This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

EasyV2V: A High-quality Instruction-based Video Editing Framework

Jinjie Mai[1,2*] Chaoyang Wang[2] Gordon Guocheng Qian[2] Willi Menapace[2]
Sergey Tulyakov[2] Bernard Ghanem[1] Peter Wonka[2,1] Ashkan Mirzaei[2*†]
[1] KAUST [2] Snap Inc.
[*] Core contributors [†] Project lead

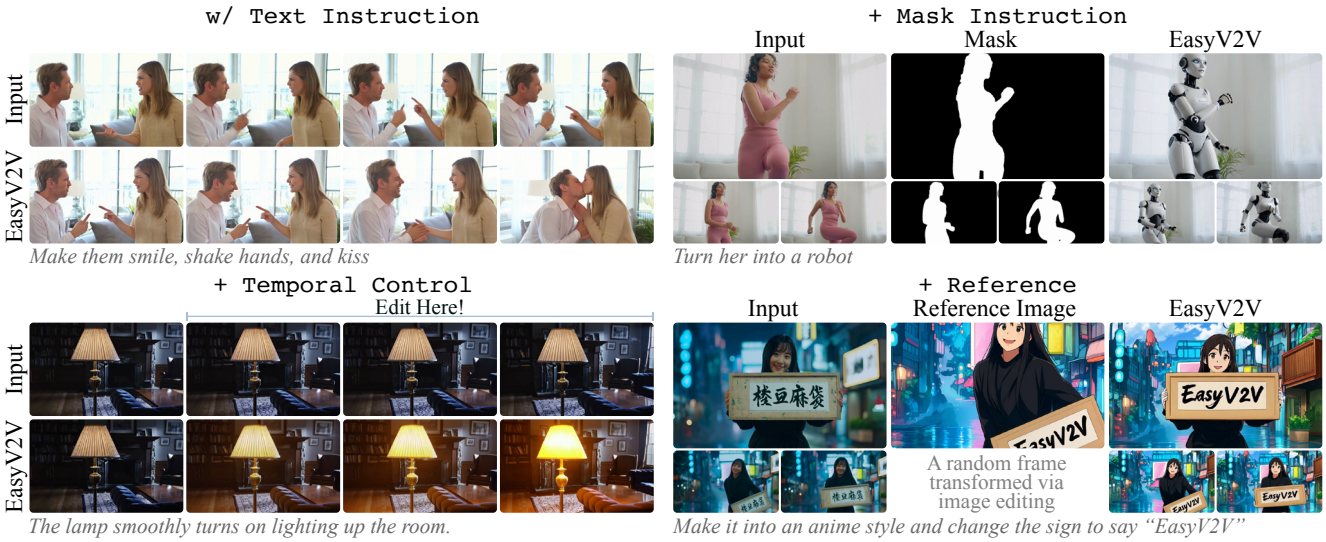


Figure 1. EasyV2V unifies data processing, architecture, and control for high-quality, instruction-based video editing with flexible inputs (text, masks, edit timing). The figure illustrates its versatility across diverse input types and editing tasks.

Abstract

While image editing has advanced rapidly, video editing remains less explored, facing challenges in consistency, control, and generalization. We study the design space of data, architecture, and control, and introduce EasyV2V*, a simple and effective framework for instruction-based video editing. On the data side, we compose existing experts with fast inverses to build diverse video pairs, lift image edit pairs into videos via single-frame supervision and pseudo pairs with shared affine motion, mine dense-captioned clips for video pairs, and add transition supervision to teach how edits unfold. On the model side, we observe that pretrained text-to-video models possess editing capability, motivating a simplified design. Simple sequence concatenation for conditioning with light LoRA fine-tuning suffices to train a strong model. For control, we unify spatiotemporal control via a single mask mechanism and support optional reference images. Overall, EasyV2V works with flexible inputs, e.g., video+text, video+mask+text, video+mask+reference+text, and achieves state-of-the-art video editing results, surpassing concurrent and commercial systems.* Project Page: https://snap-research.github.io/easy-v2v/

1. Introduction

What makes a good instruction-based video editor? We argue that three components govern performance: *data*, *architecture*, and *control*. This paper analyzes the design space of these components and distills a recipe that works great in practice. The result is a lightweight model that reaches state-of-the-art quality while accepting flexible inputs.

Training-free video editors adapt pretrained generators but are fragile and slow [10, 25]. Training-based approaches improve stability, yet many target narrow tasks such as ControlNet-style conditioning [1, 19], video inpainting [70], or reenactment [8]. General instruction-based video editors handle a wider range of edits [9, 51, 62, 71], yet still lag behind image-based counterparts in visual fidelity and control. We set out to narrow this gap. Figure 2 motivates our design philosophy: modern video models already know how to transform videos. To unlock this emerging capability with minimal adaptation, we conduct a comprehensive investigation into data curation, architectural de-



Figure 2. A pretrained text-to-video model can mimic common editing effects without finetuning. This suggests that much of the "how" of video editing already lives inside modern backbones.

sign, and instruction-control.

Data. We see three data strategies. *(A) One generalist model* renders all edits and is used to self-train an editor [9, 51, 62]. This essentially requires a single teacher model that already solves the problem in high quality. *(B) Design and train new experts* for specific edit types, then synthesize pairs at scale [71]. This yields higher per-task fidelity and better coverage of hard skills, but training and maintaining many specialists is expensive and slows iteration and adaptation to future base models. We propose a new strategy: *(C) Select existing experts* and compose them. We focus on experts with a fast inverse (e.g., edge↔video, depth↔video) and compose more complex experts from them. This makes supervision easy to obtain, and experts are readily available as off-the-shelf models. It keeps costs low and diversity high by standing on strong off-the-shelf modules; the drawback is heterogeneous artifacts across experts, which we mitigate through filtering and by favoring experts with reliable inverses. Concretely, we combine off-the-shelf video/image experts for stylization, local edits, insertion/removal, and human animation; we also convert image edit pairs into supervision via two routes: single-frame training and pseudo video-to-video (V2V) pairs created by applying the same smooth camera transforms to source and edited images. Lifting image-to-image (I2I) data increases scale and instruction variety; its limitation is weak motion supervision, and shared camera trajectories reintroduce temporal structure without changing semantics. We further leverage *video continuation*: we derive V2V pairs from densely captioned text-to-video (T2V) datasets by sampling input clips outside each captioned interval and target clips within it, while converting the caption into an instruction using an LLM. Continuation data teaches action and transition edits that are scarce in typical V2V corpora, at the cost of more careful caption-to-instruction normalization. We collect and create an extensive set of V2V pairs, the most comprehensive among published works, and conduct a detailed study of how editing capability emerges from each training source.

