image and video generation and editing, we acknowledge several limitations that open avenues for future research.

Observed Failure Cases. Despite its strong overall performance, EditVerse is not immune to failure cases including artifacts, flickering, low motion, logical flaws, wrong editing position, and blurred editing region. Figure 9 shows examples of two commonly seen failure types of EditVerse.

Figure 9: **Failure case examples of EditVerse**. (a) The model fails to add object (treasure chest) at the correct position (at the man's feet). (b) Generation of blurry artifacts within the edited region.

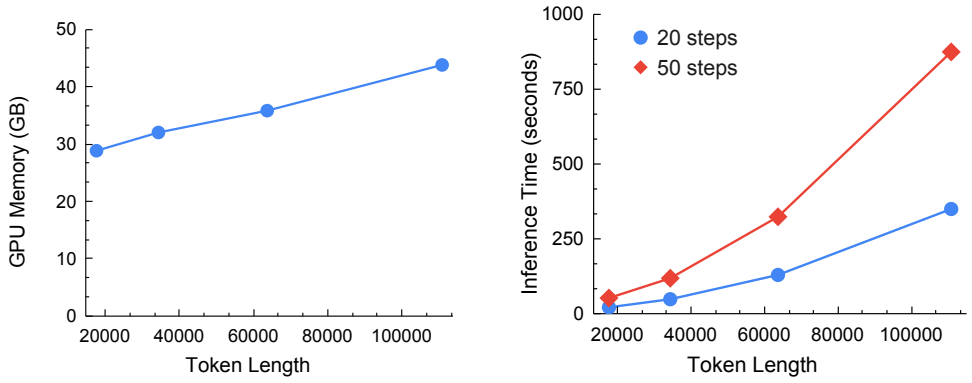
Computational Cost. Our reliance on a full self-attention mechanism across a unified one-dimensional token sequence, while powerful for in-context learning, leads to significant computational overhead. The concatenation operation results in long sequence lengths, particularly for high-resolution or long-duration videos, which translates to high FLOPs and prolonged training and inference time. Our work was not specifically designed to improve efficiency, and the reported memory usage and inference time reflect the model's raw, unoptimized performance. There are several practical ways to improve it, which we plan to explore in future work: (1) Using a higher-compression VAE. In our model, we use a VAE with relatively low compression ($8\times$ spatial downsampling), which leads to a large number of visual tokens. Recent VAEs can achieve $16\times$ spatial compression. If we replace our VAE with $16\times$ spatial compression rate models, the token length can be reduced to about one quarter, which directly lowers the attention cost. (2) Dynamic token selection. We can introduce a dynamic token selection mechanism that adaptively keeps only the most important context tokens and prunes redundant ones. This can reduce the effective sequence length for full attention, as explored in recent work FullDiT2 (He et al., 2025). (3) Distillation for faster inference. We can further apply model distillation and step distillation to reduce both the number of diffusion steps and the model size, which can noticeably speed up generation. (4) Future work could explore more efficient attention mechanisms (*e.g.*, linear attention, Mamba attention) to reduce the computational burden without compromising the model's cross-modal learning capabilities.

To assess the efficiency of our model with extended token length (as shown in Figure 10 and Table 11), we analyze the impact of token length on both efficiency and GPU memory usage. Ex-

