Motion-Conditioned Image Animation for Video Editing

Wilson Yan[1,2,†], **Andrew Brown**[1], **Pieter Abbeel**[2], **Rohit Girdhar**[1], **Samaneh Azadi**[1]

[1]GenAI, Meta, [2]UC Berkeley
[†]Work done during an internship at Meta

We introduce MoCA, a **Mo**tion-**C**onditioned Image **A**nimation approach for video editing. It leverages a simple decomposition of the video editing problem into image editing followed by motion-conditioned image animation. Furthermore, given the lack of robust evaluation datasets for video editing, we introduce a new benchmark that measures edit capability across a wide variety of tasks, such as object replacement, background changes, style changes, and motion edits. We present a comprehensive human evaluation of the latest video editing methods along with MoCA, on our proposed benchmark. MoCA establishes a new state-of-the-art, demonstrating greater human preference win-rate, and outperforming notable recent approaches including Dreamix (63%), MasaCtrl (75%), and Tune-A-Video (72%), with especially significant improvements for motion edits.

Date: November 30, 2023
Correspondence: Wilson Yan at wilson1.yan@berkeley.edu
Website: facebookresearch.github.io/MoCA

1 Introduction

Recent advancements in image and video generation models have seen tremendous progress, with existing models able to synthesize highly complex images [26, 27, 28, 30, 6] or videos [37, 31, 2, 15, 12] given textual descriptions. Outside of generating purely novel content, these models have shown to be powerful tools in achieving advanced image and video editing capabilities for downstream content creation.

Given a source video, a caption of the source video, and an editing textual prompt, a video editing method should produce a new video that is aligned with the provided editing prompt while retaining faithfulness to all other non-edited characteristics of the original source video. Video edit types can be broadly split into two main categories of spatial and temporal edits. Spatial edits generally consist of image-based edits extended to video, such as editing a video in the style of Van Gogh, inserting an object into the scene, or changing the background. Due to the added temporal dimension in video, we can also change the underlying motion of the object, such as making a panda play in a pile of ribbons, or replacing apricots in a video with apples and making them fall off a tree (see Figure 1).

Current methods in video editing focus more on spatial editing problems while ignoring the motion editing problem. Proposed methods leverage pre-trained text-to-image or video models for editing by further fine-tuning with conditioning on auxiliary information such as depth maps or edge maps [8, 44], fine-tuning for each edit example [40, 21, 29, 16, 10], or exploiting the diffusion process to restrict the generated edits to share similar features and structures with the source content [13, 17, 4, 36, 19, 42, 11]. Notably, most proposed methods either in image or video editing are generally specialized only to a subset of editing tasks and do not perform well on others. For example, methods to utilize depth or edge maps of the video [8, 44] find it more difficult to perform motion edits due to adherence to the original video structure. As such, it becomes important to assess video editing capabilities across a wide range of different edits in order to better understand their advantages and disadvantages.

In this paper, we present two key contributions for video editing. First, we introduce a simple yet strong approach for video editing, **Mo**tion-**C**onditioned Image **A**nimation (MoCA), that decomposes the problem into image editing and image animation. We first use the existing image editing methods to edit the first video

Figure 1 MoCA is able to generate a diverse range of edits, such as object replacement, style changes, and motion edits. The frames in the top row in each example represent the source video while the bottom ones show the edited frames by MoCA. The source and editing prompts are shown above each example.

frame, then produce an edited video using a motion-conditioned image animation model. We use an optical flow representation of the source video using a pretrained RAFT [34] model as the motion conditioning to retain the original motion characteristics of the source video. In the video edits consisting of a motion edit, we drop out this motion conditioning. Through our extensive experiments and human evaluations, we show that this simple baseline outperforms the state-of-the-art video editing models across a wide range of edit types.

Secondly, we introduce a dataset of 250+ video edits that comprehensively covers a wide range of video editing types. We combine existing datasets for video editing, and introduce our own subset of curated videos from YouTube-8M [1] with a stronger emphasis in including motion-based edits due to a general lacking of such examples in current public video editing datasets [41]. Using our combined dataset, we comprehensively benchmark prior video editing methods along a range of pre-categorized edits types, such as style, background, object, and motion-based edits via human evaluation and automatic metrics. Additionally, we perform an analysis of the alignment between the automatic metrics for measuring video editing quality and human judgement.

2 Related Work

By the remarkable progress in text conditional image and video generation models, text-guided image and video editing have emerged as key editing tools that enable average users and artists to create new content easily from existing photos or videos. In this section, we will discuss the existing works on diffusion-based text-driven image and video editing and their applicability to various manipulation tasks.

2.1 Text-driven Image Editing

Prior works have proposed a variety of methods for text-conditioned image editing. One family of such image editing methods focus on using diffusion models, and produce image edits by altering the backward diffusion process. SDEdit [19] is a simple image editing approach that applies various diffusion noise levels to an input

source image, and produces image edits by sampling back out through the diffusion process conditioned on an edit prompt. Plug-and-Play [36] samples edited videos initialized from the DDIM [32] inversion, with selected visual features copied between the source and generated images during diffusion sampling. Prompt-to-Prompt [13] (P2P) enables general edit types (style changes, object replacement, changing texture) through replacing self and cross attention of the generated image with the attention maps of the source image during diffusion sampling. Null-Text Inversion [20] extends P2P to enable better editing performance when editing real images through optimizing null text embeddings to allow for more faithful reconstruction of the source image during diffusion sample.

Another class of image editing techniques that allow for more global changes in visual features are built on ControlNet [44] or T2I-Adapters [22], where pretrained text-to-image models are augmented and finetuned to incorporate conditioning information, such as depth maps or contour maps computed from edge detection algorithms.