Architecture. Using a pretrained video backbone, we study two strategies to inject the source video: *channel concatenation* and *sequence concatenation*. Channel concatenation is faster in practice because it uses fewer tokens, but we show that sequence concatenation consistently yields higher edit quality. The trade-off is efficiency versus separation: channel concatenation keeps context short but entangles source and target signals; sequence concatenation costs more tokens but preserves clean roles for each stream, improving instruction following and local detail. Our final design adds small, zero-init patch-embedding routes for the source video and an edit mask used for spatiotemporal control of the edit extent, reuses the frozen video VAE for all modalities, and fine-tunes only with LoRA [15]. Full finetuning can help when massive, heterogeneous data are available, but it risks catastrophic forgetting and is costly to scale. At our training scale, LoRA transfers faster, reduces overfitting, and preserves pretrained knowledge while supporting later backbone swaps; its main downside is a slight headroom loss when unlimited data and compute are present. We inject the mask by token addition, and concatenate tokens from the source and optional reference image along the sequence. This preserves pretraining benefits, keeps token budgets tight (by not introducing new tokens for the mask), and makes the model easily portable to future backbones. Addition for masks is simple and fast; while it carries less capacity than a dedicated mask-token stream, we find it sufficient for precise region and schedule control without context bloat. We use an optional reference frame at training and test time to leverage strong image editors when available. References boost specificity and style adherence when present; randomly dropping the reference during training keeps the model robust when they are absent or noisy.

Flexible control. Prior work explores control by skeletons, segmentation, depth, and masks. A key signal is still missing: *when* the edit happens. Users often want an edit to appear gradually (e.g., "set the house on fire starting at 1.5s, then let the flames grow gradually"). We unify spatial and temporal control with a single mask video. Pixels mark *where* to edit; frames mark *when* to edit, and how the effect evolves. Alternatives include keyframe prompts or token schedules, which are flexible but harder to author and to align with motion. A single mask video is direct, differentiable, and composes well with text and optional references. The cost is requiring a mask sequence, which we keep lightweight and editable.

Combining these design choices results in a unified editor, **EasyV2V**, which supports flexible input combinations, including `video + text`, `video + mask + text`, and `video + mask + reference + text` (see Fig. 1). Despite its simplicity, our recipe consistently improves edit quality, motion, and instruction following over recently published methods. In summary, our contributions are:

- A clarified design space for instruction-based video editing and a consistent strategy across data, architecture, and control that achieves state-of-the-art results.
- A reusable data engine built from composable experts with trivial inverses, lifted high-quality image edits, and video continuation, with per-expert/per-source ablations.
- A lightweight architecture that minimally modifies a pretrained video backbone: zero-init patch-embeddings for source and mask, frozen VAE reuse, and LoRA finetuning, plus optional reference frames.
- *Temporal control* as a first-class signal, unified with spatial control via a single mask video that schedules when edits start and how they evolve.

2. Related work

Instruction-based visual editing datasets. With the success of image and video generative models trained on large-scale text-to-image and text-to-video datasets, recent work has focused on developing datasets for instruction-based image and video editing. Early approaches attempted to build large-scale paired I2I datasets, though with low success rates, requiring extensive automatic [4, 17] or manual [63] filtering. Later efforts improved success rates by leveraging task-specific models [13, 50, 55, 61, 68], leading to highly capable image editing systems [28, 45]. These models subsequently enabled the generation of higher-quality paired I2I datasets [49]. Creating paired V2V datasets is inherently more challenging, as it requires editing multiple frames while maintaining temporal coherence and faithfully applying the intended modification. Early work [9] used LLMs and Prompt-to-Prompt [14] to synthesize video pairs, but results were limited by artifacts from the underlying editing method. Another dataset [16] provided video–object-mask pairs with corresponding captions but omitted edited videos. The current trend is to generate synthetic datasets using one previous general video editing model [9, 51, 62] or a collection of more specialized video editing models [2, 71]. These efforts still lag behind image editing datasets in quality and diversity. We discuss strategies to mitigate this gap and provide extensive analysis comparing the effects of different approaches.

Instruction-based visual editing. Building on the success of diffusion models for image [12, 18, 27, 38, 40, 45] and video [5, 24, 44, 46, 56] generation, early visual editing methods used pretrained diffusion models in a training-free manner for image editing [7, 14, 20, 26, 29, 32, 47, 60], typically manipulating attention maps and latent spaces, or by noise inversion. These approaches generally produce low-quality outputs, have slow inference, and exhibit low success rates. Subsequent work showed that even relatively low-quality paired datasets can outperform training-free methods [4, 41]. Such models concatenate input image latents with noisy latents along the channel dimension and fine-tune a pretrained image generator on synthetic paired I2I data produced by training-free approaches. Another line of research performs concatenation along the sequence dimension, improving quality at the cost of efficiency [6, 22, 53, 67]. Similar directions adopt LLM-style architectures for unified or token-based editing [11, 69]. Instruction-based video editing has been explored far less. Early attempts include training-free video editing pipelines [10, 25]. Some works adapt image generative models [33, 42], while others train on low-quality synthetic video pairs and are limited by training data quality [9, 14, 62, 66]. Additional methods include propagating edits from the first frame to subsequent frames [30] or designing architectures with task-specific conditioning for multiple editing tasks [59]. Another line of work explores task-specific editing, such as ControlNet-style [64] video generation [19] or pose/face conditioned human video synthesis [8]. Two concurrent frameworks recently appeared on arXiv, focusing on patch-wise concatenation [43] and LLM-style architectures [21]. In contrast, our method systematically studies different data sources for training V2V models, identifies those most critical for performance, and enables more controllable video generation through optional reference images and spatiotemporal edit control.

3. Method

Overview. We build on a modern video generative backbone [46] and introduce lightweight conditioning modules for video editing. A pivotal design choice is how to inject the source video condition. While one simple approach is to concatenate the source video latent and the noisy latent along the channel dimension [43], requiring only a modification of the initial patch embedding layer, we found this method struggles to learn edits efficiently. We therefore adopt a sequence-wise concatenation strategy, appending the source video tokens to the noisy latent token sequence. This approach provides a more robust and effective conditioning mechanism, as validated in our ablations (Tab. 3).

