Video Editing via Factorized Diffusion Distillation

Uriel Singer*, Amit Zohar*, Yuval Kirstain, Shelly Sheynin, Adam Polyak, Devi Parikh, and Yaniv Taigman

Meta AI

Abstract. We introduce Emu Video Edit (EVE), a model that establishes a new state-of-the art in video editing without relying on any supervised video editing data. To develop EVE we separately train an image editing adapter and a video generation adapter, and attach both to the same text-to-image model. Then, to align the adapters towards video editing we introduce a new unsupervised distillation procedure, Factorized Diffusion Distillation. This procedure distills knowledge from one or more teachers simultaneously, without any supervised data. We utilize this procedure to teach EVE to edit videos by jointly distilling knowledge to (i) precisely edit each individual frame from the image editing adapter, and (ii) ensure temporal consistency among the edited frames using the video generation adapter. Finally, to demonstrate the potential of our approach in unlocking other capabilities, we align additional combinations of adapters.

Keywords: Video · Editing · Diffusion · Adapters · Distillation

1 Introduction

The increasing usage of video as online content has led to a rising interest in developing text-based video editing capabilities. However, due to the scarcity of supervised video editing data, developing such capabilities has proven to be challenging. To address this challenge, the research community has mostly focused on training-free methods [10, 16, 17, 34, 40]. Unfortunately, these methods thus far appear to be limited both in terms of performance, and in the range of editing capabilities that they offer (Fig. 3).

Therefore, we introduce a new approach that allows us to train a *state-of-the-art* video editing model *without* any supervised video editing data. The main insight behind our approach is that we can decouple our expectations from a video editing model into two distinct criteria: (i) precisely edit each individual frame, and (ii) ensure temporal consistency among the edited frames.

Leveraging this insight we follow two phases. In the first phase, we train two separate adapters on top of the same frozen text-to-image model; an image editing adapter, and a video generation adapter. Then, by applying both adapters

* Equal contribution.

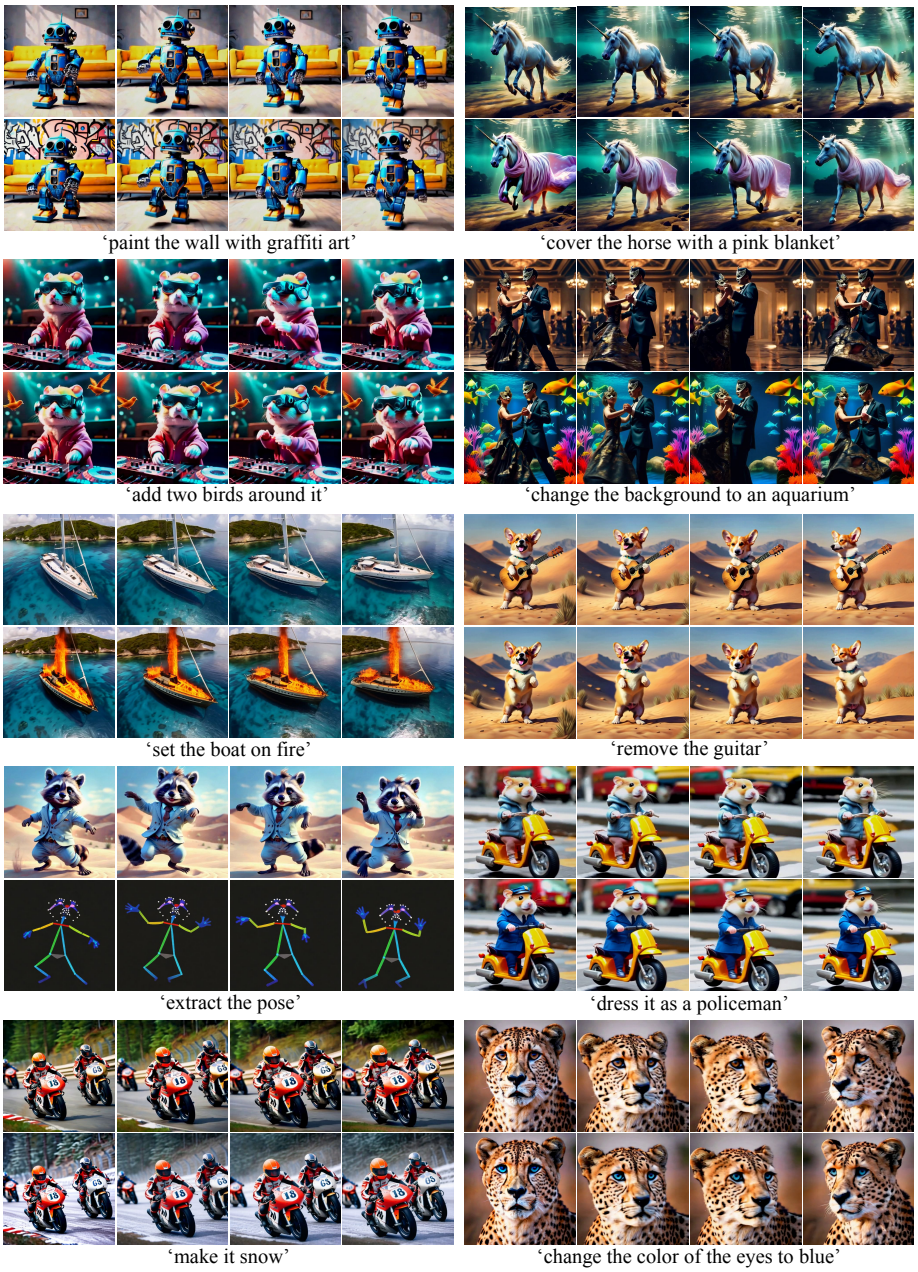

Fig. 1: EVE is a text-guided video editing model that enables various editing tasks.

simultaneously we enable limited video editing capabilities. In the second phase, we introduce a new unsupervised alignment method, Factorized Diffusion Distillation (FDD), that drastically improves the video editing capabilities of our model. FDD assumes a student model and one or more teacher models. We employ the adapters as teachers, and for our student, we utilize trainable low-rank adaptation (LoRA) weights on top of the frozen text-to-image model and adapters. At each training iteration FDD generates an edited video using the student. Then, it uses the generated video to provide supervision from all teachers via Score Distillation Sampling [27] and adversarial losses [12] (Fig. 2).