Most prior image editing methods are generally constrained structurally, and have a more difficult time producing image edits with large pose changes, such as editing an image of a bird to spread its wings. MasaCtrl [4] achieves this through mutual self-attention, where select self-attention layers in the diffusion networks attend to the keys and values of the corresponding layers of the source image during the diffusion process. Larger pose changes are enabled by only enabling mutual self-attention replacement during later diffusion timesteps. Imagic [16] similarly achieves image edits with larger pose changes through a text embedding optimization and model fine-tuning process.

Lastly, the Instruct-Pix2Pix [3] family of models propose to treat the image editing problem as a supervised learning problem. Core work around these model requires collecting supervised data as pairs of (text editing instructions and images before/after the edit), and fine-tuning a pre-trained text-to-image model on the collected data.

In this paper, we focus on video editing, as it provides a more challenging task in accomplishing complex edits over both space and time.

2.2 Text-driven Video Editing

Video editing similarly can be deployed for various manipulation tasks including style transfer, object or scene manipulations, and motion editing. However, these manipulation tasks are more challenging in videos since the generated content should be consistent across frames. Most of the existing works in video editing focus more on the first two manipulation types while ignoring the motion editing problem, as they generally propose video editing methods that leverage pretrained text-to-image models. Pix2Video [5] and FateZero [24] both propose different variants of extending self-attention with cross-frame attention. TokenFlow [11] Rerender a Video [42], and CoDef [23] perform video edits through image editing techniques and propagating edited features temporally using estimated motions by computing temporal inter-frame correspondences [11], optical flow estimation and warping [42], or estimating canonical images and temporal deformation fields using optical flow of the source video [23], respectively.

Methods such as Gen-1 [8], VideoComposer [38], and ControlVideo [45] train text-to-video models with additional conditioning inputs such as depth maps or motion vectors to allow for controllability of general structure in the resulting video edits.

Tune-a-Video [40] and Dreamix [21] both propose fine-tuning a pre-trained diffusion model for each source video, where Tune-a-Video finetunes a pretrained text-to-image model and Dreamix fine-tunes a pre-trained text–to-video model. For both methods, edits are produced using the fine-tuned video model to sample back out conditioned on given the edit prompts.

Most prior works tend to target specific types of edits and evaluate on their own constructed sets of edit prompts. As such, the benefits of each method across different kinds of edits are less clear, and motivates us to propose a benchmark centered around a more rigorous analysis of the pros and cons of each method. In addition, we propose our own video editing method that leverages existing text conditional image editing models to edit the first frame of a source video and extrapolate its future frames via a motion conditional video generation diffusion model to enable alignment of the edit with the source video and the editing prompt.

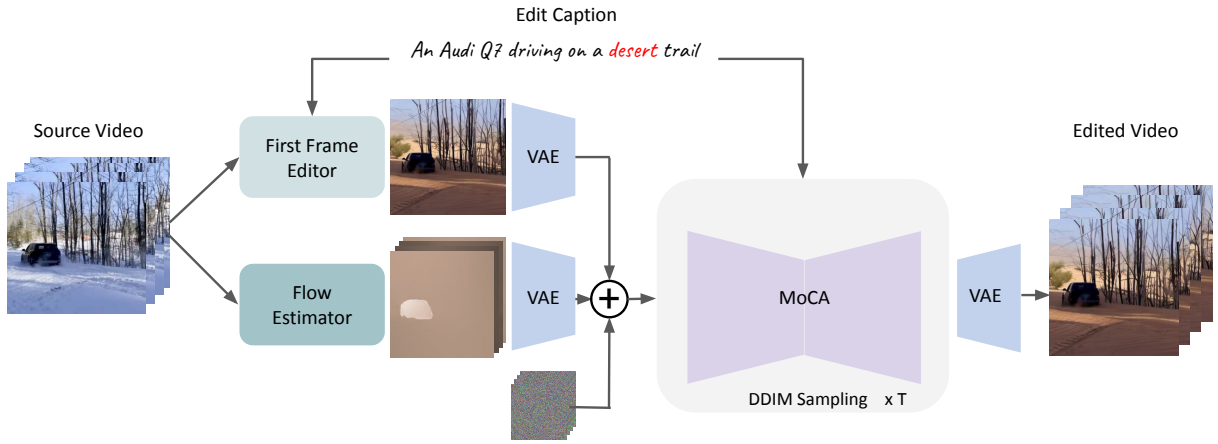

Figure 2 An overview of MoCA. Given a source video, we compute its optical flow, and apply image editing techniques on the first frame. To produce the resulting video edit, we sample our model conditioned on motion, the edited first frame, and the edit caption. For motion-based edits, we dropout the optical flow conditioning.

3 Background

Conditional latent diffusion models. Diffusion models learn to generate samples from a training distribution by reversing a gradual noising process. At the sampling time starting from a Gaussian noise, the model generates less noisy samples in T time-steps where each time-step, t, corresponds with a specific noise level [7]. In latent diffusion models for each input image x, this noising and denoising process is applied on the latent space, $z = \mathcal{E}(x)$, of a pretrained variational autoencoder with encoder $\mathcal{E}$, resulting in more efficient training and sampling steps. In a text-conditional latent diffusion model, text features are extracted from a pre-trained language model, and then fed into the latent diffusion U-Net blocks via cross-attention modules. In addition to the text conditioning, the image or video generation models can be also conditioned on an additional input image by concatenating its features with the noisy latent features at each time-step, z_t, and adding extra input channels to the first convolutional layer of the U-Net [3, 12, 43]. The network will be trained to predict the noise added to the noisy latent features given image and text conditioning inputs, respectively.

Classifier free guidance. Classifier-free guidance was proposed in [14] and is widely used to improve the fidelity and diversity of the generated samples and their correspondence with the conditioning input in a diffusion model. During training, the diffusion model is trained jointly in a conditional and unconditional setting where the conditioning input is set to NULL with a specific frequency. At inference, the generated samples are guided to be more faithful to the conditioning input while being further away from the NULL input with a guidance scale $s >= 1$.