Concretely, we add separate patch-embedding layers for each control signal (source video and mask) and support an optional reference image at training and inference. After encoding and patch embedding, condition signals are injected either by element-wise addition (for the edit mask) or by sequence concatenation (for source video and optional reference image). To preserve the generative priors of the T2V backbone and ensure stable training, we adopt a parameter-efficient finetuning strategy. We freeze the original backbone weights and optimize only the new patch-embedding layers and low-rank adaptation (LoRA) weights added to the DiT's attention layers. We found this approach essential, as full-model finetuning easily led to training instability and source video inconsistency.

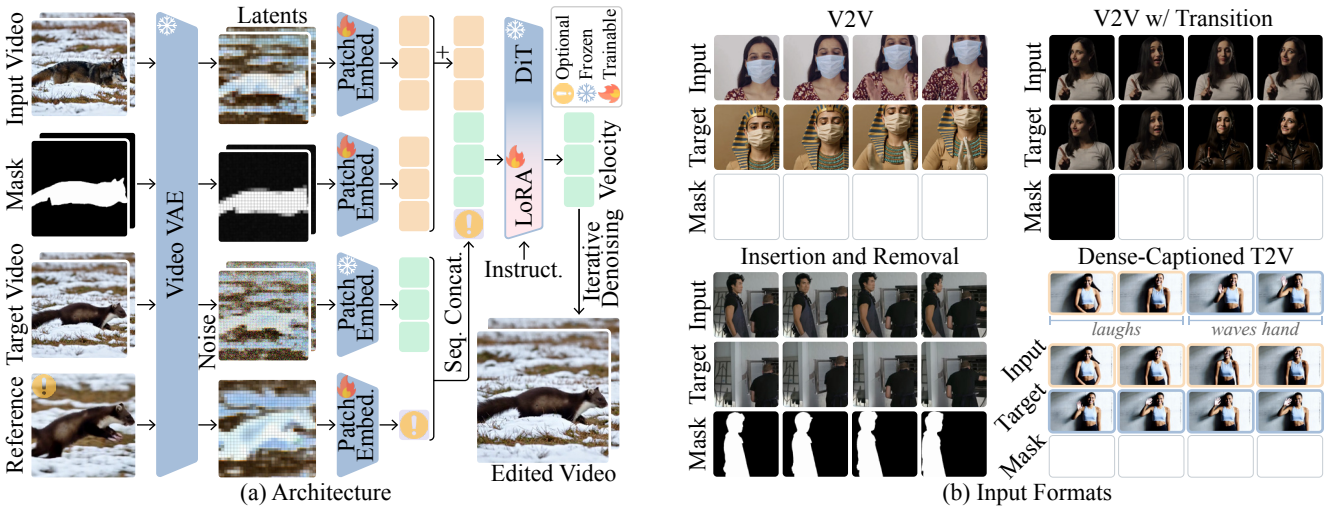


Figure 3. (a) Overview of our video editing architecture. The frozen video VAE encodes control signals. Mask tokens are added to the input video tokens and concatenated with the noisy latent tokens and optional reference image. The DiT is trained using LoRA. (b) Input formats used for different dataset types during training. When object masks are available, the spatiotemporal mask input equals the input mask. Otherwise, the mask indicates transitions from the unedited to the edited video.

Input video conditioning. As shown in Fig. 3, we adopt a simple yet effective strategy to condition on the input video. Given a source video and a target video, each of shape $V \in \mathbb{R}^{NCHW}$, we encode them with a video VAE into latents $Z_{\text{src}}, Z_{\text{tgt}} \in \mathbb{R}^{nchw}$. We use *separate* patch-embedding layers for Z_{src} and Z_{tgt}. After applying edit-mask conditioning (below), their tokens are concatenated along the sequence dimension. We place Z_{src} first, followed by Z_{tgt}, so the model can learn in-context video editing behavior that resembles video continuation, which is supported by our dense-caption video data.

Edit mask conditioning. The mask video $M \in \mathbb{R}^{NCHW}$ is a binary indicator of where and when to edit the source video. It can specify (1) pixel-wise regions for inpainting or removal and/or (2) frame-wise intervals to control when an edit occurs and for how long, enabling both spatial and temporal controllability. We process the encoded mask latents Z_{msk} with a dedicated patch-embedding layer and inject them by addition into the source video tokens. This addition-based injection is a deliberate choice for computational efficiency. Because the mask is a low-frequency signal, its information can be effectively fused without appending it to the DiT's input sequence. At inference time, if no mask video is provided, we default to blank masks, treating the task as a canonical, instruction-only video edit.

Reference image conditioning. We optionally provide a reference image during both training and inference. The reference can be sampled from the target video during training. During inference, it can be produced by an external image-editing model by applying it to a frame from the source video or provided by the user. Because references can be imperfect (e.g., spurious zooms from Qwen-Image-Edit [35]), we apply random crops/rotations and randomly drop the reference during training so the model is robust to its absence or noise at test time. When present, the reference is encoded with the video VAE and $\texttt{Patch_embedding}_{\text{ref}}$, and its tokens are concatenated at the end of the sequence. This preserves a fixed token distance between Z_{src} and Z_{tgt} while positioning Z_{ref} closer to Z_{tgt} for stronger guidance, leveraging off-the-shelf image-editing priors without sacrificing inference flexibility.

Table 1. Summary of datasets used for training. We collect $\sim$ 4.3M open-sourced and licensed data and curate $\sim$ 3.4M data, resulting in a total of $\sim$ 8M high-quality data for training.

	Dataset	Type	Mask	Reference	# Pairs
	GPT-Edit-1.5M [49]	I2I	✗	✗	$\sim$ 1.5M
	Ditto-1M [2]	V2V	✗	✗	$\sim$ 1M
	Señorita-2M [71]	V2V	✗	✗	$\sim$ 1.8M
Ours	Image Editing	I2I	✗	✗	$\sim$ 2M
Ours	Human Animate	V2V	✗	✓	$\sim$ 60K
Ours	Object Removal / Insertion	V2V	✓	✗	$\sim$ 110K
Ours	Actor Transmutation	V2V	✗	✓	$\sim$ 4K
Ours	Video Stylization	V2V	✗	✓	$\sim$ 90K
Ours	Controllable Video Generation	V2V	✗	✗	$\sim$ 1.1M
Ours	Human Action	T2V	✗	✓	$\sim$ 150K

4. Data pipeline

We propose a data generation framework that utilizes various expert pipelines for generating paired V2V datasets. Our approach yields significantly stronger results than only using a single general editing pipeline. Figure 4 shows samples of each of our new datasets. Next, we detail the experts.