The resulting model, Emu Video Edit (EVE), sets state-of-the-art results on the Text Guided Video Editing (TGVE) benchmark [38]. Additionally, we improve two aspects of the evaluation protocol that was set in TGVE. First, we utilize the recent ViCLIP [35] model to introduce additional automatic metrics that are temporally aware. Second, we introduce TGVE+, an extended version of the TGVE benchmark which facilitates new important editing tasks like adding, removing, and changing the texture of objects in the video. Importantly, EVE also exhibits state-of-the-art results when tasked with these additional editing operations (Tab. 1).

Notably, our approach can theoretically be applied to any arbitrary group of diffusion-based adapters. To verify that this holds in practice, we utilize our approach to develop personalized image editing models by aligning an image editing adapter with different LoRA adapters (Fig. 5).

In summary, our method utilizes an image editing adapter and a video generation adapter, and aligns them to accommodate video editing using an unsupervised alignment procedure. The resulting model, EVE, exhibits state-of-the-art results in video editing while offering diverse video editing capabilities. Furthermore, we broaden the evaluation protocol for video editing by suggesting additional automatic metrics and presenting TGVE+, a benchmark that extends TGVE with additional editing tasks. Finally, we verify that our approach can be used to align other adapters, and therefore, holds the potential to unlock new capabilities.

2 Related Work

The lack of supervised video editing data poses a major challenge in training precise and diverse video editing models. A common strategy to address this challenge is via training-free solutions. Initial work proposed the use of Stochastic Differential Editing (SDEdit) [24]. This approach performs image editing by adding noise to the input image and then denoising it while conditioning the model on a caption that describes the edited image. Recently, several video foundation models, such as Lumiere [1] and SORA [3], showcased examples in which they utilize SDEdit for video editing. While this approach can preserve the general structure of the input video, adding noise to the input video results in the loss of crucial information, such as subject identity and textures. Hence, SDEdit may work well when attempting to change the style of an image, but by design, it is unsuitable for *precise* editing.

A more dominant approach is to inject information about the input or generated video from key frames via cross-attention interactions [4, 10, 16, 17, 19, 23, 34, 36, 40, 42]. Another common strategy is to extract features that should persist in the edited video, like depth maps or optical flow, and train the model to denoise the original video while using them [8, 20, 39]. Then, during inference time, one can predict an edited video while using the extracted features to ensure faithfulness to the structure or motion of the input video. The main weakness of this strategy is that the extracted features may lack information that should persist (e.g. pixels of a region that should remain intact), or hold information that should be altered (e.g. if the editing operation requires adding new motion to the video). Consequently, the edited videos may still suffer from unfaithfulness to the input video or editing operation.

To improve faithfulness to the input video at the cost of latency, some works [37, 41] invert the input video using the input caption. Then, they generate a new video while using the inverted noise and a caption that described the output video. Another work [5] adapts the general strategy of InstructPix2Pix [2] to video editing, which allows them to generate and train a video editing model using synthetic data. While this approach seems to be effective, recent work in image editing [32] shows that Prompt-to-Prompt [13] can yield sub-optimal results for various editing operations.

In this paper we deviate from prior work. Instead, we distill distinct video editing capabilities from an image editing teacher and a video generation teacher. Similarly to the Adversarial Diffusion Distillation (ADD) [30] loss, our approach involves combining a Score Distillation Sampling [27] loss and an adversarial loss [12]. However, it significantly differs from ADD. First, our method is unsupervised, and thus generates all data that is used for supervision rather than utilizing a supervised dataset. Second, we use distillation to learn a new capability, rather than reduce the number of required diffusion steps. Third, we learn this new capability by factorizing the distillation process and leverage more than one teacher model in the process.

3 Method

The key insight behind our approach is that video editing requires two main capabilities: (1) precisely editing images, and (2) ensuring temporal consistency among generated frames. In Sec. 3.1 we detail how we develop a dedicated adapter for each capability. Next, we describe how our final architecture combines the adapters to enable video editing. Finally, in Sec. 3.2 we introduce Factorized Diffusion Distillation (FDD), our method to align the adapters. In Fig. 2 we provide an illustration of our model's architecture and FDD.

3.1 Architecture

Our architecture involves stacking an image editing adapter and a video generation adapter on top of the same text-to-image backbone. We employ the latent diffusion model, Emu [7], as our backbone model, and denote its weights with

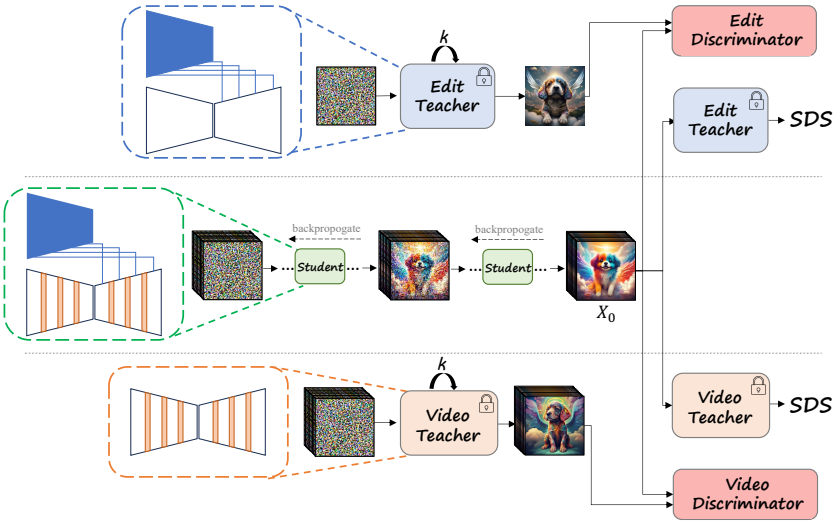

Fig. 2: Model architecture and alignment procedure. We train an adapter for image editing (in blue) and video generation (in orange) on top of a shared text-to-image backbone. Then, we create a student network by stacking both adapters together on the shared backbone (in green) and align the two adapters. The student is trained using (i) score distillation from each frozen teacher adapter (marked as SDS), (ii) adversarial loss for each teacher (in pink). SDS is calculated on samples generated by the student from noise and the discriminators attempt to differentiate between samples generated by the teachers and the student.