4 MoCA

Inspired by the success of image conditioning for video generation [12, 43] and text-driven image editing methods [13, 35, 3, 29, 44], we introduce a simple yet strong baseline for text-driven video editing that can be deployed in a wide range of video editing applications. We decompose the video editing problem into image editing, and motion-conditioned image animation. We choose to adopt this decomposition, as (1) image editing has shown large success, with the growing availability of more capable image editors that are able to edit a variety of complex editing prompts, and (2) the recent success of image animation methods for video generation to model temporal dynamics in videos [12, 43]. As a simple approach, we could first leverage image editing techniques to edit the first frame given the editing prompt, and then use an image animation model to predict the rest of the frames. However, this typically results in videos that diverge from the motion of the original source video, which is important to retain especially for edits that only target to change the style, or background of the video. Therefore, for these cases, we propose to additionally condition on a motion representation of the video. When editing video motion, we dropout conditioning on the motion and predict

future frames conditioned only on the edited image. An overview of MoCA is presented in Figure 2.

To achieve this, we train a latent diffusion video generation model conditioned on (1) a text prompt, (2) an image as the first frame of each video, and (3) an optical flow representing the motion in the video. We use a variational autoencoder (VAE) pre-trained on an internal image database to encode an input video with shape $T \times 3 \times H \times W$ frame-wise to a tensor of shape $T \times C \times H' \times W'$. We learn the diffusion process on this latent space and finally decode the latent to the video pixel space via its decoder. We use a pre-trained Flan-T5-XXL [18] text encoder to extract text features, c_T, and feed them into the model via cross-attention modules. At inference time, we edit the first frame of the source video via off-the-shelf text-based image editing models [13, 19] and use that as the image conditioning input while using the optical flow estimation of the source video as the motion conditioning. In the case where motion edits are desired, we dropout the motion-conditioning.

Image Editing We leverage a diverse range of existing image editing methods, and find certain editing methods to be useful for specific editing types. We have found Prompt-to-Prompt [13] effective for style, background, and object replacement edits, and SDEdit [44] for multi-spatial edits that contain larger feature and pose changes. For motion-only edits, we keep the original source frame.

Image conditioning. Inspired by Emu-video [12], to condition the video generation model on the first frame as an input, c_I, we encode the first frame using the same auto-encoder, $\mathcal{E}(c_I)$, repeat it T times to have the same shape as the video latent features, and then concatenate it channel-wise with the noisy latent features at each time-step.

Motion conditioning. To condition the video generation model on the optical flow motion representation, c_M, we convert the optical flow into an RGB video, encode each frame using the same auto-encoder, $\mathcal{E}(\text{toRGB}(c_M))$ and concatenate it channel-wise with the noisy latent and image conditioning features at each time-step. Since conversion to RGB re-normalizes the frames, we additionally compute an average flow magnitude term, and condition the video model on it. This conditioning is performed similar to the diffusion time-step conditioning.

Classifier free guidance for three conditionings. Similar to [3, 12], we leverage classifier-free guidance with respect to all three conditioning inputs to control faithfulness to each of the inputs at inference time. During training, we randomly set each of the individual conditioning inputs and their pairwise combination to NULL for 10% of the examples, respectively. To train for both motion-conditioned image animation, and image animation only, we train with 50% dropout on motion conditioning. We have therefore, three guidance scales for the text, image, and motion conditioning inputs that we adjust based on the editing application. During inference, we use the following conditioning order to compute the classifier guidance, where v_θ is the output from the U-Net, and $\tilde{v}_\theta$ is used to denoise the input image:

$$\begin{aligned}\tilde{v}_\theta(z_t, c_M, c_T, c_I) = v_\theta(z_t, \varnothing, \varnothing, \varnothing) \\ + s_I \cdot (v_\theta(z_t, \varnothing, \varnothing, c_I) - v_\theta(z_t, \varnothing, \varnothing, \varnothing)) \\ + s_T \cdot (v_\theta(z_t, \varnothing, c_T, c_I) - v_\theta(z_t, \varnothing, \varnothing, c_I)) \\ + s_M \cdot (v_\theta(z_t, c_M, c_T, c_I) - v_\theta(z_t, \varnothing, c_T, c_I))\end{aligned}$$

5 Experiments

5.1 Implementation Details

Our video generation model is built from a text-to-image U-Net based latent diffusion model pre-trained on our internal database of 400M (image, text) pairs. Similar to earlier works in video generation [31, 2, 12], we expand this 2D U-Net to video generation by adding temporal modules consisting of 1D temporal convolution layers and 1D temporal attention blocks after each spatial convolution and attention block, respectively.

We initialize all the spatial parameters from the pre-trained text-to-image model and fine-tune all the temporal and spatial layers of our video prediction model, with 1.4B trainable parameters, on an internal licensed dataset consisting of 34M pairs of video-text samples. We sample random 256×256 2-second clips from each

video using a frame rate of 4 frames per second with the first frame as the conditioning image. Videos are encoded via the pre-trained VAE to the $4 \times 8 \times 32 \times 32$ resolution. We additionally use RAFT [34] to extract an estimated optical flow representation for each video during training. Similar to [12], we train our model using zero terminal-SNR and v-prediction, on a batch size of 512 split across 32 A100 GPUs. During inference we use 64 DDIM steps for sampling.

5.2 Evaluation Dataset

We introduce a dataset of 271 edit tasks, defined as a set of (source video, edit prompt) pairs designed to comprehensively evaluate and benchmark video editing capabilities of current methods. Our dataset consists of a combination of existing video editing datasets, as well as our own curated subset:

- **LOVEU-TGVE Dataset** [41]: comprises of 35 source videos, with 4 different manipulation tasks proposed for each video (140 edits total). We filtered out videos with human faces and hands.
- **Dreamix Dataset** [21]: consists of 14 videos downloaded from the dreamix paper website, with edits primarily focusing on scene changes with motion.
- **Our Custom Dataset**: we curate an additional 37 videos from YouTube-8m [1], focused on including a diverse range of motion edits as well as a composition of scene and motion edits (117 edits total).