Human animation. Previous V2V datasets [71] lack expert annotations ensuring consistent human poses and facial expressions between input and edited videos. To overcome

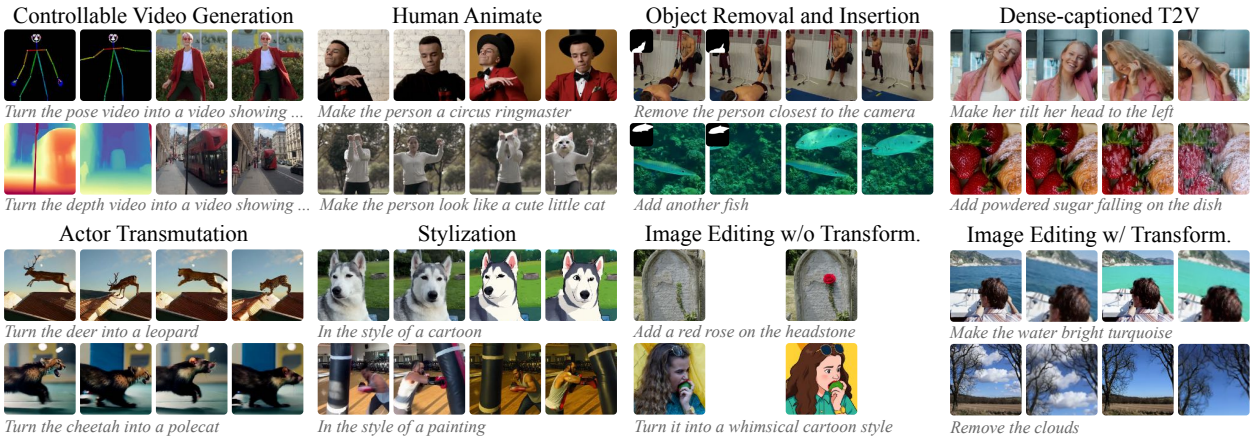


Figure 4. Overview of our datasets. In each example, the left images are inputs and the right images are outputs.

this, we use Wan Animate [8] to generate paired V2V data where the output video is conditioned on (1) the subject's pose in the input video for pose consistency, (2) cropped facial regions for expression consistency, and (3) a reference image produced by an image editing model that modifies the first frame to alter the subject or scene. A large language model (LLM) generates diverse edit instructions involving actor swaps, clothing and style changes, and the addition or removal of accessories.

Object removal and insertion. We construct paired videos by removing selected objects from original clips. First, we apply an open-set object detection model [65] to the first frame of each video to detect candidate objects and generate corresponding bounding boxes and tags. Next, an LLM refines the tags by removing adjectives (e.g., color, material) and scenery-related labels (e.g., road, sky). For each video, we randomly sample a fixed number of objects, with sampling probability proportional to the bounding box area to prioritize salient objects. For each selected object, we use a video segmentation model [37] to obtain per-frame segmentation masks. A video inpainting model [70] then generates a new version of the video with the target object removed. Human annotators review the results and discard videos with noticeable inpainting artifacts. Finally, we provide the input and output videos, along with the object category and input video captions, to a vision-language model (VLM) [3] and prompt it to generate a detailed instructional caption describing the edit. By replacing the source and target videos and corresponding VLM-based changes to the instruction, this dataset is used for object insertion.

Actor transmutation. While the human animate dataset already includes high-quality examples of replacing one humanoid with another, a general editing model should also support actor transmutation for other categories such as animals. To achieve this, we leverage a zero-shot image editing model [26] and build a zero-shot V2V pipeline on top of a pretrained video generative model [46]. We use this zero-shot V2V model to extend editing types beyond humanoid-to-humanoid, including (1) quadruped-to-quadruped, (2) bipedal-to-bipedal, and (3) avian-to-avian transformations. We observe that the zero-shot editing model [26] performs noticeably better when the input video is also generated by the same video generative model. Consequently, for each video pair, we sample two actors from the same category, along with an action and a scene, and generate a video of "*<actor1> performing <action> in <scene>*". We then apply our zero-shot V2V pipeline to produce the edited version described as "*<actor2> performing <action> in <scene>*". The lists of actors, actions, and scenes for each category are generated by an LLM.

Video stylization. A general-purpose V2V model should be capable of applying global style changes to input videos. To create suitable datasets for this task, we extract edge maps from input videos to preserve structure and motion while removing appearance and style information. We then use an image editing model [28] to apply an LLM generated style transfer instruction to the first frame of the input video. This edited reference image, along with the edge video, are used as control signals to generate the stylized output video [1]. The style transfer instructions include art movements and styles, lighting conditions, aesthetics, photographic and cinematic styles, weather conditions, traditional arts and patterns, and color palettes and tones.

Controllable video generation. Video generation using control signals such as depth and edge maps has been widely studied [1, 19]. The datasets for training such models consist of video pairs that are simple and inexpensive to obtain, making them suitable as supporting data for training V2V models. These datasets enhance controllable generation capabilities and potentially promote emergent skills through data diversification. We include controllable video generation pairs in our training data, using control signals such as depth, HED edge, Canny edge, optical flow, human poses, noisy videos, and grayscale videos.

V2V transition data. To supervise *how* an edit unfolds over time and enable temporal mask control, we synthe-

size transition effects on top of paired videos V^{src} and V^{tgt}. Given an edit onset t_i, we form a training target $V' = [\, V^{\text{src}}_{t_0:t_i},\, V^{\text{tgt}}_{t_i:t_N} \,]$ and derive a frame-wise mask that activates the edit after t_i. We then apply transition operators like linear blending to ensure a natural transition effect centered at t_i.

I2I data High-quality instructional V2V datasets are limited, while image editing benefits from mature models and large-scale data. To bridge this gap, we train our instructional V2V editor with I2I edit pairs collected from open-source datasets [17, 49, 50, 68] and additional pairs synthesized from VLM-curated image–caption corpora. From each caption, an LLM produces diverse instruction-style edit prompts; instruction-following image editors [28, 35] generate the edited images. A VLM-based filter retains only successful, high-quality edits, yielding instruction-aligned I2I supervision without manual annotation.

I2I $\rightarrow$ V2V via affine transformations. Treating an I2I pair as a single-frame video is insufficient because it lacks temporal and motion cues. To address this, we convert each I2I pair into a *pseudo* video pair using a shared 2D affine camera trajectory. Specifically, we sample smooth sequences of small rotations, zooms, and translations by interpolating between a neutral and a randomly drawn target pose, with adjustments to prevent out-of-frame regions. The same trajectory is applied to both the source and target images, producing temporally consistent videos that differ only by the intended edit. This preserves the I2I supervision signal while introducing realistic temporal structure.