θ. We next describe how we develop and combine the different components to enable video editing.

Video Generation Adapter For the video generation adapter we make us of Emu Video [11], which coincidentally aligns with our adapter methodology, as it contains temporal layers trained on top of a frozen text-to-image model. However, we still retrain Emu Video to ensure both adapters share the same text-to-image backbone and diffusion process configuration (such as zero-SNR). Formally, the text-to-video output is denoted as $\hat{x}_\rho(x_s, s, c_{out})$, where $\rho = [\theta, \theta_{video}]$ are the text-to-image and video adapter weights, x_s is a noisy video sample, s is the timestep, and c_{out} is the output video caption.

Image Editing Adapter To create an image editing adapter, we train a ControlNet [43] adapter, with parameters θ_{edit}, on the training dataset developed to train Emu Edit [32]. This differs from the original Emu Edit model, where the entire text-to-image model was fine-tuned. We follow the standard practice of ControlNet training, and initialize the adapter with copies of the down and middle blocks of the text-to-image model. During training we condition the text-to-image model on the output image caption, while using the input image and the edit instruction as inputs to our ControlNet image editing adapter. Hence, we can denote the output of the image editing model with $\hat{x}_\psi(x_s, s, c_{out}, c_{instruct}, c_{img})$,

where $\psi = [\theta, \theta_{edit}]$ are the text-to-image and image editing adapter weights, x_s is a noisy image sample, s is the timestep, c_{out} is the output image caption, $c_{instruct}$ is the textual edit instruction, and c_{img} is the input image.

Combining The Adapters To enable video editing capabilities we attach both of the adapters simultaneously to the text-to-image backbone [1]. Formally, we aim to denoise a noisy edited video x_s, using an input video c_{vid}, editing instruction $c_{instruct}$, and an output video caption c_{out}.

Notably, when attaching the image editing adapter alone, the resulting function will process each frame independently. Therefore, each frame in the predicted video should be precise and faithful to the input frame and editing instruction, but lack consistency with respect to the rest of the edited frames. Similarly, when attaching the video generation adapter alone, the resulting function will generate a temporally consistent video that is faithful to the output caption, but not necessarily faithful to the input video.

When combining both adapters with the shared text-to-image backbone, the resulting function is $\hat{x}_\eta(x_s, s, c_{out}, c_{instruct}, c_{vid})$, where $\eta = [\theta, \theta_{edit}, \theta_{video}]$. This formulation should enable editing a video that is both temporally consistent and faithful with respect to the input. In practice, we observe that even though this "plug-and-play" approach enables video editing capabilities, it still includes significant artifacts, as we show in Sec. 4.3.

Final Architecture As the necessary knowledge already exists in the adapters, we expect a small alignment to be sufficient. Hence, we keep the adapters frozen and utilize low-rank adaptation (LoRA) [14] weights θ_{align} over the text-to-image backbone. Our final architecture becomes $\phi = [\theta, \theta_{edit}, \theta_{video}, \theta_{align}]$. We describe in the following section how we train θ_{align} to improve the video editing quality of our model.

3.2 Factorized Diffusion Distillation

To train θ_{align} and align the adapters without supervised video editing data, we propose a new unsupervised distillation procedure, Factorized Diffusion Distillation (FDD). In this procedure we freeze both adapters, and jointly distill their knowledge into a video editing student. Since our approach cannot assume supervised data, we only collect a dataset for the inputs. Each data point in our dataset consists of $y = (c_{out}, c_{instruct}, c_{vid})$, where c_{out} is an output video caption, $c_{instruct}$ is the editing instruction, and c_{vid} is the input video. We provide further details on this dataset in Sec. 4.

In each iteration of FDD, we first utilize the student to generate an edited video x'_0 using a data point y for k diffusion steps (see Sec. 3.3 for more details). Importantly, we will later backpropagate the loss through all of these diffusion steps. We then apply Score Distillation Sampling (SDS) [27] loss using each

[1] To enable the editing adapter to process videos, we stack the frames independently as a batch.

teacher. Concretely, we sample noise ϵ and a time step t, and use them to noise x'_0 into x'_t. Then, we task each of the teachers to predict the noise from x'_t independently. For a teacher $\hat{\epsilon}$, the SDS loss is the difference between ϵ and the teacher's prediction:

$$\mathcal{L}_{\text{SDS}}(\hat{x}) = \mathbb{E}_{x'_0,t,\epsilon}[c(t)sg(\hat{\epsilon}(x'_t,t)-\epsilon)x'_0],$$

where $c(t)$ is a weighting function, and sg indicates that the teachers are kept frozen. The metric is averaged over student generations x'_0, sampled timesteps t and noise ϵ. Plugging in the edit and video teachers, the loss becomes

$$\mathcal{L}_{\text{SDS-Edit}} = \mathcal{L}_{\text{SDS}}(\hat{x}_\psi), \mathcal{L}_{\text{SDS-Video}} = \mathcal{L}_{\text{SDS}}(\hat{x}_\rho),$$

For brevity, we omit input conditions from $\hat{x}_\phi, \hat{x}_\psi, \hat{x}_\rho$. Each teacher provides feedback for a different criterion: the image editing adapter for editing faithfully and precisely, and the video generation adapter for temporal consistency.