	LOVEU TGVE	Dreamix Dataset	Our Dataset
Style	35	1	11
Background	35	1	7
Object	35	7	14
Motion	0	2	68
Multi-Spatial	35	0	0
Multi-Motion	0	0	17
Total	140	14	117
# Unique Videos	35	9	37
Avg Edits Per Video	4	1.56	3.16

Table 1 Details for each individual dataset in the VideoEdit benchmark, as well the total distribution of edits types. Our custom curated dataset focuses more heavily on motion editing due to the general lack of motion edits available in the combined LOVU-TGVE and Dreamix datasets.

We group each edit task into one of following edit types, some of which are explored in [41, 21]:

- *Style*: changes in the composition of the video, such as making the video reflect a specific artistic style (crayon drawing, oil painting, impressionism),
- *Object*: adding or replacing objects in the scene, such as replacing a lion with a zebra, or placing a hat on a person's head,
- *Background*: changes in the background scene of the video, such as replacing a snowy mountain background with a desert,
- *Motion*: changes in the motion of an entity compared to the source video, such as making a monkey jump, or a car turn in a different direction,
- *Multi-Spatial*: a combination of style, object, and background changes in the video,
- *Multi-Motion*: a combination of style, object, and background changes in addition to a motion change.

Table 1 shows a break-down of the number of videos and edits of each type for each dataset.

5.3 Baselines

We compare against a set of SOTA baselines to comprehensively target different families of video editing models.

- **TokenFlow** is a tuning-free text-to-image based editing model [11]. It leverages a pre-trained text-to-image diffusion model to edit videos by computing and propagating spatial edits across temporal correspondences found in the original source videos. We use the public repo [1] to run this baseline.

[1] https://github.com/omerbt/TokenFlow

Figure 3 Comparison of our method against baselines for a given video editing task. Our method is able to accurately edit both the spatial and temporal properties of the source video.

- **Tune-a-Video** is a fine-tuning text-to-image based editing model [40]. For each edit task, Tune-a-Video extends a pre-trained text-to-image diffusion model to the temporal domain and fine-tunes the weights on a given source video. During inference, the edit result is generated through a DDIM initialized sampling using the fine-tuned model conditioned on the edit prompt. We use the public repo [2] to run this baseline.
- **Dreamix** is a fine-tuning text-to-video based editing model [21]. It fine-tunes a text-to-video model for *each source video*, and edits are generated by sampling at different levels of noise strengths, conditioned on the edit prompt. Due to the lack of public code and models, we perform Dreamix fine-tuning using our own internal text-to-video model (as indicated by the tilde). Our text-to-video model follows the same training parameters and model architecture (LDM) as MoCA, only without the initial concatenating for image and motion conditioning.
- **MasaCtrl** is a tuning-free text-to-video based editing model [4]. Originally presented in the text-to-image domain, MasaCtrl enables more robust structural (e.g. pose) changes in image edits compared to the prior methods. It replaces self-attention layers with mutual self-attention to query correlated local structures and textures from a source image. We extend MasaCtrl to the video domain using our text-to-video model (same as that of the Dreamix baseline). Importantly here, we found it crucial to only apply mutual self-attention layers in the spatial, and not temporal transformer layers of our network.
- **Gen-1** is a tuning-free video editing model [8]. Gen-1 video-to-video model generates a video conditioned on a given edit prompt and a depth map of the source video. We use the public web interface in generating edit results.
- **VideoComposer** is a tuning-free text-to-video generation model [38]. VideoComposer is a motion-conditioned video model that can generate videos conditioned on a single image, and desired motion

[2] https://github.com/showlab/Tune-A-Video

	Style	Background	Object	Motion	Multi-Spatial	Multi-Motion	Total
Dreamix [21]	53%	53%	63%	81%	49%	65%	63%
Gen-1 [8]	66%	40%	80%	99%	57%	90%	74%
MasaCtrl [4]	74%	72%	80%	76%	71%	65%	75%
Tune-a-Video [40]	53%	67%	70%	86%	74%	85%	72%
TokenFlow [11]	70%	77%	63%	83%	83%	85%	76%
VideoComposer [38]	78%	76%	92%	84%	74%	85%	82%

Table 2 Human evaluation results for preference of our method over each of the baselines. User ratings generally show greater preference for our method, with the exception of Gen-1 for background edits, and Dreamix for multi-spatial edits.

extracted from the source video. We use the public repo [3] to run this baseline. When generating edits, we condition VideoComposer on the same edited image as given to our method.

All baselines are run in their native resolutions and frame rates, and spatio-temporally down-sampled to 256 resolution, 4 frames per second for fair comparisons to our method. For each method and (video, edit prompt) pair, we perform a hyper-parameter sweep to generate 10 candidate edits, and use human evaluators to select the best edit. The hyper-parameter sweeps vary for each method, with some over one to three hyper-parameters while still restricted to the same max 10 candidate generations.

5.4 Human Evaluation

Methodology We use crowd-sourced workers from Amazon Mechanical Turk (AMT) for our human evaluations. For each task given the source video and the editing prompt, evaluators are asked to perform a binary selection on their preferred video edit out of two given edits, one from our proposed method. Inspired by the JUICE metric introduced in [12], they are also required to choose the reasoning for their selection as either a better consistency with the source video or a higher alignment with the editing prompt or both. The same task is given to five different evaluators, and the overall preferred video edit is selected through a majority vote. We evaluate paired comparisons between our method and all given baselines, and report the final metrics as the percentage of video edit examples for which our method is preferred.