Dense-captioned text-to-video data Existing V2V datasets cover object manipulation, stylization, and animation but lack edits like changing *human action*. As a result, instruction-guided V2V models often fail to modify actions effectively. Dense-captioned text-to-video datasets [52], however, provide diverse action descriptions at scale, enabling action-centric supervision. To curate T2V data for video editing, given a video V and a caption c localized to a temporal window $[t_i, t_j]$ (e.g., *"he sits down"*), we slice V to form a *source* clip $V_{t_{i-N}:t_i}$ (frames preceding the caption) and a *target* clip $V_{t_i:t_{i+N}}$ (frames containing the caption), where N is the frame count used for training. We discard windows that are too long ($t_j - t_i \gg N$), too short, or contain scene cuts. The caption c is then converted into an imperative instruction c' using an LLM [54] (e.g., *"make him sit down"*). This yields canonical triples for training V2V editors: `(source video, target video, instruction)`. Our approach scales efficiently to large captioned video corpora while emphasizing action edits. Qualitative results (Figure 1) reveal that curated action edits enable EasyV2V to perform accurate and complex human action edits.

5. Experiments

Implementation details. We adopt the pretrained video generation model, Wan-2.2-TI2V-5B [46], as our base model for LoRA adaption and Wan-2.2-VAE with a spatiotemporal compression ratio of $4 \times 16 \times 16$. We perform training and report our results based on a spatiotemporal resolution of $81 \times 832 \times 480$ following Wan-2.1-FunControl. We also provide high-quality results fine-tuned on $81 \times 1280 \times 704$ in the supplement. For LoRA training, we set the rank to 256 and adopt a constant learning rate of $1e^{-4}$ with AdamW [31] optimizer. We conduct our training on 32 NVIDIA H100s for our EasyV2V. All newly introduced parameters are zero-initialized. We perform random reference image dropout and video transition augmentation each with 50% probability. A complete list of all training data used is provided in Tab. 1.

Benchmarks and metrics. We evaluate our method on the latest EditVerseBench [21] which has 20 edit types. We evaluate on categories covered by our training data—excluding unsupported tasks such as camera-pose changes—resulting in 160 videos across 16 edit types. Following the benchmark's protocol, we conduct a comprehensive quantitative evaluation using these metrics: *Frame and video text alignment.* We measure alignment at both frame and video levels using image-text [36] and video-text [48] encoders by computing cosine similarity between (i) each frame and the target prompt and (ii) a joint embedding of uniformly sampled frames and the prompt. *Preference score.* We report a preference-tuned image-text PickScore [23] that correlates with human aesthetic judgments, applied between frames and the target prompt. *VLM quality assessment.* We use OpenAI GPT-4o [45] to evaluate three sampled frame pairs on prompt following, edit quality, and background consistency (0-3 each), providing justifications and a total score of 9. In our analysis, we observe that the VLM score aligns most closely with human qualitative assessment, so we designate the VLM score as our primary metric for comparison.

Baseline comparison. We perform a comprehensive quantitative comparison against a diverse set of baseline methods in Table 2. With our primary VLM score of $7.73/9$, without any guidance, we outperform not only the previously best published method but also concurrent work and a commercial solution. When a reference image is given, we can achieve better visual-text alignment performance.

We provide a qualitative comparison with the baselines in Fig. 5. Since EditVerse [21] has not released its code, we take their gallery videos from their webpage. InsViE-1M [51] supports only videos with short horizons and often fails to edit properly, producing severe visual artifacts. Señorita-2M [71] depends on the image editor for the first frame and, even with successfully edited first frames, shows

Table 2. Benchmark comparison across methods on EditVerse Benchmark. Higher is better for all metrics (↑).

Method	VLM evaluation Editing Quality ↑	Video Quality Pick Score ↑	Text Alignment ↑ Frame	Text Alignment ↑ Video
Attention Manipulation (Training-free)				
TokenFlow [34]	5.02	19.59	25.10	22.49
STDF [57]	4.20	19.32	24.74	22.09
Instruction-Guided (w/ End-to-End Training)				
Señorita-2M [71] (w/ Ref., Qwen-Image-Edit)	6.45	20.26	26.51	23.24
InsViE-1M [51] (w/ Ref)	4.36	19.25	25.06	21.28
InsV2V [9]	4.95	19.33	24.98	22.74
Ours (w/o Ref.)	7.73	20.36	27.59	24.46
Ours (w/ Ref., Qwen-Image-Edit)	7.36	20.54	27.50	24.31
Ours (w/ Ref., Flux-Kontext)	7.53	20.61	28.10	25.13
Closed-Source Commercial Models				
Runway Aleph [39]	7.48	20.56	27.96	24.68
Concurrent unpublished work				
Lucy Edit [43]	5.96	19.64	26.03	23.37
EditVerse [21] (code unavailable)	7.64	20.33	27.70	25.37

motion mismatch and artifacts after the first frame. Lucy Edit Dev [43] supports limited categories of edit types and frequently exhibits motion mismatches. Compared to these methods, our model achieves higher-quality outputs and better instruction following. For example, in the first row, EditVerse fails to produce a "heavy fog" across the "whole video." In the second, the tree branch between her fingers remains. In the third, the output lacks "visible pen lines." In the last, EditVerse also removes the background trees. Please refer to our supplement for visualizations.

5.1. Ablation study

Image Editing. As shown in Figure 6, we evaluated EasyV2V on a recent image-editing benchmark [58], interpreting each image as a single-frame video with a resolution of $1 \times 832 \times 480$. Notably, although EasyV2V was not specifically designed for image editing, EasyV2V outperformed EditVerse by a margin of **0.54**. The model also demonstrated strong results on both **action** and **hybrid** editing tasks, underscoring the effectiveness of our unified data pipeline that jointly leverages image-editing datasets and video–caption datasets featuring human actions.

Architecture. We ablate the architecture choice for input video conditioning and compare full model finetuning with LoRA. We evaluate the VLM score for each protocol trained with 20K and 40K steps in Tab. 3. We show that full model training tends to overfit, while LoRA tuning quickly transforms a T2V model into a V2V model. And *"EmbedAdd."* refers to adopting our patch embedding addition strategy for both mask video latent and control video latent. Technically, this is similar to channel-wise concatenation of source video latent and noisy target latent except for the bias term. However, our results show that this remains suboptimal compared to sequence-wise concatenation of control latents. We also show that LoRA tuning via patch embedding addition performs worse than sequence concatenation.