Similar to previous works employing distillation methods [25], we observe blurry results, and therefore utilize an additional adversarial objective [12] for each of the teachers, akin to Adversarial Diffusion Distillation (ADD) [31]. Specifically, we train two discriminators. The first, D_e, receives an input frame, instruction, and output frame and attempts to determine if the edit was performed by the image editing teacher or video editing student. The second, D_v, receives a video and caption, and attempts to determine if the video was generated by the video generation teacher or video editing student. We further follow ADD and employ the hinge loss [21] objective for adversarial training. Hence, the discriminators minimize the following objectives:

$$\mathcal{L}_{\text{D-Edit}} = \mathbb{E}_{x'_\psi}[\max(0, 1 - D_e(x'_\psi))] + \mathbb{E}_{x'_0}[\max(0, 1 + D_e(x'_0))],$$
$$\mathcal{L}_{\text{D-Video}} = \mathbb{E}_{x'_\rho}[\max(0, 1 - D_v(x'_\rho))] + \mathbb{E}_{x'_0}[\max(0, 1 + D_v(x'_0))],$$

while the student minimizes the following objectives:

$$\mathcal{L}_{\text{G-Edit}} = -\mathbb{E}_{x'_0}[\max(0, 1 + D_e(x'_0))],$$
$$\mathcal{L}_{\text{G-Video}} = -\mathbb{E}_{x'_0}[\max(0, 1 + D_v(x'_0))],$$

where x'_ψ and x'_ϕ are samples generated from random noise by applying the image editing and video generation teachers accordingly for multiple forward diffusion steps using DDIM sampling.

The combined loss to train our student model is:

$$\mathcal{L}_{\text{G-FDD}} = \alpha(\mathcal{L}_{\text{G-Edit}} + \lambda\mathcal{L}_{\text{SDS-Edit}}) + \beta(\mathcal{L}_{\text{G-Video}} + \lambda\mathcal{L}_{\text{SDS-Video}}),$$

and the discriminators are trained with:

$$\mathcal{L}_{\text{D-FDD}} = \alpha\mathcal{L}_{\text{D-Edit}} + \beta\mathcal{L}_{\text{D-Video}}.$$

In practice, we set both α and β to 0.5. Following [31], we set λ to 2.5.

3.3 Implementation Details

We provide below further implementation details on the timestep sampling during FDD, and the architecture of our discriminators.

K-Bin Diffusion Sampling As previously mentioned, our student generates an edited video using k diffusion steps, and we backpropagate the loss through all of the steps. During training we set $k = 3$, as it is the maximum number of diffusion steps that fits into memory. Notably, naively using the same k timesteps during training, and setting a larger k during inference time may lead to a train-test discrepancy. To avoid such a train-test discrepancy we divide the T diffusion steps into k evenly sized bins, each containing T/k steps. Then, during each training generation iteration, we randomly select a step from its corresponding bin. We ablate this strategy in Sec. 4.3.

Discriminator Architecture The base architecture of our discriminators is similar to [31]. Specifically, we utilize DINO [26] as a frozen feature network and add trainable heads to it. To add conditioning to the input image for D_e, we use an image projection in addition to the text and noisy image projection, and combine the conditions with an additional attention layer. To support video conditioning for D_v, we add a single temporal attention layer over the projected features of DINO, applied per pixel.

4 Experiments

We perform a series of experiments to evaluate and analyze our approach. We start by evaluating our method on the task of instruction-guided video editing. Specifically, we benchmark our video editing model, Emu Video Edit (EVE), versus multiple baselines on the Text-Guided Video Editing (TGVE) [38] benchmark. Additionally, we expand TGVE with additional editing tasks, and benchmark our model against the expanded benchmark as well. Then, we carry out an ablation study to measure the impact of different design choices that we introduced into our approach. We continue by examining the capability of EVE to perform zero shot video editing over tasks which were not presented during the alignment phase, but are part of the editing adapter's knowledge. Finally, we conduct a qualitative study to verify that our approach can be applied to align other combinations of adapters. Video editing examples are presented in Fig. 1, with additional samples and qualitative comparisons available on the supplementary material and website.[2]

Metrics. The goal of text-based video editing is to modify a video in accordance with the input text, while preserving elements or aspects of the video that should not change. To this end, our evaluation in based on subjective and objective success metrics. We utilize two categories of objective metrics for evaluation. The first category includes metrics used by the TGVE competition: (i)

[2] `https://fdd-video-edit.github.io/`

CLIPFrame (Frame Consistency) [38] – measuring the average cosine similarity among CLIP image embeddings across all video frames, and (ii) PickScore [18] – measuring the predicted human preference averaged over all video frames.

One inherent limitation of both metrics is their lack of consideration for temporal consistency. For example, CLIPFrame applies a simple average over a similarity score between images. As a result, CLIPFrame favours static videos with limited or no motion. When evaluating video-editing methods, output videos with little or no modification will score higher on this metric.

Therefore, we introduce additional metrics that makes use of ViCLIP [35], a video CLIP [28] model that considers temporal information when processing videos. Concretely, we add the following metrics: (i) ViCLIP text-video direction similarity (ViCLIP_{dir}, inspired by CLIP_{dir} [9]) – measuring agreement between change in captions and the change in videos, and (ii) ViCLIP output similarity (ViCLIP_{out} [28]) – measuring edited image similarity with output caption.

For subjective evaluation, we follow the TGVE [38] benchmark and rely on human raters. We present the human raters with the input video, a caption describing the output video, and two edited videos. We then task the raters to answer the following questions: (i) Text alignment: Which video better matches the caption, (ii) Structure: Which video better preserves the structure of the input video, and (iii) Quality: Aesthetically, which video is better. In addition, we report an overall human evaluation score by averaging the preference score of all three questions.