Results Table 2 shows human evaluation results comparing our method against each of the baselines, partitioned by edit type. A value of 50% means that both methods perform equally, with values greater than 50% showing a stronger human preference towards the edits produced by our method. Evaluators significantly preferred our method over all other baselines. When examining results split by edit type, human raters showed a larger gap in preference for our method's motion edits, with a more narrow gap on spatial edits. Gen-1 notably shows capabilities in background edits, as seen by the 40% preference for our method, and 60% for Gen-1. We hypothesize that since background edits require larger visual feature deviations from the original source video, Gen-1 performs well on this task by generating high quality videos. However, it is less preferred on other video edit types since it does not preserve the visual features and style of the source video due to only conditioning on depth maps.

Dreamix shows similarly competitive results in the spatial edits, but struggles more on the motion edits. We found that Dreamix has a tendency to overfit to the motion of the source video, even when adjusting the number of fine-tuning steps. MasaCtrl shows strong motion-editing abilities, but struggles more with spatial edits, especially those with larger feature changes, due to its reliance on mutual self-attention on visual features of the source content. Tune-a-Video has strong spatial editing capabilities, but generates less temporally coherent motion, and performs poorly on the motion edits. Similarly, TokenFlow has a reasonable performance in spatial edits, but struggles with motion edits due to its reliance on a pre-trained text-to-image model. Lastly, VideoComposer generally struggles to remain faithful to the image conditioning input, or produces less temporally coherent motions.

[3] https://github.com/damo-vilab/videocomposer

		Style	Background	Object	Motion	Multi-Spatial	Multi-Motion	Total
ImageCLIP	M_{sim}	45%	43%	47%	63%	44%	50%	51%
	M_{dir}	80%	72%	74%	53%	81%	66%	68%
	M_{geo}	78%	71%	70%	50%	82%	67%	67%
VideoCLIP	M_{sim}	42%	44%	53%	74%	40%	51%	55%
	M_{dir}	78%	74%	77%	59%	76%	73%	72%
	M_{geo}	79%	72%	77%	56%	71%	78%	69%

Table 3 Classification accuracy of each CLIP-based automatic metric, considering binary human decisions comparing MoCA edits against different baselines as the ground truth labels. Note that random guessing achieves roughly 50% accuracy. M_{dir} and M_{geo}, standing for CLIP text-video directional and geometric similarity scores, show relatively high accuracy (up to 80%) on spatial-based edits, such as style, background, object, and multi-spatial. However, both methods have a much harder time selecting the correct motion-based edits.

Lastly, Figure 4 shows the distribution of factors selected in which human raters preferred our method over baselines. In general human raters preferred MoCA due to its stronger alignment with the edit prompt. VideoComposer shows a slightly different distribution, of which we hypothesize may be due to the fact that it would generally produce videos with high text-video alignment, but may deviate far from the source or edited image, thus the higher distribution in selecting consistency with the source video as a deciding factor. An example video edit by all models is shown in Figure 3.

5.5 Automatic Evaluation

In addition to presenting human evaluation results, we also investigate automatic evaluation metrics using pre-trained Video and Image CLIP models [25, 39]. We perform an analysis over several CLIP-based metrics to measure video editing quality.

CLIP video similarity score. Given a source video V_{source} and an edit result V_{edit}, we first measure faithfulness between the source video and the resulting edited video:

$$M_{sim} = \mathcal{E}_V(V_{source}) \cdot \mathcal{E}_V(V_{edit})$$

where $\mathcal{E}_V$ is the VideoCLIP encoder.

CLIP text-video directional similarity score. Given a source prompt T_{source} and an edit prompt T_{edit}, we measure the edit quality using a CLIP text-video directional similarity metric [9, 3], defined as:

$$\Delta T = \mathcal{E}_T(T_{edit}) - \mathcal{E}_T(T_{source})$$
$$\Delta V = \mathcal{E}_V(V_{edit}) - \mathcal{E}_V(V_{source})$$
$$M_{dir} = \frac{\Delta V \cdot \Delta T}{\|\Delta V\|_2 \, \|\Delta T\|_2}$$

where $\mathcal{E}_T$ is the VideoCLIP text encoder. This metric measures the consistency of the change between the two videos (in CLIP space) with the change between the two prompts.

CLIP text-video geometric similarity score. Lastly, since both metrics are important in measuring the overall editing quality [3], we consider an additional metric consisting of the geometric average of both metrics which would be penalized if one of the metrics is too low.

$$M_{geo} = \sqrt{M_{sim} * M_{dir}}$$

We similarly compute these scores using a pre-trained image-CLIP encoder as the average of per-frame similarity scores.

Correlation between automatic and human scores. In order to measure the alignment of each evaluation metric to the human judgements, we treat the paired edit selection task as a binary classification problem,

where for each pair of given video edits, we compute the ground-truth label as the majority vote among human raters. Table 3 shows the classification results for each of the automatic evaluation metrics using both Image and Video-based CLIP models.We use the original L/14 Image CLIP model, and a VideoCLIP model introduced in [39]. Note that random guessing would achieve roughly 50% accuracy.

Both encoder models show similar trends across all metrics, where even the highest measured overall accuracy (72%) for M_{dir} using VideoCLIP is still far off from perfectly aligning with human judgement. When split by edit type, metrics computed for spatial-type edits (style, background, object, multi-spatial) are most aligned with human raters (up to 80% accurate), whereas motion-based edits are more difficult to automatically evaluate (56% for motion-only edits). We hypothesize that this may be due to both models having only elementary understanding of motion, and being more biased towards spatial features of videos.

For further comparisons of our method and baselines, we use the two metrics with highest observed correlations, M_{dir} and M_{geo}. Results are shown in Table 4, where our method outperforms all baseline methods. Table 7 in the Supplementary shows a more detailed breakdown of evaluation results by edit type. However, in general, we note that due to the relatively low correlation between automatic metrics and human ratings, human judgements are more reliable in these evaluations.

