Structure and Content-Guided Video Synthesis with Diffusion Models

Patrick Esser  Johnathan Chiu  Parmida Atighehchian
Jonathan Granskog  Anastasis Germanidis

Runway
https://research.runwayml.com/gen1

Figure 1. Guided Video Synthesis We present an approach based on latent video diffusion models that synthesizes videos (top and bottom) guided by content described through text (top) or images (bottom) while keeping the structure of an input video (middle).

Abstract

Text-guided generative diffusion models unlock powerful image creation and editing tools. Recent approaches that edit the content of footage while retaining structure require expensive re-training for every input or rely on error-prone propagation of image edits across frames.

In this work, we present a structure and content-guided video diffusion model that edits videos based on descriptions of the desired output. Conflicts between user-provided content edits and structure representations occur due to insufficient disentanglement between the two aspects. As a solution, we show that training on monocular depth estimates with varying levels of detail provides control over structure and content fidelity. A novel guidance method, enabled by joint video and image training, exposes explicit control over temporal consistency. Our experiments demonstrate a wide variety of successes; fine-grained control over output characteristics, customization based on a few reference images, and a strong user preference towards results by our model.

1. Introduction

Demand for more intuitive and performant video editing tools has increased as video-centric platforms have been popularized. But editing in the format is still complex and time-consuming due to the temporal nature of video data. State-of-the-art machine learning models have shown great promise in improving editing workflows.

Generative approaches for image synthesis recently experienced a rapid surge in quality and popularity due to the introduction of powerful diffusion models trained on large-scale datasets. Text-conditioned models, such as DALL-E 2 [34] and Stable Diffusion [38], enable novice users to generate detailed imagery given only a text prompt as input. Latent diffusion models especially enable efficient methods for producing imagery via synthesis in a perceptually compressed space.

Motivated by this progress, we investigate generative models suited for interactive applications in video editing.
Current methods repurpose existing image models by either propagating edits with approaches that compute explicit correspondences [5] or by finetuning on each individual video [63]. We aim to circumvent expensive per-video training and correspondence calculation to achieve fast inference for arbitrary videos.

We propose a controllable structure and content-aware latent video diffusion model trained on a large-scale dataset of uncaptioned videos and images. We opt to represent structure with monocular depth estimates, and content with embeddings predicted by a pre-trained neural network. Our approach offers several powerful modes of control. First, we train our model such that the content of inferred videos, e.g. their appearance or style, match user-provided images or text prompts (Fig. 1). Second, we vary the fidelity of the structure representation during training to allow selecting the strength of the structure preservation at test-time. Finally, we also adjust the inference process via a custom guidance method, inspired by classifier-free guidance, to enable control over temporal consistency.

In summary, we present the following contributions:

• We extend latent diffusion models to video generation by introducing temporal layers into a pre-trained image model and by joint training on images and videos.

• We present a structure and content-aware model that edits videos given example images or text. Our method does not require per-video training or pre-processing.

• We demonstrate full control over temporal, content and structure consistency. We show for the first time that joint image-video training enables control over temporal stability. And, training on varying levels of detail in the structure representation allows choosing the desired level of preservation during inference.

• We show that our approach is preferred over several other approaches in a user study. We further improve the accuracy of previously unseen content by finetuning on a small set of images of the desired subject.

2. Related Work

Controllable video editing and media synthesis is an active area of research. In this section, we review prior work in related areas and connect our method to these approaches.

Unconditional video generation Generative adversarial networks (GANs) [12] can learn to synthesize videos based on specific training data [59, 45, 1, 56]. These methods often struggle with stability during optimization, and produce fixed-length videos [59, 45] or longer videos where artifacts accumulate over time [50]. [6] synthesize longer videos using a GAN with a better encoding of the time axis. Autoregressive transformers have also been proposed for unconditional video generation [11, 64]. Our focus is on providing user control over the synthesis process whereas these approaches are limited to sampling random content resembling their training distribution.

Diffusion models for image synthesis Diffusion models (DMs) [51, 53] can synthesize detailed media in many formats, such as images [34, 38], 3d shapes [66] and animations [54]. Many works improve diffusion-based image synthesis by changing the parameterization [14, 27, 46], introducing advanced sampling methods [52, 24, 22, 47, 20], designing more powerful architectures [3, 15, 57, 30], or conditioning on additional information [25]. Text-conditioning, based on embeddings from CLIP [32] or T5 [33], has become a particularly powerful approach for providing artistic control over model output [44, 28, 34, 3, 65, 10]. Latent diffusion models (LDMs) [38] perform diffusion in a compressed latent space reducing memory requirements and runtime. Our video model is an LDM trained simultaneously on videos and images.