Unsupervised Dataset Our FDD approach requires a dataset with inputs for the student and teachers. In the case of video editing, each data point contains $y = (c_{out}, c_{instruct}, c_{vid})$, where c_{out} is an output video caption, $c_{instruct}$ is the editing instruction, and c_{vid} is the input video. To create this dataset we utilize the videos from Emu Video's high-quality dataset, which has 1600 videos. For each video, we follow [32] and task Llama-2 [33] with generating seven editing instructions, one for each of the following tasks from Emu Edit: Add, Remove, Background, Texture, Local, Style, Global.

Editing Tasks Unseen During Alignment Throughout the alignment phase, we train our student model on a subset of the tasks that our image editing adapter was originally trained on. For example, we do not train the student to segment objects in the input video, extract the pose, convert the video into a sketch, or derive depth maps from the input video. However, we observe a significant improvement in student performance on these tasks following the alignment stage. This suggests that our student model aligns with the entire knowledge of the teacher model, even if we only expose it to a subset of this knowledge. We provide a qualitative comparison in Fig. 4 to illustrate the contribution of the alignment phase to the tasks mentioned above.

Training Details We train both adapters using a frozen Emu backbone, with zero-terminal SNR [22] enforced during training. We train the model for a total of 1500 iterations with a batch size of 64 and a fixed learning rate of 1e-5. We train

on 8-frame 512 × 512 video clips, with SDS losses for the first 1000 iterations and adversarial losses added for the next 500. Training takes approximately 7 hours on 64 H100 GPUs. Notably, our adapter-based approach significantly reduces computational overhead, allowing the entire model to fit within a single GPU, as a single instance of the text-to-image and adapters are instantiated for both the student and teachers. Editing a video on a single H100 GPU takes approximately 6 seconds, which is comparable to generating a similar length video using Emu Video. We condition the edit adapter on the task label [32] and the video adapter on the first frame [11] edited by the edit adapter. To generate videos longer than 8 frames, we follow [6] and apply a sliding window approach.

4.1 Video Editing Benchmark

At present, the TGVE [38] benchmark is the established standard for evaluating text-based video editing methods. The benchmark contains seventy six videos, and each has four editing prompts. All videos are either 32 or 128 frames with a resolution of 480×480. The benchmark encompasses four types of editing tasks: (i) local object modification, (ii) style change, (iii) background change, and (iv) execution of multiple editing tasks simultaneously.

Due to TGVE's focus on a narrow range of editing tasks we choose to increase its diversity, by adding three new editing tasks: (i) object removal (Remove), (ii) object addition (Add), and (iii) texture alterations (Texture). For each video in TGVE and for each new editing operation, we assign crowd workers the task of writing down an editing instruction and an output caption describing a desired output video. In the remainder of this section, we report the performance of our method on the TGVE benchmark and our extension thereof, which we name TGVE+. In the supplementary material, we provide examples of our benchmark, which we are publicly releasing to support ongoing research in video editing.

4.2 Baseline Comparisons

We benchmark our model against InsV2V [6], which is the leading performer in the TGVE benchmark. For completeness, we also compare to Space-Time Diffusion Features (STDF) [42], which is one of the latest motion transfer works, Tune-A-Video (TAV) [37], which served as the baseline in the TGVE contest, SDEdit [24] which is a popular diffusion editing baseline, and Fairy [36]. For SDEdit, we use a noising level of 0.75 after comparing multiple noising levels and choosing the best one with respect to automatic metrics. Unlike the official TGVE contest, which compared all participating methods to TAV, we directly compare our model with the different baselines.

Tab. 1 shows our results versus the baselines. As can be seen, human raters prefer EVE over all baselines by a significant margin. Moreover, when considering automatic metrics, EVE presents state-of-the-art results on all of the objective metrics except for CLIPFrame. Interestingly, while STDF and Fairy achieve the highest scores on the CLIPFrame metric, human raters prefer our model in 72.4% and 71.7% of the time respectively. In addition, Fig. 3 provides a visual comparison between the outputs of EVE and top performing baselines.

Table 1: Comparison with video-editing baselines on the TGVE and TGVE+ benchmarks. We report PickScore, CLIPFrame, ViCLIP metrics and human ratings. Human evaluation shows the percentage of raters that prefer the results of EVE.

Dataset	Method	PickScore↑	CLIPFrame↑	ViCLIP$_{dir}$ ↑	ViCLIP$_{out}$↑	Text	Struct.	Quality	Avg.
TGVE	TAV [37]	20.36	0.924	0.162	0.243	72.4	74.0	85.2	77.2
	SDEdit [24]	20.18	0.896	0.172	0.253	75.7	67.4	79.0	74.0
	STDF [42]	20.40	**0.933**	0.110	0.226	81.3	65.8	70.1	72.4
	Fairy [36]	19.80	**0.933**	0.164	0.208	77.3	62.8	75.0	71.7
	InsV2V [6]	**20.76**	0.911	0.208	**0.262**	57.9	55.9	65.1	59.6
	EVE (Ours)	**20.76**	0.922	**0.221**	**0.262**	–	–	–	–
TGVE+	TAV [37]	20.47	**0.933**	0.131	0.242	72.2	74.0	77.2	74.5
	SDEdit [24]	20.35	0.899	0.144	0.246	74.5	68.5	77.9	73.6
	STDF [42]	20.60	**0.933**	0.093	0.227	78.6	70.2	72.6	73.8
	Fairy [36]	19.81	**0.933**	0.140	0.197	74.4	70.8	77.8	74.3
	InsV2V [6]	20.37	0.925	0.174	0.236	62.9	56.4	61.4	60.2
	EVE (Ours)	**20.88**	0.926	**0.198**	**0.251**	–	–	–	–

Table 2: Ablation study on our different contributions. We report human ratings on video-editing results on the TGVE benchmark. Human evaluation shows the percentage of raters that prefer the results of EVE.