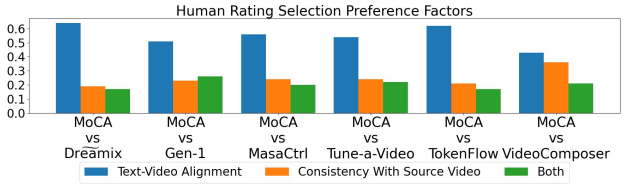

Figure 4 Percentage of each reason selected when human evaluators prefer MoCA edits to each of the baselines. The reasons for picking one model over another on each video edit could be either its better alignment with the edit prompt, higher consistency with the source video, or both. Generally, human raters preferred our method in terms of better alignment with the desired edit prompt.

Method	$\mathbf{M_{dir}}(\uparrow)$	$\mathbf{M_{geo}}(\uparrow)$
MoCA	**0.145**	**0.301**
Dreamix [21]	0.107	0.252
Gen-1 [8]	0.111	0.254
MasaCtrl [4]	0.090	0.231
Tune-a-Video [40]	0.116	0.265
TokenFlow [11]	0.098	0.235
VideoComposer [38]	0.128	0.278

Table 4 Automatic scores evaluating the editing quality of each model. We compute the VideoCILP-based M_{dir} and M_{geo} scores as the CLIP text-video directional and geometric similarity scores, respectively, averaged across all edit tasks for each baseline. Our method shows higher editing capabilities compared to the baseline methods.

Figure 5 MoCA edits for "A boat sailing on the moon" with and without motion conditioning. Using motion conditioning allows the model to more faithfully follow the boat's movement in the original source video. Without motion conditioning, the model tends to generate more random movement directions, such as moving backwards.

	Style	Object	Back-ground	Multi Spatial	Total
$s_M = 0$	57%	60%	57%	57%	58%

Table 5 Ablation study on the motion conditioning in MoCA comparing the video edits conditioned on the motion of the source video against those without any motion conditioning. Human raters show a preference to our model with motion conditioning.

5.6 Effect of Motion Conditioning

Lastly, we perform an ablation study on the motion conditioning introduced in our method. Table 5 shows a comparison between our method with and without the motion conditioning support. Both models are trained on the same data for the same amount of iterations. We only perform evaluations on a subset of edit types

(Style, Object, Background, Multi-Spatial) as our method does not use motion conditioning for the other motion-based edits ($s_M = 0$). For spatial edits, we find we are able to better preserve the original motion of the entities through motion conditioning. Figure 5 Shows an example of when motion conditioning is beneficial in our model.

6 Discussion

We introduced MoCA, a method that decomposes the video editing problem into spatial and temporal components. Spatial edits are applied to the first frame of a source video, and then extrapolated using a motion-conditioned image animation model to preserve the motion of the original video. In addition, we allow motion editing by removing the motion conditioning and letting the animation model generate new frames according to the motion described in the edit prompt. We demonstrate that this simple method is a strong baseline outperforming existing methods on video editing. In addition, we introduce a new curated subset of video edits focused on motion editing, as well as a comprehensive analysis and benchmarking across a wide range of other video edits. By providing this comprehensive framework, we aim to facilitate the assessment of advancements and abilities of video editing techniques in subsequent research. We identify several limitations as directions for future work.

- Analysis in Section 5.5 showed that all existing evaluation metrics for video editing are rather lacking in their alignment with human judgement. As such, there remains room for developing more accurate evaluation metrics for video editing, as human evaluations can be time consuming or expensive when using crowd sourced workers. In addition, a strong automatic metric may be useful for automatic selection of desired edits for hyperparameter searching or when comparing results from different random seeds.
- Due to our reliance on video extrapolation as means for video editing, our method has less fidelity when preserving any aspects of source videos that is introduced after the first frame, such as longer videos, or videos with more camera motion. Further work may involve incorporating other conditioning schemes that aim to preserve these parts of source videos, similar to our proposal of augmenting a video extrapolation model with motion conditioning to preserve motion changes.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.

[3] 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*, pages 18392–18402, 2023.

[4] 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. *arXiv preprint arXiv:2304.08465*, 2023.

[5] Duygu Ceylan, Chun-Hao P Huang, and Niloy J Mitra. Pix2video: Video editing using image diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23206–23217, 2023.

[6] Huiwen Chang, Han Zhang, Jarred Barber, AJ Maschinot, Jose Lezama, Lu Jiang, Ming-Hsuan Yang, Kevin Murphy, William T Freeman, Michael Rubinstein, et al. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023.

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[8] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.

[9] Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *arXiv preprint arXiv:2108.00946*, 2021.

[10] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

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

[12] Rohit Girdhar, Mannat Singh, Andrew Brown, Quentin Duval, Samaneh Azadi, Sai Saketh Rambhatla, Akbar Shah, Xi Yin, Devi Parikh, and Ishan Misra. Emu video: Factorizing text-to-video generation by explicit image conditioning. *arXiv preprint arXiv:2311.10709*, 2023.

[13] 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.

[14] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[15] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

[16] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[17] Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761*, 2023.

[18] Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*, 2023.

[19] 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.

[20] Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6038–6047, 2023.

[21] Eyal Molad, Eliahu Horwitz, Dani Valevski, Alex Rav Acha, Yossi Matias, Yael Pritch, Yaniv Leviathan, and Yedid Hoshen. Dreamix: Video diffusion models are general video editors. *arXiv preprint arXiv:2302.01329*, 2023.

[22] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023.

[23] Hao Ouyang, Qiuyu Wang, Yuxi Xiao, Qingyan Bai, Juntao Zhang, Kecheng Zheng, Xiaowei Zhou, Qifeng Chen, and Yujun Shen. Codef: Content deformation fields for temporally consistent video processing. *arXiv preprint arXiv:2308.07926*, 2023.

[24] 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. *arXiv preprint arXiv:2303.09535*, 2023.

[25] 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.

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[29] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[30] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022.

[31] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

[32] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[33] Charles Spearman. The proof and measurement of association between two things. 1961.

[34] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.

[35] Linoy Tsaban and Apolinário Passos. Ledits: Real image editing with ddpm inversion and semantic guidance. *arXiv preprint arXiv:2307.00522*, 2023.