Diffusion models for video synthesis Recently, diffusion models, masked generative models and autoregressive models have been applied to text-conditioned video synthesis [17, 13, 58, 67, 18, 49]. Similar to [17] and [49], we extend image synthesis diffusion models to video generation by introducing temporal connections into a pre-existing image model. Our model edits videos rather than synthesizing them from scratch. We demonstrate through a user study that our model with explicit conditioning over structure is preferred over other related approaches.

Video translation and propagation Image-to-image translation models, such as pix2pix [19, 62], can modify each frame in a video individually. This produces temporal inconsistencies as the time axis is ignored. Accounting for temporal or geometric information, such as flow, can increase consistency across frames when repurposing image synthesis models [42, 9]. We can extract such structural information to aid our spatio-temporal LDM in text- and image-guided video synthesis. Many generative adversarial methods, such as vid2vid [61, 60], leverage this type of input to guide synthesis.

Video style transfer takes a reference style image and statistically applies its style to an input video [40, 8, 55]. In contrast, our method edits both style and content while preserving the structure of a video instead of matching feature statistics only. Text2Live [5] allows editing input videos using text prompts by decomposing a video into neural layers [21]. Once available, a layered video representation [37] provides consistent propagation across frames. SinFusion [29] and Tune-a-Video [63] use diffusion models to edit videos but require per-video training. This limits the practicality of the approaches in creative tools. We opt to instead train our model on a large-scale dataset permitting inference on any video without individual training.
3. Method

For our purposes, it will be helpful to think of a video in terms of its content and structure. By structure, we refer to characteristics describing geometry and dynamics, e.g. shapes and locations of subjects as well as their temporal changes. We define content as features describing the appearance and semantics of the video, such as the colors and styles of objects and the lighting. The goal of our model is to edit the content of a video while retaining its structure.

To achieve this, we learn a generative model \( p(x|s, c) \) of videos \( x \) conditioned on representations of structure \( s \) and content \( c \). We infer the shape representation \( s \) from an input video, and modify it based on a text prompt \( c \) describing the edit. First, we describe our realization of the generative model as a conditional latent video diffusion model and, then, we describe our choices for shape and content representations. Finally, we discuss the optimization process of our model. See Fig. 2 for an overview.

3.1. Latent diffusion models

**Diffusion models** Diffusion models [51] learn to reverse a fixed forward diffusion process, which is defined as

\[
q(x_t|x_{t-1}) := \mathcal{N}(x_t, \sqrt{1 - \beta_t} x_{t-1}, \beta_t I) .
\]

Normally-distributed noise is slowly added to each sample \( x_{t-1} \) to obtain \( x_t \). The forward process models a fixed Markov chain and the noise is dependent on a variance schedule \( \beta_t \) where \( t \in \{1, \ldots, T\} \), with \( T \) being the total number of steps in our diffusion chain, and \( x_0 := x \).

**Learning to Denoise** The reverse process is defined according to the following equation with parameters \( \theta \)

\[
p_\theta(x_0) := \int p_\theta(x_{0:T})dx_{1:T} \quad (2)
\]

\[
p_\theta(x_{0:T}) = p(x_T) \prod_{t=1}^T p_\theta(x_{t-1}|x_t), \quad (3)
\]

\[
p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}, \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) . \quad (4)
\]

Using a fixed variance \( \Sigma_\theta(x_t, t) \), we are left learning the means of the reverse process \( \mu_\theta(x_t, t) \). Training is typically performed via a reweighted variational bound on the maximum likelihood objective, resulting in a loss

\[
L := \mathbb{E}_{t,q} \lambda_t \| \mu_t(x_t, x_0) - \mu_\theta(x_t, t) \|^2 , \quad (5)
\]

where \( \mu_t(x_t, x_0) \) is the mean of the forward process posterior \( q(x_{t-1}|x_t, x_0) \), which is available in closed form [14].

**Parameterization** The mean \( \mu_\theta(x_t, t) \) is then predicted by a UNet architecture [39] that receives the noisy input \( x_t \) and the diffusion timestep \( t \) as inputs. Other parameterizations and weightings, such as \( \epsilon \) [14] and \( v \)-parameterizations [46], can significantly improve sample quality compared to directly predicting the mean. Similar to [13], we found that \( v \)-parameterization improves color consistency thus all our experiments use it (see supp. material for more details).