Using I2I datasets. Since generating V2V editing datasets is much more expensive than creating I2I editing datasets, and because strong I2I models and high-quality open-source datasets already exist, we argue that V2V editing models should also be trained on I2I data. A simple approach is to treat image editing samples as single-frame videos, but this lacks motion. To narrow the domain gap, we apply a consistent sequence of affine transformations to the input and edited images to simulate pseudo V2V editing pairs. We perform this ablation study by sampling an equal number of I2I and V2V editing pairs and train different models for different dataset compositions. We test performance on a benchmark containing 100 videos from edit types supported by both datasets. As shown in Table 4, adding affine transformations improves downstream V2V editing performance, and jointly training on I2I datasets outperforms training solely on V2V datasets.

Table 3. Ablation on architecture choice. *"Full"* denotes full model training, and *"EmbedAdd."* refers to summing control and target latents after patch embedding.

Method	VLM score @ 20K ↑	VLM score @ 40K ↑
Full w/ EmbedAdd.	4.67	4.57
Full w/ SeqCat.	3.66	3.94
LoRA w/ EmbedAdd. (Ours)	6.11	6.29
LoRA w/ SeqCat. (Ours)	7.05	7.47

Table 4. Ablation study on using I2I data. *Single Image* treats I2I pairs as single-frame videos, while *Affine Image* applies affine transformations to I2I data to create V2V pairs.

Training Datasets			VLM Eval.	Video Quality
Single Image	Affine Image	Video Edit	Edit Quality	Pick Score
✓	✗	✗	5.52	19.49
✓	✓	✗	6.24	19.67
✗	✗	✓	6.69	19.90
✓	✓	✓	**6.86**	19.94

Using V2V and dense-captioned datasets. To examine the effect of data curated by our pipeline, we train different models on various V2V datasets using the same number of training steps. We then evaluate performance on a mini benchmark with 10 videos for each edit type. As shown in Table 5, training on each of our proposed V2V datasets significantly improves performance on certain edit types, which validates the effectiveness of our proposed data pipeline. The only exception is our *Human Animate* dataset, which is outperformed by our *Actor Transmutation* dataset according to the VLM score. Although both tasks are similar, the latter includes more diverse subjects. However, the Human Animate dataset remains useful for preserving human identities across edits and maintaining consistent facial expressions.

6. Conclusion

We introduced *EasyV2V*, a lightweight, instruction-based video editor that achieves state-of-the- performance. First, we propose robust data engines that curate diverse training data, notably by "lifting" static image edits into

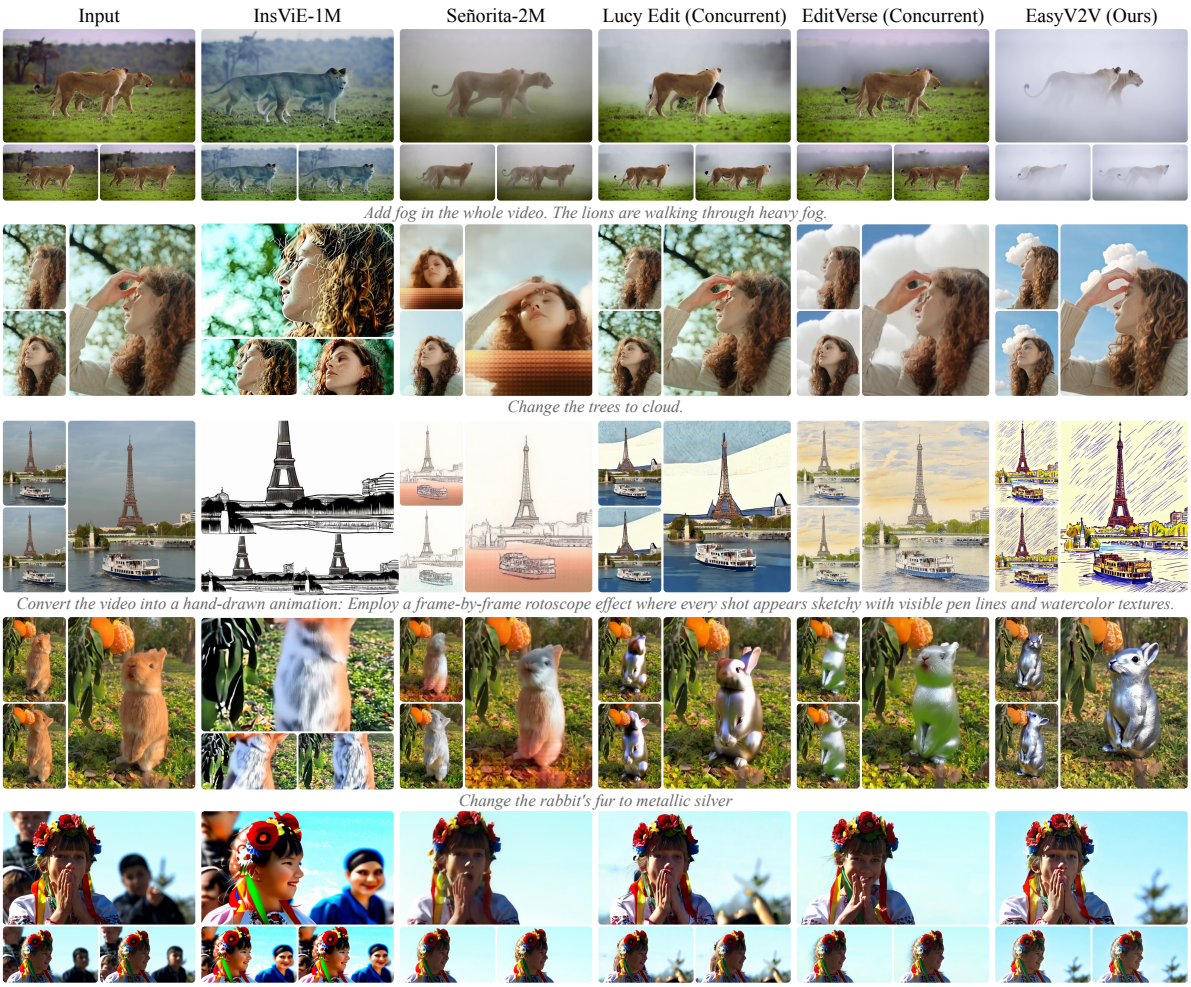


Figure 5. Qualitative comparison of our video editing results with the baselines.