[36] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[37] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022.

[38] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023.

[39] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[40] 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*, pages 7623–7633, 2023.

[41] Jay Zhangjie Wu, Xiuyu Li, Difei Gao, Zhen Dong, Jinbin Bai, Aishani Singh, Xiaoyu Xiang, Youzeng Li, Zuwei Huang, Yuanxi Sun, Rui He, Feng Hu, Junhua Hu, Hai Huang, Hanyu Zhu, Xu Cheng, Jie Tang, Mike Zheng Shou, Kurt Keutzer, and Forrest Iandola. Cvpr 2023 text guided video editing competition, 2023.

[42] Shuai Yang, Yifan Zhou, Ziwei Liu, and Chen Change Loy. Rerender a video: Zero-shot text-guided video-to-video translation. *arXiv preprint arXiv:2306.07954*, 2023.

[43] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. *arXiv preprint arXiv:2311.10982*, 2023.

[44] 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*, pages 3836–3847, 2023.

[45] Yabo Zhang, Yuxiang Wei, Dongsheng Jiang, Xiaopeng Zhang, Wangmeng Zuo, and Qi Tian. Controlvideo: Training-free controllable text-to-video generation. *arXiv preprint arXiv:2305.13077*, 2023.

Appendix

A Qualitative Examples

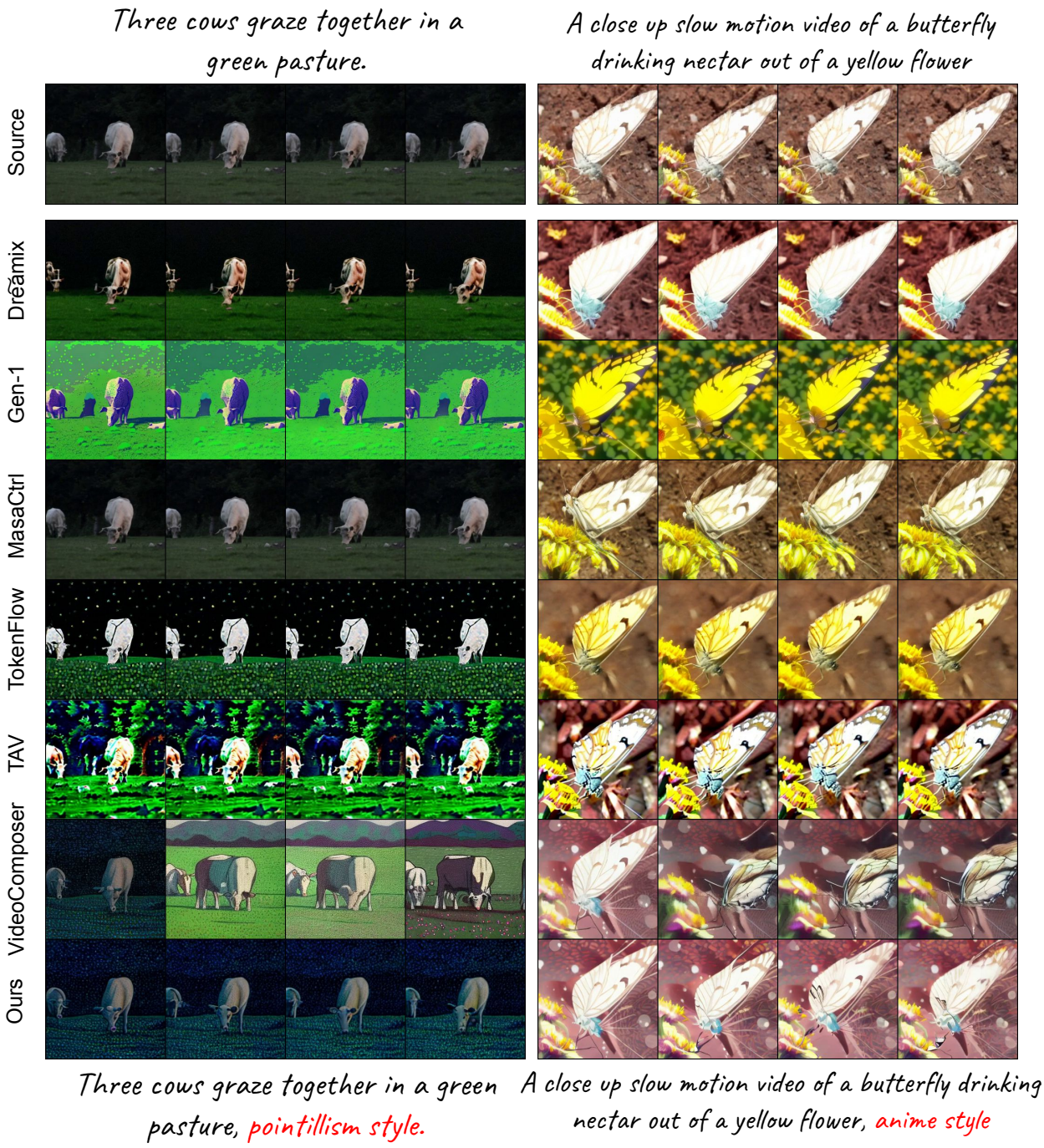

Figure 6 Comparisons for style video edit prompts

Figure 7 Comparisons for background video edit prompts

Figure 8 Comparisons for object video edit prompts

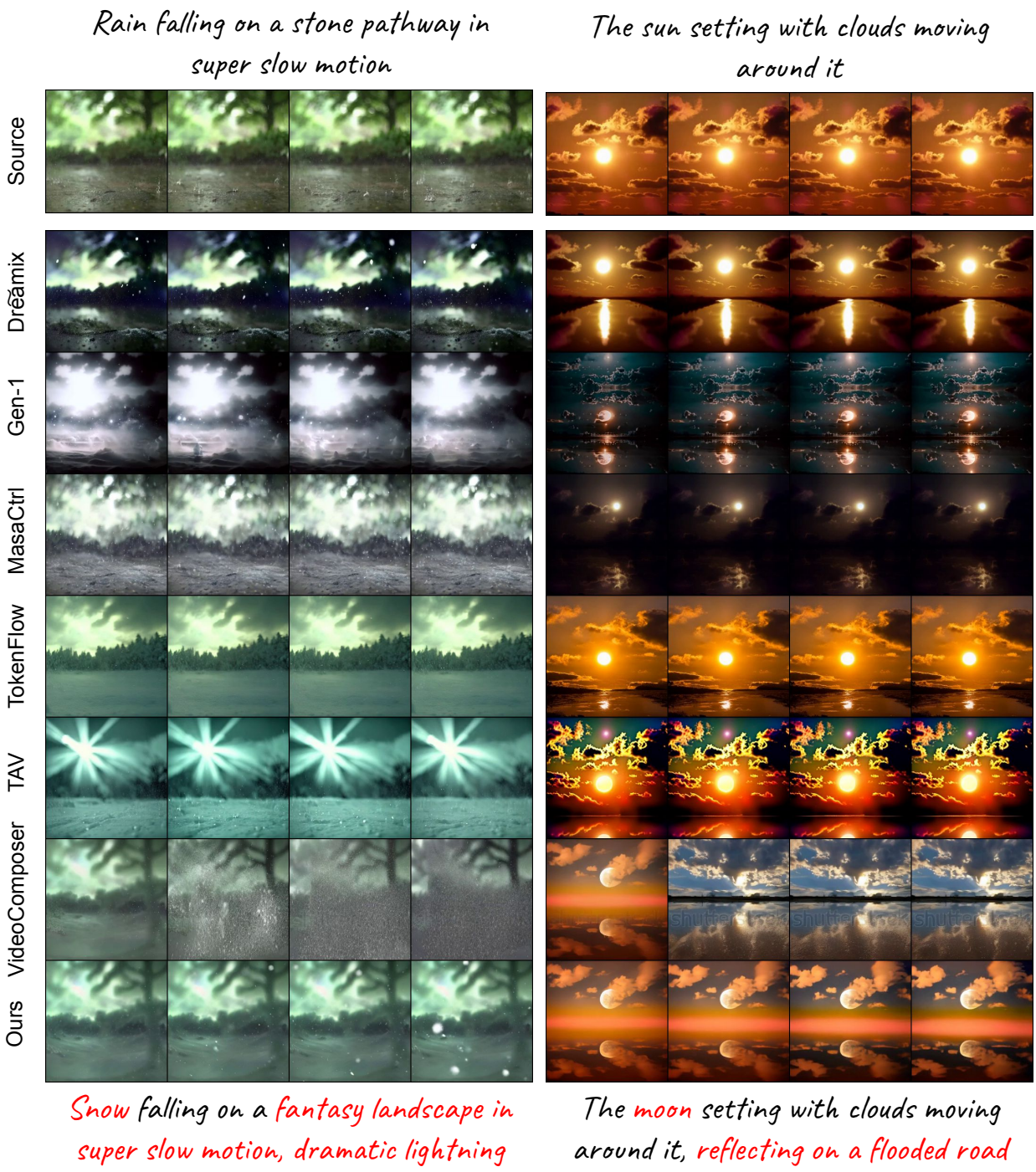

Figure 9 Comparisons for multi-spatial video edit prompts

Figure 10 Comparisons for motion video edit prompts

Figure 11 Comparisons for multi-motion video edit prompts

B Automatic Evaluation Results