**Latent diffusion** Latent diffusion models [38] (LDMs) take the diffusion process into the latent space. This provides improved separation between compressive and generative learning phases of the model. Specifically, LDMs use an autocoder where an encoder \( \mathcal{E} \) maps input data \( x \) to a lower dimensional latent code according to \( z = \mathcal{E}(x) \) while a decoder \( D \) converts latent codes back to the input space such that perceptually \( x \approx D(\mathcal{E}(x)) \).

Our encoder downsamples RGB-images \( x \in \mathbb{R}^{3 \times H \times W} \) by a factor of eight and outputs four channels, resulting in a latent code \( z \in \mathbb{R}^{4 \times H/8 \times W/8} \). Thus, the diffusion UNet operates on a much smaller representation which significantly

![Figure 2. Overview: During training (left), input videos \( x \) are encoded to \( z_0 \) with a fixed encoder \( \mathcal{E} \) and diffused to \( z_t \). We extract a structure representation \( s \) by encoding depth maps obtained with MiDaS, and a content representation \( c \) by encoding one of the frames with CLIP. The model then learns to reverse the diffusion process in the latent space, with the help of \( s \), which gets concatenated to \( z_t \), as well as \( c \), which is provided via cross-attention blocks. During inference (right), the structure \( s \) of an input video is provided in the same manner. To specify content via text, we convert CLIP text embeddings to image embeddings via a prior.](image-url)
Diffusion models are Conditional Diffusion Models

3.3. Representing Content and Structure

The frame index into temporal transformer blocks. We also input learnable positional encodings of transformer block, we also include one temporal 1D trans- after each 2D convolution. Similarly, after each spatial 2D each residual block, we introduce one temporal convolution parameters to benefit from better generalization obtained by architecture must account for temporal relationships. We extending an image-based UNet improving runtime and memory efficiency. The latter is crucial for video modeling where the additional time axis increases memory costs.

3.2. Spatio-temporal Latent Diffusion

To correctly model a distribution over video frames, the architecture must account for temporal relationships. We also want to jointly learn an image model with shared parameters to benefit from better generalization obtained by training on large-scale image datasets.

To achieve this, we extend an image architecture by introducing temporal layers, which are only active for video inputs. All other layers are shared between the image and video model. The autoencoder remains fixed and processes each frame in a video independently.

The UNet consists of two main building blocks: Resid- ual blocks and transformer blocks (see Fig. 3). Similar to [17, 49], we extend them to videos by adding both 1D convolutions across time and 1D self-attentions across time. In each residual block, we introduce one temporal convolution after each 2D convolution. Similarly, after each spatial 2D transformer block, we also include one temporal 1D transformer block, which mimics its spatial counterpart along the time axis. We also input learnable positional encodings of the frame index into temporal transformer blocks.

3.3. Representing Content and Structure

Conditional Diffusion Models Diffusion models are well-suited to modeling conditional distributions such as $p(x|s, c)$. The forward process $q$ remains unchanged while the conditioning variables $s, c$ become additional inputs to the model.

Our goal is to edit a video based on a text prompt describing the desired output. We choose to train on un- captioned video data due to the lack of large-scale paired video-text datasets of similar quality as image datasets like CLIP. To support text-based editing at inference, we train a prior model that allows sampling image embeddings from text embeddings [35, 49].

Content Representation We utilize CLIP [32] to infer a content representation from both text inputs $t$ and video inputs $x$ similar to previous works [35, 3]. CLIP embeddings are a promising content representation as they are more sensitive to semantic and stylistic properties while being more invariant towards geometric attributes [34]. During training, we encode a random frame in each input video with CLIP. To support text-based editing at inference, we train a prior model that allows sampling image embeddings from text embeddings [35, 49].

Structure Representation We need a representation that provides adequate separation between structure and content. We find that depth estimates extracted from input video frames provide the desired properties as they encode significantly less content information compared to simpler structure representations, such as edge filters which also encode textural properties. Still, depth maps reveal the silhouettes of objects which can prevent content edits involving changes in object shape.

To offer control over the amount of structure to preserve, we propose to train a model on structure representations with varying amounts of information. In particular, we blur depth estimates given a parameter $t_s$. During training, $t_s$ is randomly sampled between 0 and $T_s$. The parameter can then be controlled at inference to achieve different editing effects (see Fig. 10).