Method	Text	Struct.	Quality	Avg.
Random Init	96.7	70.1	94.7	87.2
w/o alignment	77.6	91.4	89.8	86.3
w/o SDS	77.6	87.5	92.1	85.7
w/o discriminators	74.3	84.2	83.9	80.8
w/o K-Bin Sampling	57.6	49.7	51.6	53.0

4.3 Ablations

We provide an ablation study of human ratings in Tab. 2 to assess the effectiveness of our different contributions on the TGVE+ benchmark. We begin our study by ablating our decision to add the pre-trained adapters to our student model rather than learning them jointly during the alignment procedure. In this experiment (Random Init), we initialize the ControlNet edit adapter with the weights from the text-to-image encoder, and the temporal layers are initialized to identity. We then fine-tune the entire resulting model. Our observation indicates that this particular variant is unsuccessful in acquiring proficiency in the video-editing task, implying that FDD is more adept at aligning pre-trained adapters rather than initiating their training from scratch.

We continue by ablating the design of the alignment procedure itself, examining three methods of combining the adapters: (i) without any alignment (w/o alignment), (ii) using only the adversarial loss and excluding SDS (w/o SDS), and (iii) incorporating SDS but excluding the adversarial loss (w/o Discriminators). As expected, using no alignment leads to poor results for both structure preservation and quality aspects. This suggests that FDD is essential when combining adapters that were trained separately for different tasks. When evaluating the contribution of each term in EVE, namely SDS and adversarial loss, the SDS

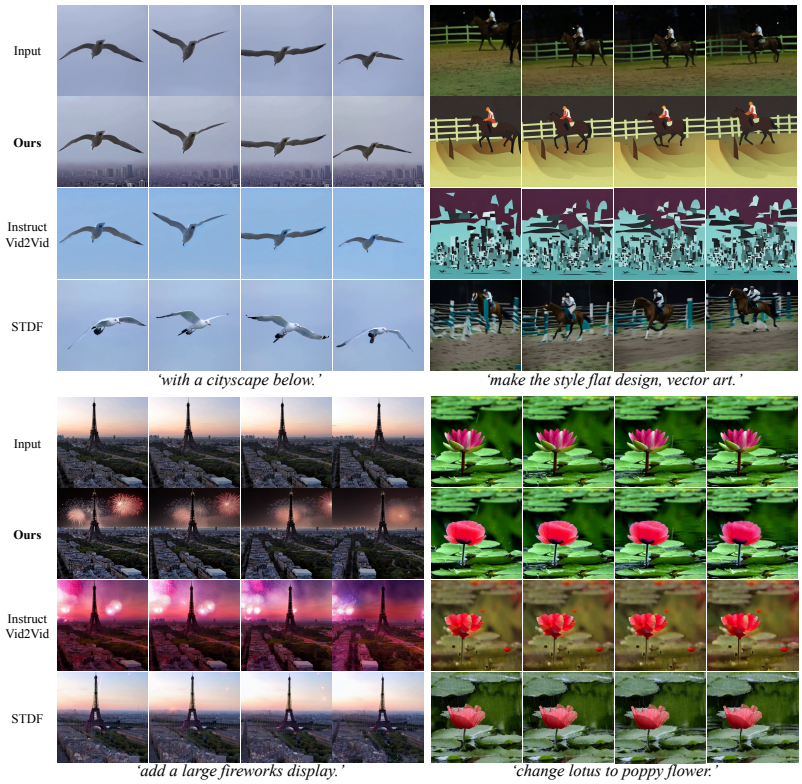

Fig. 3: Comparison of our model against baselines using examples from the Text-Guided Video Editing (TGVE) [38] benchmark and our extension of it.

term has a larger impact on the alignment process. Interestingly, employing the adversarial term alone is sufficient to achieve some level of alignment. However, the use of both terms is vital for successful alignment.
We conclude by validating the contribution of using K-Bin diffusion sampling. For this ablation, we sample k steps uniformly throughout training instead of randomly sampling them from k buckets. As evident from the results, the process of sampling steps from k bins further enhances the performance of FDD.

4.4 Additional Combinations of Adapters

In this section we explore the ability of FDD to align other adapters. To this end, we train four different LoRA adapters on our text-to-image backbone; two for subject-driven generation [29] and two for style-driven generation. We then align each of them with our image editing adapter to facilitate personalized and stylized image editing capabilities.

To create an unsupervised dataset for stylized editing, we utilize 1,000 (input caption, instruction, output caption) triplets from Emu Edit's dataset. For personalized editing, we use 1,000 input captions and task Llama-2 with generating instructions for adding the subject or replacing items with the subject. Notably,

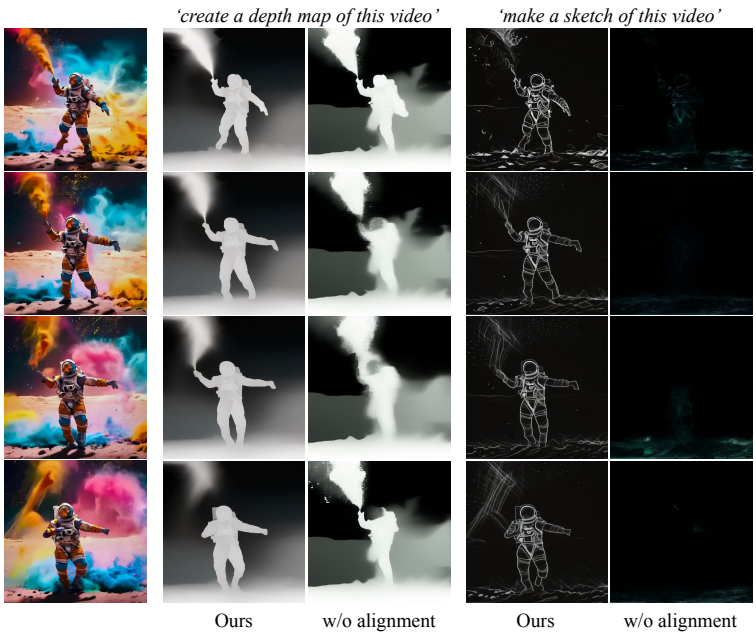

Fig. 4: Our model performs zero-shot video editing for tasks that Emu Edit can execute on images, without explicitly training on them during alignment.

we do not use images during training and instead generate the input images using the LoRA adapter. While each LoRA adapter requires a different alignment, we note that one can use a subject-conditioned adapter such as ReferenceNet [15] and perform a single alignment for all subjects and styles at once.