We provide a more detailed comparison on different video editing methods using the VideoCLIP-based M_{geo} metric in Table 7. The results confirm the superiority of MoCA to other methods for all edit tasks. Additionally, we analyze the Spearman correlation [33] between automatic metrics introduced in Section 5.5 and human judgements. The results in Table 6 suggest the VideoCLIP based M_{geo} and M_{dir} metrics as the most reliable automatic metric in evaluating the performance of video editing models.

		Style	Background	Object	Motion	Multi-Spatial	Multi-Motion	Total
ImageCLIP	M_{sim}	-0.095	-0.102	-0.042	0.174	-0.076	-0.025	-0.006
	M_{dir}	0.238	0.290	0.161	0.035	0.219	0.141	0.150
	M_{geo}	0.240	0.288	0.156	0.047	0.209	0.137	0.152
VideoCLIP	M_{sim}	-0.080	-0.128	0.010	0.323	-0.080	0.006	0.036
	M_{dir}	0.290	0.328	0.183	0.049	0.202	0.189	0.189
	M_{geo}	0.300	0.304	0.189	0.140	0.201	0.205	0.203

Table 6 [33] correlation measuring the alignment between human judgements for editing quality and various CLIP-based metrics. Among all six metrics in the table, M_{geo} and M_{dir} scores computed with a VideoCLIP model show the highest correlation with human ratings. Trends are similar to classification results shown in Table 3, with higher correlation in spatial edits, and lower for motion-based edits.

Method	Style	Background	Object	Motion	Multi-Spatial	Multi-Motion
MoCA	**0.331**	**0.375**	**0.370**	0.185	**0.349**	**0.334**
Dreamix	0.2223	0.304	0.356	0.141	0.290	0.321
Gen-1	0.254	0.317	0.295	0.146	0.309	0.209
MasaCtrl	0.225	0.253	0.295	0.154	0.270	0.283
Tune-a-Video	0.223	0.261	0.346	0.164	0.303	0.273
TokenFlow	0.206	0.239	0.314	0.0963	0.301	0.226
VideoComposer	0.259	0.328	0.326	**0.187**	0.301	0.202

Table 7 M_{geo} metric computed for each model based on VideoCLIP features, averaged over all edit examples per manipulation type. This table presents a more detailed split of results shown in Table 4 by edit type.