While depths map work well for our usecase, our approach generalizes to other geometric guidance features or combinations of features that might be more helpful for other specific applications. For example, models focusing on human video synthesis might benefit from estimated poses or face landmarks.

Conditioning Mechanisms We account for the different characteristics of our content and structure representations with two different conditioning mechanisms. Since structure represents a significant portion of the spatial information of video frames, we use concatenation for conditioning to make effective use of this information. In contrast, attributes described by the content representation are not tied to particular locations. Hence, we leverage cross-attention which can effectively transport this information to any position.

We use the spatial transformer blocks of the UNet architecture for cross-attention conditioning. Each contains two attention operations, where the first one performs a spatial self-attention and the second one a cross attention with keys and values computed from the CLIP image embedding.

In contrast, during inference, structure $s$ and content $c$ are derived from an input video $y$ and from a text prompt $t$ respectively. An edited version $x$ of $y$ is obtained by sampling the generative model conditioned on $s(y)$ and $c(t)$:

$$z \sim p_{\theta}(z|s(y), c(t)), \quad x = D(z). \quad (7)$$

Content Representation

We utilize CLIP [32] to infer a content representation from both text inputs $t$ and video inputs $x$ similar to previous works [35, 3]. CLIP embeddings are a promising content representation as they are more sensitive to semantic and stylistic properties while being more invariant towards geometric attributes [34]. During training, we encode a random frame in each input video with CLIP. To support text-based editing at inference, we train a prior model that allows sampling image embeddings from text embeddings [35, 49].

Structure Representation

We need a representation that provides adequate separation between structure and content. We find that depth estimates extracted from input video frames provide the desired properties as they encode significantly less content information compared to simpler structure representations, such as edge filters which also encode textural properties. Still, depth maps reveal the silhouettes of objects which can prevent content edits involving changes in object shape.

To offer control over the amount of structure to preserve, we propose to train a model on structure representations with varying amounts of information. In particular, we blur depth estimates given a parameter $t_s$. During training, $t_s$ is randomly sampled between 0 and $T_s$. The parameter can then be controlled at inference to achieve different editing effects (see Fig. 10).

While depths map work well for our usecase, our approach generalizes to other geometric guidance features or combinations of features that might be more helpful for other specific applications. For example, models focusing on human video synthesis might benefit from estimated poses or face landmarks.

Conditioning Mechanisms

We account for the different characteristics of our content and structure representations with two different conditioning mechanisms. Since structure represents a significant portion of the spatial information of video frames, we use concatenation for conditioning to make effective use of this information. In contrast, attributes described by the content representation are not tied to particular locations. Hence, we leverage cross-attention which can effectively transport this information to any position.

We use the spatial transformer blocks of the UNet architecture for cross-attention conditioning. Each contains two attention operations, where the first one performs a spatial self-attention and the second one a cross attention with keys and values computed from the CLIP image embedding.

$$z \sim p_{\theta}(z|s(y), c(t)), \quad x = D(z). \quad (7)$$
Figure 4. **Temporal Control**: By training image and video models jointly, we obtain explicit control over the temporal consistency of edited videos via a temporal guidance scale $\omega_t$. On the left, frame consistency measured via CLIP cosine similarity of consecutive frames increases monotonically with $\omega_t$, while mean squared error between frames warped with optical flow decreases monotonically. On the right, lower scales (0.5 in the middle row) achieve edits with a "hand-drawn" look, whereas higher scales (1.5 in the bottom row) result in smoother results. Top row shows the original input video, the two edits use the prompt "pencil sketch of a man looking at the camera".

<table>
<thead>
<tr>
<th>Prompt</th>
<th>Driving Video (top) and Result (bottom)</th>
</tr>
</thead>
<tbody>
<tr>
<td>a man using a laptop inside a train, anime style</td>
<td>![Images of a man using a laptop inside a train in anime style]</td>
</tr>
<tr>
<td>a woman and man take selfies while walking down the street, claymation</td>
<td>![Images of a woman and man taking selfies while walking down the street in claymation style]</td>
</tr>
<tr>
<td>kite-surfer in the ocean at sunset</td>
<td>![Images of a kite-surfer in the ocean at sunset]</td>
</tr>
<tr>
<td>alien explorer hiking in the mountains</td>
<td>![Images of an alien explorer hiking in the mountains]</td>
</tr>
</tbody>
</table>

Figure 5. Our approach enables a wide range of video edits, including changes to animation styles such as anime or claymation, changes of environment such as time of day, and changing characters such as humans to aliens.
To condition on structure, we first estimate depth maps for all input frames using the MiDaS DPT-Large model [36]. We then apply \( t_s \) iterations of blurring and downsampling to the depth maps, where \( t_s \) controls the amount of structure to preserve. We resample the perturbed depth map to the resolution of the RGB-frames and encode it using \( \mathcal{E} \). This latent representation of structure is concatenated with the input \( z_t \) given to the UNet. We also input four channels containing a sinusoidal embedding of \( t_s \).