Dataset	#Samples	#Ratio	Information
Image Editing			
MagicBrush	8,802	10	Manually annotated with real image. High-quality. 7 editing categories.
ShareGPT-4o-Image	46,489	10	Generated by GPT-4o. 14 editing categories. Most are high-quality, but some cases contain noise.
Object Removal & Addition‡	118,972	4	Manually captured photos with object-present and object-absent scenes. High-quaity.
OmniEdit*	185,500	2	Generated by task-specific models. 7 editing categories. Good-quality but contains large noise in some editing categories.
ImgEdit*	245,986	1	Generated by segmentation and inpainting. 13 editing categories. Fair quality. Need filtering.
NHR-Edit	358,463	5	Generated with a designed pipeline using internal image editing model. High-quality. 17 editing categories.
UltraEdit	500,000	1	Generated by a specially designed editing model. Fair quality. 9 editing categories.
AnyEdit*	1,244,033	1	Generated by task-specific pipelines. 25 editing categories. Fair quality. Need filtering.
GPT-Image-Edit-1.5M	1,500,000	1	Re-process OmniEdit, UltraEdit, and HQ-Edit with GPT-4o. Most are high-quality, but some cases contain noise.
Instruction-based Editing‡	1,824,969	1	An internal instruction-based image editing dataset.
Sum	**6,033,214**		
Image Generation			
BLIP3o-60k	60,000	1	Text-to-Image instruction tuning dataset distilled from GPT-4o.
LLaVA-pretrain	500,000	1	Text-to-Image data re-captioned using Qwen2-VL (from text-to-image-2M).
Text-to-Image‡	609,950	1	Internal high-quality text-to-image dataset.
LLaVA-next fine-tuning	700,000	1	Text-to-Image data generated by Flux-dev (from text-to-image-2M).
Sum	**1,869,950**		
Video Editing			
Camera Change	8,000	20	Camera change data pair generated with ReCamMaster
Style Transfer	10,327	10	Style transfer data pair generated with Step1X-Edit and VACE.
Mask Detection	15,741	5	Editing region detection with prompt "I want to [edit prompt]. Detect the region that needs to be edited". Contain object removal, object addition, and object Replacement.
Object Replacement	31,482	10	Object replacement data pair generated with VACE. Contain w/ mask version and w/o mask version in training.
CG Removal & Addition‡	38,900	2	Rendered videos with object-present and object-absent scenes.
Propagation	59,826	10	Containing editing propagation for object removal, object addition, object replacement, and style transfer.
Object Removal & Addition	67,516	10	Object removal and addition pairs generated with DiffuEraser. Contain w/ mask version and w/o mask version in training.
Señorita-2M*	55,711	2	Generated wih task-specific models. 5 editing categories. Low quality. Need filtering.
Sum	**287,503**		
Video Generation			
Depth-to-Video	182,097	2	Depth is estimated with Depth Anything v2.
Video-to-Depth	182,097	2	Depth is estimated with Depth Anything v2.
Sketch-to-Video	207,749	2	Sketch is computed with OpenCV Canny.
Video-to-Sketch	207,749	2	Sketch is computed with OpenCV Canny.
Pose-to-Video	233,068	2	Pose is estimated with RTM-Pose.
Video-to-Pose	233,068	2	Pose is estimated with RTM-Pose.
Video Inpainting	1,495,020	2	Video inpainting data pair generated with Grounded SAM 2. Contain w/ mask version and w/o mask version in training.
Text-to-Video‡	223,494	10	Internal high-quality text-to-video dataset.
Image-to-Video‡	217,038	5	Internal image-to-video dataset.
Customization	740,111	1	High-quality video customization dataset from OmniVCus.
Sum	**3,921,491**		

‡ Internal datasets.
* We filter these datasets to improve their quality.

Table 10: **Detailed Overview of the Training Datasets.** We combine high-quality open-source datasets, internal datasets, and EditVerse datasets generated following our data pipeline. This table presents the dataset name, sample counts, training ratios, and key details of each dataset.



(a) GPU Memory v.s. Token Length (b) Inference Latency v.s. Token Length

Figure 10: GPU memory and inference time scaling with token length.

Token Length	17652	34342	63662	110822
Inference Time (second/20 steps)	21	48	129	349
Inference Time (second/50 steps)	52	118	323	873
GPU Memory (MB)	29511	32768	36663	44824
TFLOPs/second	67.9	58.2	39.4	25.4

Table 11: Inference efficiency across different token lengths

periment results demonstrate that resource consumption growth remains within a manageable and predictable range. Specifically, GPU Memory usage exhibits a strong linear relationship with token length. As the sequence length was scaled from 17,652 up to 110,822 tokens, the peak memory footprint increased modestly from 29.5 GB to 44.8 GB. This predictable and relatively slow scaling rate confirms that the extended context itself does not impose an unconstrained memory ceiling. Similarly, inference time shows a systematic, controllable increase with token length.

Image Editing Performance. While our unified model demonstrates strong generalization and performs on par with many image editing models, it does not currently achieve state-of-the-art performance in image editing. Targeted optimizations, such as employing a more sophisticated data-mixing strategy or fine-tuning the model on high-quality, image-only editing datasets, could be explored to boost its performance and close the gap with specialized, state-of-the-art image editors.

Dataset Quality. Although our data curation pipeline is crucial for enabling instruction-based video editing, the resulting dataset inevitably contains noise. The editing instructions are often concise (averaging around 10 words) and may lack the detail required for highly complex or nuanced edits. Future efforts could focus on developing more advanced data generation and filtering techniques.

Generalist vs. Specialist Models. Our work highlights the potential of unified models, but it is plausible that for specific, well-defined tasks with abundant high-quality data (e.g., inpainting), a dedicated specialist model might still yield superior results. A systematic investigation into the trade-offs between our generalist framework and specialist models would be a fruitful direction for future research. This could help delineate the precise scenarios where a unified approach offers the most significant advantages and where specialized architectures remain preferable.

A.5 Image and Video Copyrights

Figure 1 videos are from `pixabay` (Pixabay, 2025), stockbusters – stock.adobe.com (the first video on the top), andreybiling – stock.adobe.com (the second video on the top), and Mara Zemgaliete – stock.adobe.com (the third video on the top). Comparison images in Figure 1 are from ImgEdit-Bench (Ye et al., 2025a). Example videos in Figure 3 are from `pixabay` (Pixabay, 2025) and black-boxguild – stock.adobe.com (the first video in "More Examples"). Example videos in Figure 4, 6, and 8 are all from `pixabay` (Pixabay, 2025).