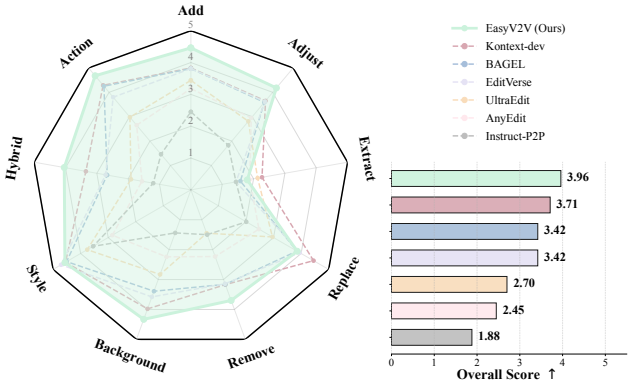


Figure 6. Category-wise image-editing performance on ImgEdit Bench [58].

dynamic pseudo-video pairs using shared affine motion, supplemented with video continuation supervision from text2video data. Second, we propose a parameter-efficient architecture which effectively conditions on the source video, enabling unified spatiotemporal control from a single mask with optional reference images. This combination of a novel data strategy and a minimal-tuning architecture provides a strong and effective recipe for high-quality video editing with extensive controllability.

Table 5. Ablation study on the effectiveness of our proposed video datasets using the VLM score as the metric.

Edit types	Training on Datasets							
	Ditto-1M [2]	Señorita-2M [71]	Stylization	Human Animate	Controllable Video	Flow Edit	Inpainting	Dense Caption
Stylization	7.13	4.97	**7.97**	5.33	5.30	5.77	4.63	5.20
Animation	3.58	3.88	3.65	7.20	3.80	**7.48**	3.13	4.18
Change object	3 .97	3.33	2.40	4.43	3.20	**5.13**	3.13	4.27
Control video	4.09	5.47	4.42	5.20	**6.13**	3.46	4.00	5.04
Actor transmutation	2.40	4.37	2.53	6.23	3.90	**8.30**	2.83	5.00
Edit w/ Mask	1.03	3.40	2.73	3.17	3.03	2.43	**4.63**	1.10
Change human action	2.17	4.97	4.50	5.03	3.53	6.50	2.50	**6.87**

Limitations. While EasyV2V is robust on a wide range of edit types, common to other concurrent diffusion-based video models, inference takes around one minute and precludes its use in real-time applications. This framework can be naturally extended to support more advanced control abilities. For example, adding geometric cinematographic camera pose controls is a compelling direction to further broaden our creative applications.

Acknowledgement. The authors thank Runjia Li and Haozhe Liu for helpful discussion, Maya Goldenberg for help with visualization, and Maxwell Jones for data processing. Jinjie Mai conducted this work during his internship at Snap. Part of this work was also supported by funding from the King Abdullah University of Science and Technology (KAUST) - Center of Excellence for Generative AI under award number 5940 and the SDAIA-KAUST Center of Excellence in Data Science and Artificial Intelligence.

References

[1] VideoX-Fun: A more flexible framework that can generate videos at any resolution and creates videos from images. https://github.com/aigc-apps/VideoX-Fun, 2025. 1, 5

[2] Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. *arXiv*, 2025. 3, 4, 8

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. 5

[4] Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. *CVPR*, 2023. 3

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[6] 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*, 2025. 3

[7] 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 *ICCV*, 2023. 3

[8] Gang Cheng, Xin Gao, Li Hu, Siqi Hu, Mingyang Huang, Chaonan Ji, Ju Li, Dechao Meng, Jinwei Qi, Penchong Qiao, Zhen Shen, Yafei Song, Ke Sun, Linrui Tian, Feng Wang, Guangyuan Wang, Qi Wang, Zhongjian Wang, Jiayu Xiao, Sheng Xu, Bang Zhang, Peng Zhang, Xindi Zhang, Zhe Zhang, Jingren Zhou, and Lian Zhuo. Wan-Animate: Unified character animation and replacement with holistic replication, 2025. 1, 3, 5

[9] Jiaxin Cheng, Tianjun Xiao, and Tong He. Consistent video-to-video transfer using synthetic dataset. *ICLR*, 2024. 1, 2, 3, 7

[10] Paul Couairon, Clément Rambour, Jean-Emmanuel HAUGEARD, and Nicolas THOME. VidEdit: Zero-shot and spatially aware text-driven video editing. *TMLR*, 2024. 1, 3

[11] Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, and Feng Li. Emerging properties in unified multimodal pretraining: The BAGEL model. *arXiv*, 2025. 3

[12] Yu Gao, Lixue Gong, Qiushan Guo, Xiaoxia Hou, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xuanda Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Xin Xia, Xuefeng Xiao, Zhonghua Zhai, Xinyu Zhang, Qi Zhang, Yuwei Zhang, Shijia Zhao, Jianchao Yang, and Weilin Huang. Seedream 3.0 technical report, 2025. 3

[13] Yuying Ge, Sijie Zhao, Chen Li, Yixiao Ge, and Ying Shan. SEED-Data-Edit technical report: A hybrid dataset for instructional image editing. *arXiv*, 2024. 3

[14] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv*, 2022. 3

[15] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 2

[16] 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, 2025. 3

[17] 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*, 2024. 3, 6

[18] Imagen-Team-Google. Imagen 3, 2024. 3

[19] Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. VACE: All-in-one video creation and editing. In *ICCV*, 2025. 1, 3, 5

[20] Xuan Ju, Ailing Zeng, Yuxuan Bian, Shaoteng Liu, and Qiang Xu. PnP inversion: Boosting diffusion-based editing with 3 lines of code. *ICLR*, 2024. 3

[21] Xuan Ju, Tianyu Wang, Yuqian Zhou, He Zhang, Qing Liu, Nanxuan Zhao, Zhifei Zhang, Yijun Li, Yuanhao Cai, Shaoteng Liu, Daniil Pakhomov, Zhe Lin, Soo Ye Kim, and Qiang Xu. EditVerse: Unifying image and video editing and generation with In-Context learning, 2025. 3, 6, 7

[22] 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, 2025. 3

[23] 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. 6

[24] 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*, 2024. 3

[25] Max Ku, Cong Wei, Weiming Ren, Harry Yang, and Wenhu Chen. AnyV2V: A tuning-free framework for any video-to-video editing tasks. *TMLR*, 2024. 1, 3

[26] Vladimir Kulikov, Matan Kleiner, Inbar Huberman-Spiegelglas, and Tomer Michaeli. FlowEdit: Inversion-free text-based editing using pre-trained flow models. *ICCV*, 2025. 3, 5