**Sampling** While Eq. (2) provides a direct way to sample from the trained model, many other sampling methods [52, 24, 22] require only a fraction of the number of diffusion timesteps to achieve good sample quality. We use DDIM [52] throughout our experiments. Furthermore, classifier-free diffusion guidance [16] significantly improves sample quality. For a conditional model \( \mu_\theta(x_t, t, c) \), this is achieved by training the model to also perform unconditional predictions \( \mu_\theta(x_t, t, \emptyset) \) and then adjusting predictions during sampling according to

\[
\tilde{\mu}_\theta(x_t, t, c) = \mu_\theta(x_t, t, \emptyset) + \omega (\mu_\theta(x_t, t, c) - \mu_\theta(x_t, t, \emptyset))
\]

where \( \omega \) is the guidance scale that controls the strength. Based on the intuition that \( \omega \) extrapolates the direction between an unconditional and a conditional model, we apply this idea to control temporal consistency of our model. Specifically, since we are training both an image and a video model with shared parameters, we can consider predictions by both models for the same input. Let \( \mu_\theta(z_t, t, c, s) \) denote the prediction of our video model, and let \( \mu_\theta^u(z_t, t, c, s) \) denote the prediction of the image model applied to each frame individually. Taking classifier-free guidance for \( c \) into account, we then adjust our prediction according to

\[
\tilde{\mu}_\theta(z_t, t, c, s) = \mu_\theta^u(z_t, t, \emptyset, s) + \omega_t (\mu_\theta(x_t, t, \emptyset, s) - \mu_\theta^u(x_t, t, \emptyset, s)) + \omega (\mu_\theta(x_t, t, c, s) - \mu_\theta(x_t, t, \emptyset, s))
\]

where \( \omega_t \) is the guidance scale that controls the strength.

3.4. Optimization

We train on an internal dataset of 240M images and 6.4M video clips. We use image batches of size 9216 with resolutions of \( 320 \times 320, 384 \times 320 \) and \( 448 \times 256 \), as well as the same resolutions with flipped aspect ratios. We sample image batches with a probability of 12.5%. For videos, we use batch size 1152 and 8 frames from each video sampled four frames apart with a resolution of \( 448 \times 256 \).

We train our model in multiple stages. First, we initialize model weights based on a pretrained text-conditional latent diffusion model [38]. We change the conditioning from CLIP text embeddings to CLIP image embeddings and fine-tune for 15k steps on images only. Afterwards, we introduce temporal connections as described in Sec. 3.2 and train jointly on images and videos for 75k steps. We then add conditioning on structure \( s \) with \( t_s \equiv 0 \) fixed and train for 25k steps. Finally, we resume training with \( t_s \) sampled uniformly between 0 and 7 for another 10k steps.

4. Results

To evaluate our approach, we use videos from DAVIS [31] and various stock footage. To automatically create edit prompts, we first run a captioning model [23] to obtain a description of the original video content. We then use GPT-3 [7] to generate edited prompts.

4.1. Qualitative Results

We demonstrate that our approach performs well on a number of diverse inputs (see Fig. 5). Our method handles a large variety of footage, such as landscapes and close-ups, and diverse camera motion without any explicit tracking of the input. Our depth-based structure representation combined with large-scale image-video joint training enable strong generalization and powerful editing capabili-

---

Footnotes:

ties. For example, we can produce various animation styles, changes in time of day, and more complex changes of subject, such as turning a hiker into an alien (see Fig. 5). Please see the supplementary material for more results.

Using CLIP image embeddings as the content representation allows users to specify content through images. As an example application, we demonstrate character replacement in Fig. 9. For every video in a set of six videos, we re-synthesize it five times, each time providing a single content image taken from another video in the set. We can retain content characteristics with $t_s = 3$ despite large differences in their pose and shape.