In Fig 5, we present qualitative examples of our method when applied to these combinations. For each input image and instruction we display the samples obtained by using: (i) vanilla Emu Edit, (ii) attaching both adapters without alignment, and (iii) after alignment. As anticipated, Emu Edit is not capable of personalized editing as it lacks awareness of the desired subject. Similarly, for stylized editing, it has difficulty maintaining the style of the input. When using a "plug-and-play" approach, the model either fails to maintain the style or subject identity, or it produces significant artifacts. However, after alignment, the edits become more consistent with the reference style and subject.

5 Limitations

We identify two main limitations in our approach. Fundamentally, the performance of our model is upper bounded by the capabilities of the different teacher models. For instance, we anticipate that the editing capabilities of our student model will not exceed those of the image editing adapter teacher. Secondly, our method is intended to align a student which is initialized with pre-trained adapters. As we show in Sec. 4.3, training it from scratch results in poor performance. Thus, it requires the teachers to be trained as adapters over a frozen

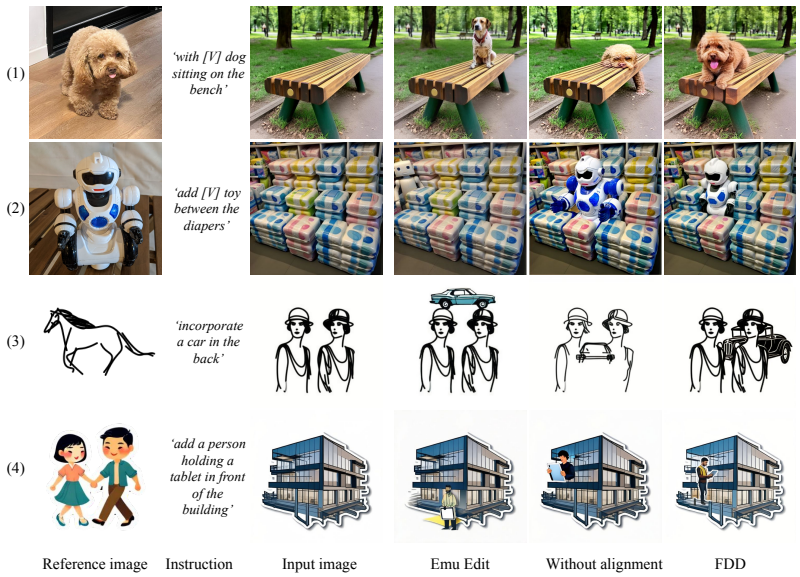

Fig. 5: We apply FDD to combine an editing adapter with LoRA-based adapters: (1) a subject-driven adapter for a dog, (2) a subject-driven adapter for a toy robot, (3) a style-driven adapter for line art, and (4) a style-driven adapter for stickers.

text-to-image backbone. We hope that future work will overcome these limitations to enable more efficient training and better performing models.

6 Conclusions

In this study, we proposed an approach for learning how to edit videos without supervised data. Our approach is based on the key insight that video editing can be factorized into two distinct criteria: (1) precise editing of video frames, and (2) ensuring temporal consistency among the edited frames. Leveraging this insight, we proposed a two-stage approach for training a video editing model. In the first stage, we separately train an image editing adapter and a video generation adapter, and attach both to a frozen text-to-image backbone. Subsequently, we align the adapters towards video editing using Factorized Diffusion Distillation (FDD). Our results demonstrate that the resulting model, Emu Video Edit (EVE), establishes a new state-of-the-art in video editing. Finally, we show the potential of our approach for learning other tasks by aligning additional adapters. There are still many opportunities for applying our approach to other tasks, and alleviating some of its limitations, and we are excited to see how the research community will utilize and build upon this work in the future.

Acknowledgements

Andrew Brown, Bichen Wu, Ishan Misra, Saketh Rambhatla, Xiaoliang Dai, Zijian He. Thank you for your contributions!

References

1. Bar-Tal, O., Chefer, H., Tov, O., Herrmann, C., Paiss, R., Zada, S., Ephrat, A., Hur, J., Li, Y., Michaeli, T., Wang, O., Sun, D., Dekel, T., Mosseri, I.: Lumiere: A space-time diffusion model for video generation. ArXiv **abs/2401.12945** (2024), https://api.semanticscholar.org/CorpusID:267095113
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
3. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
4. Ceylan, D., Huang, C.H.P., Mitra, N.J.: Pix2video: Video editing using image diffusion. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 23149–23160 (2023), https://api.semanticscholar.org/CorpusID:257663916
5. Cheng, J., Xiao, T., He, T.: Consistent video-to-video transfer using synthetic dataset. ArXiv **abs/2311.00213** (2023), https://api.semanticscholar.org/CorpusID:264833165
6. Cheng, J., Xiao, T., He, T.: Consistent video-to-video transfer using synthetic dataset. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=IoKRezZMxF
7. Dai, X., Hou, J., Ma, C.Y., Tsai, S., Wang, J., Wang, R., Zhang, P., Vandenhende, S., Wang, X., Dubey, A., Yu, M., Kadian, A., Radenovic, F., Mahajan, D., Li, K., Zhao, Y., Petrovic, V., Singh, M.K., Motwani, S., Wen, Y., Song, Y., Sumbaly, R., Ramanathan, V., He, Z., Vajda, P., Parikh, D.: Emu: Enhancing image generation models using photogenic needles in a haystack (2023)
8. Esser, P., Chiu, J., Atighehchian, P., Granskog, J., Germanidis, A.: Structure and content-guided video synthesis with diffusion models. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 7312–7322 (2023), https://api.semanticscholar.org/CorpusID:256615582
9. Gal, R., Patashnik, O., Maron, H., Chechik, G., Cohen-Or, D.: Stylegan-nada: Clip-guided domain adaptation of image generators. arXiv preprint arXiv:2108.00946 (2021)
10. Geyer, M., Bar-Tal, O., Bagon, S., Dekel, T.: Tokenflow: Consistent diffusion features for consistent video editing. ArXiv **abs/2307.10373** (2023), https://api.semanticscholar.org/CorpusID:259991741
11. Girdhar, R., Singh, M., Brown, A., Duval, Q., Azadi, S., Rambhatla, S.S., Shah, A., Yin, X., Parikh, D., Misra, I.: Emu video: Factorizing text-to-video generation by explicit image conditioning (2023)
12. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial networks. 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) pp. 1–7 (2022), https://api.semanticscholar.org/CorpusID:1033682
13. Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., Cohen-Or, D.: Prompt-to-prompt image editing with cross attention control. arXiv preprint arXiv:2208.01626 (2022)
14. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=nZeVKeeFYf9