[27] Black Forest Labs. FLUX. https://github.com/black-forest-labs/flux, 2024. 3

[28] 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, 2025. 3, 5, 6

[29] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-P2P: Video editing with cross-attention control. *CVPR*, 2024. 3

[30] Shaoteng Liu, Tianyu Wang, Jui-Hsien Wang, Qing Liu, Zhifei Zhang, Joon-Young Lee, Yijun Li, Bei Yu, Zhe Lin, Soo Ye Kim, and Jiaya Jia. Generative video propagation. *CVPR*, 2025. 3

[31] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[32] 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. *ICCV*, 2023. 3

[33] Bosheng Qin, Juncheng Li, Siliang Tang, Tat-Seng Chua, and Yueting Zhuang. InstructVid2Vid: Controllable video editing with natural language instructions. In *ICME*, 2024. 3

[34] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K. Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. In *CVPR*, 2025. 7

[35] Qwen Team. Qwen-Image: A unified foundation model for image generation and editing. *arXiv*, 2025. 4, 6

[36] 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*, pages 8748–8763. PmLR, 2021. 6

[37] 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. *arXiv*, 2024. 5

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[39] Runway. Introducing runway aleph. https://runwayml.com/research/introducing-runway-aleph. 7

[40] Team Seedream, :, Yunpeng Chen, Yu Gao, Lixue Gong, Meng Guo, Qiushan Guo, Zhiyao Guo, Xiaoxia Hou, Weilin Huang, Yixuan Huang, Xiaowen Jian, Huafeng Kuang, Zhichao Lai, Fanshi Li, Liang Li, Xiaochen Lian, Chao Liao, Liyang Liu, Wei Liu, Yanzuo Lu, Zhengxiong Luo, Tongtong Ou, Guang Shi, Yichun Shi, Shiqi Sun, Yu Tian, Zhi Tian, Peng Wang, Rui Wang, Xun Wang, Ye Wang, Guofeng Wu, Jie Wu, Wenxu Wu, Yonghui Wu, Xin Xia, Xuefeng Xiao, Shuang Xu, Xin Yan, Ceyuan Yang, Jianchao Yang, Zhonghua Zhai, Chenlin Zhang, Heng Zhang, Qi Zhang, Xinyu Zhang, Yuwei Zhang, Shijia Zhao, Wenliang Zhao, and Wenjia Zhu. Seedream 4.0: Toward next-generation multimodal image generation, 2025. 3

[41] Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *CVPR*, 2023. 3

[42] Uriel Singer, Amit Zohar, Yuval Kirstain, Shelly Sheynin, Adam Polyak, Devi Parikh, and Yaniv Taigman. EVE: Video editing via factorized diffusion distillation. *ECCV*, 2024. 3

[43] Decart Team. Lucy edit: Open-weight text-guided video editing. https://d2drjpuinn46lb.cloudfront.net/Lucy_Edit__High_Fidelity_Text_Guided_Video_Editing.pdf, 2025. 3, 7

[44] Genmo Team. Mochi 1. https://github.com/genmoai/models, 2024. 3

[45] OpenAI Team. GPT-4o system card, 2024. 3, 6

[46] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models, 2025. 3, 5, 6

[47] 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*, 2023. 3

[48] 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*, 2023. 6

[49] Yuhan Wang, Siwei Yang, Bingchen Zhao, Letian Zhang, Qing Liu, Yuyin Zhou, and Cihang Xie. GPT-IMAGE-EDIT-1.5M: A million-scale, GPT-generated image dataset, 2025. 3, 4, 6

[50] Cong Wei, Zheyang Xiong, Weiming Ren, Xinrun Du, Ge Zhang, and Wenhu Chen. OmniEdit: Building image editing

generalist models through specialist supervision. In *ICLR*, 2025. 3, 6

[51] Yuhui Wu, Liyi Chen, Ruibin Li, Shihao Wang, Chenxi Xie, and Lei Zhang. InsViE-1M: Effective instruction-based video editing with elaborate dataset construction. *ICCV*, 2025. 1, 2, 3, 6, 7

[52] Ziyi Wu, Aliaksandr Siarohin, Willi Menapace, Ivan Skorokhodov, Yuwei Fang, Varnith Chordia, Igor Gilitschenski, and Sergey Tulyakov. Mind the time: Temporally-controlled multi-event video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 23989–24000, 2025. 6

[53] 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 *CVPR*, 2025. 3

[54] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv*, 2025. 6

[55] 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*, 2024. 3

[56] 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. *ICLR*, 2025. 3

[57] Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. Space-time diffusion features for zero-shot text-driven motion transfer. In *CVPR*, 2024. 7

[58] 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*, 2025. 7, 8

[59] 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, 2025. 3

[60] Jaehong Yoon, Shoubin Yu, and Mohit Bansal. RACCooN: A versatile instructional video editing framework with auto-generated narratives. *EMNLP*, 2025. 3

[61] 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. In *CVPR*, 2025. 3

[62] Shoubin Yu, Difan Liu, Ziqiao Ma, Yicong Hong, Yang Zhou, Hao Tan, Joyce Chai, and Mohit Bansal. VEGGIE: Instructional editing and reasoning video concepts with grounded generation. *ICCV*, 2025. 1, 2, 3

[63] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. MagicBrush: A manually annotated dataset for instruction-guided image editing. In *NeurIPS*, 2023. 3

[64] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3

[65] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. Recognize anything: A strong image tagging model. *arXiv*, 2023. 5

[66] Zhenghao Zhang, Zuozhuo Dai, Long Qin, and Weizhi Wang. EffiVED: Efficient video editing via text-instruction diffusion models, 2024. 3

[67] 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 transformers. *NeurIPS*, 2025. 3

[68] Haozhe Zhao, Xiaojian 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. In *NeurIPS*, 2024. 3, 6

[69] 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, 2024. 3

[70] Bojia Zi, Weixuan Peng, Xianbiao Qi, Jianan Wang, Shihao Zhao, Rong Xiao, and Kam-Fai Wong. MiniMax-Remover: Taming bad noise helps video object removal. *arXiv*, 2025. 1, 5

[71] Bojia Zi, Penghui Ruan, Marco Chen, Xianbiao Qi, Shaozhe Hao, Shihao Zhao, Youze Huang, Bin Liang, Rong Xiao, and Kam-Fai Wong. Señorita-2M: A high-quality instruction-based dataset for general video editing by video specialists. In *NeurIPS*, 2025. 1, 2, 3, 4, 6, 7, 8