Lastly, we illustrate the use of masked video editing in Fig. 8, where the model predicts everything outside the masked area(s) while retaining the original content inside the masked area. Notably, this technique resembles inpainting with diffusion models [43, 25].

4.2. User Study

We benchmark against Text2Live [5], a recent approach for text-guided video editing that employs layered neural atlases [21]. As a baseline, we compare against SDEdit [26] in two ways; per-frame generated results and a first-frame result propagated by a few-shot video stylization method [55] (IVS). We also include two depth-based versions of Stable Diffusion; one trained with depth-conditioning [2] and one that retains past results based on depth estimates [9]. We also include an ablation: applying SDEdit to our video model trained without conditioning on a structure representation (ours, $\sim s$).

We judge the success of our method qualitatively based on a user study. We run the user study using Amazon Mechanical Turk (AMT) on an evaluation set of 35 representative video editing prompts. For each example, we ask 5 annotators to compare faithfulness to the video editing prompt (“Which video better represents the provided edited caption?”) between a baseline and our method, presented in random order, and use a majority vote for the final result.

The results can be found in Fig. 7. Across all compared methods, results from our approach are preferred roughly 3 out of 4 times. A visual comparison among the methods can be found in the supplementary. We observe that SDEdit is sensitive to the editing strength. Low values fail to achieve the desired editing effect whereas high values change the structure of the input. Even with a fixed seed, both style and structure can change in unnatural ways between frames as their relationship is ignored by image-based approaches. Propagation of SDEdit outputs (IVS) leads to more consistent results but often introduces propagation artifacts especially with large motion. Depth-SD produces accurate, structure-preserving edits for individual frames but frames are inconsistent across time. The outputs of Text2Live tend to be temporally smooth due to its reliance on Layered Neural Atlases [21], but it often produces edits that represent the edit prompt inaccurately. A direct comparison with Text2Live is difficult as it requires input masks and separate edit prompts for foreground and background. In addition, computing a neural atlas takes about 10 hours whereas our approach requires approximately a minute.

4.3. Quantitative Evaluation

We quantify trade-offs between frame consistency and prompt consistency with the following two metrics.

Frame consistency We compute CLIP image embeddings on all frames of output videos and report the average cosine similarity between all pairs of consecutive frames.

Prompt consistency We compute CLIP image embeddings on all frames of output videos and the CLIP text embedding of the edit prompt. We report average cosine similarity between text and image embedding over all frames.

Fig. 6 shows the results of each model using our frame consistency and prompt consistency metrics. Our model tends to outperform the baseline models in both aspects (placed higher in the upper-right quadrant of the graph). We also notice a slight tradeoff with increasing the strength parameters in the baseline models: larger strength scales implies higher prompt consistency at the cost of lower frame consistency. Increasing the temporal scale ($\omega_t$) of our model...
results in higher frame consistency but lower prompt consistency. We also observe that an increased structure scale ($t_s$) results in higher prompt consistency as the content becomes less determined by the input structure.

4.4. Customization

Customization of image models enables generation of previously unseen content, such as specific people or styles, based on a small dataset used for finetuning [41]. We finetune our depth-conditioned latent video diffusion model on a set of 15-30 images and produce videos containing the desired subject. Half of the batch elements are of the subject and the other half belong to the original training dataset.

Fig. 10 shows an example with different numbers of customization steps as well as different levels of structure preservation $t_s$. Customization improves fidelity to the style and appearance of the character. In combination with higher $t_s$ values, accurate animations are possible despite using a driving video of a person with different characteristics.

5. Conclusion

Our latent video diffusion model synthesizes new videos given structure and content information. We ensure structural consistency by conditioning on depth estimates while content is controlled with images or natural language. Temporal layers and joint image-video training achieve stable results across frames. A novel guidance method, inspired by classifier-free guidance, allows for control over temporal consistency. By training on depth maps with varying degrees of detail, we can adjust the level of structure preservation. This, together with model customization, improves content fidelity. Our quantitative evaluation and user study show that our method is preferred over related approaches. Future works should investigate other conditioning data, such as facial landmarks and pose estimates, and additional 3d-priors to improve generated results. Our model is intended for creative applications in content creation, but we realize the risks of dual-use and hope that further work will be aimed at combating abuse of generative models.
References


[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022. 1, 2, 4


[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 1, 2, 3, 6


[50] Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2, 2021. 2

of Machine Learning Research, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. 2, 3


[65] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagöl Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022. 2