15. Hu, L., Gao, X., Zhang, P., Sun, K., Zhang, B., Bo, L.: Animate anyone: Consistent and controllable image-to-video synthesis for character animation. arXiv preprint arXiv:2311.17117 (2023)
16. Kara, O., Kurtkaya, B., Yesiltepe, H., Rehg, J.M., Yanardag, P.: Rave: Randomized noise shuffling for fast and consistent video editing with diffusion models. arXiv preprint arXiv:2312.04524 (2023)
17. Khachatryan, L., Movsisyan, A., Tadevosyan, V., Henschel, R., Wang, Z., Navasardyan, S., Shi, H.: Text2video-zero: Text-to-image diffusion models are zero-shot video generators. arXiv preprint arXiv:2303.13439 (2023)
18. Kirstain, Y., Polyak, A., Singer, U., Matiana, S., Penna, J., Levy, O.: Pick-a-pic: An open dataset of user preferences for text-to-image generation. Advances in Neural Information Processing Systems **36** (2024)
19. Li, X., Ma, C., Yang, X., Yang, M.H.: Vidtome: Video token merging for zero-shot video editing. arXiv preprint arXiv:2312.10656 (2023)
20. Liang, F., Wu, B., Wang, J., Yu, L., Li, K., Zhao, Y., Misra, I., Huang, J.B., Zhang, P., Vajda, P., Marculescu, D.: Flowvid: Taming imperfect optical flows for consistent video-to-video synthesis. ArXiv **abs/2312.17681** (2023), `https://api.semanticscholar.org/CorpusID:266690780`
21. Lim, J.H., Ye, J.C.: Geometric gan. arXiv preprint arXiv:1705.02894 (2017)
22. Lin, S., Liu, B., Li, J., Yang, X.: Common diffusion noise schedules and sample steps are flawed. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5404–5411 (2024)
23. Ma, H., Mahdizadehaghdam, S., Wu, B., Fan, Z., Gu, Y., Zhao, W., Shapira, L., Xie, X.: Maskint: Video editing via interpolative non-autoregressive masked transformers. arxiv preprint (2023)
24. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
25. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14297–14306 (2023)
26. Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., et al.: Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193 (2023)
27. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: The Eleventh International Conference on Learning Representations (2023), `https://openreview.net/forum?id=FjNys5c7VyY`
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
29. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: 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. pp. 22500–22510 (2023)
30. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation. ArXiv **abs/2311.17042** (2023), `https://api.semanticscholar.org/CorpusID:265466173`
31. Sauer, A., Lorenz, D., Blattmann, A., Rombach, R.: Adversarial diffusion distillation (2023)

32. Sheynin, S., Polyak, A., Singer, U., Kirstain, Y., Zohar, A., Ashual, O., Parikh, D., Taigman, Y.: Emu edit: Precise image editing via recognition and generation tasks (2023)
33. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
34. Wang, W., Xie, k., Liu, Z., Chen, H., Cao, Y., Wang, X., Shen, C.: Zero-shot video editing using off-the-shelf image diffusion models. arXiv preprint arXiv:2303.17599 (2023)
35. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942 (2023)
36. Wu, B., Chuang, C.Y., Wang, X., Jia, Y., Krishnakumar, K., Xiao, T., Liang, F., Yu, L., Vajda, P.: Fairy: Fast parallelized instruction-guided video-to-video synthesis. ArXiv **abs/2312.13834** (2023), https://api.semanticscholar.org/CorpusID:266435967
37. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7623–7633 (2023)
38. Wu, J.Z., Li, X., Gao, D., Dong, Z., Bai, J., Singh, A., Xiang, X., Li, Y., Huang, Z., Sun, Y., et al.: Cvpr 2023 text guided video editing competition. arXiv preprint arXiv:2310.16003 (2023)
39. Yan, W., Brown, A., Abbeel, P., Girdhar, R., Azadi, S.: Motion-conditioned image animation for video editing. ArXiv **abs/2311.18827** (2023), https://api.semanticscholar.org/CorpusID:265506378
40. Yang, S., Zhou, Y., Liu, Z., , Loy, C.C.: Rerender a video: Zero-shot text-guided video-to-video translation. In: ACM SIGGRAPH Asia 2023 Conference Proceedings (2023)
41. Yang, S., Mou, C., Yu, J., Wang, Y., Meng, X., Zhang, J.: Neural video fields editing. arXiv preprint arXiv:2312.08882 (2023)
42. Yatim, D., Fridman, R., Tal, O.B., Kasten, Y., Dekel, T.: Space-time diffusion features for zero-shot text-driven motion transfer. arXiv preprint arXiv:2311.17009 (2023)
43. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3836–3847 (October 2023)