# 1 Rebuttal

## 1.1 Formatting

We received some constructive feedback on how we can improve the formatting of our paper including the arrangement of figures, the removing of citations within the abstract, and repositioning of some subsections for more literary flow. To address these concerns, we rid of all citations in the abstract and made proofreading edits throughout the paper. Furthermore, we consolidate the related works section into short paragraphs as opposed to many subsections. We also move some of our extraneous figures to the appendix in order to make the contents of the report more concise. For better understanding of our methodology, we provide pseudo code instead of raw Python code that details our training algorithms used for each technique utilized. We specifically rearrange subsections describing the experimental results of the Tune-A-Video technique to be adjacent in order to promote continuity of content (Section 5).

## 1.2 Paper Contents

We first address the requests we received about our background on Denoising Diffusion Probabilistic Models (DDPMs). We believe that the technique description is an important element to keep in this paper as-is due to the fundamental importance of the idea with regards to the entire paper as a whole. Most of the techniques mentioned in Section 3 such as Tune-A-Video, Control-A-Video, ControlNet, Uni-ControlNet, and StableVideo all base their methodology on the basic principle of the DDPM. Since our work seeks to experiment with these techniques and extend them, we believe the background information on DDPMs to be paramount to this paper even if the concepts are already well-known.

Some feedback we received for our paper pertains to mentioning experimentation with the Tune-A-Video technique with a Stable Diffusion 2.1 model as well as mentioning unreleased models such as Stable Diffusion 3 and SORA as possible areas of study and experimentation. While the criticism is against including such statements, we believe that because this paper has evolved into an overall study of generative text-to-video model architecture, it would be imperative to include such information within the report. Since we aim for this paper to provide an overview of all experimentation efforts made to evolve the techniques mentioned in Section 3, it would be reasonable for us to mention all the methods experimented with.

A large portion of the criticism received is regarding the lack of novelty of the techniques proposed within this paper. The paper, as it currently stands, only serves as an overview of existing methods of generating coherent text-to-video models, with some achieving output controlled by additional conditions while others achieving consistency of foreground and background objects across frames. We also mention all attempted efforts of training a neural network model that is capable of achieving both qualities at the same time. Throughout the semester, we encountered issues with getting our base methods working, having a consistent research direction, and coming up with something truly novel. When we explored new research objectives, we realized that there were already similar works detailing the same approach. We then read the related literature to understand the respective methodology and attempt to propose some new research objective to see if would could implement an improvement. As we discovered more existing works, it became challenging to understand the new methodologies, and as a result we spent a significant amount of time further reviewing literature rather than running experiments. We then eventually reached a point where we decided it was better to thoroughly investigate existing methods rather than coming up with a completely new approach. Specifically, our challenges can be listed in chronological order:

- We first decide to develop a novel architecture that generate videos with consistent scenery.
- We realize this has already been done by the Tune-A-Video technique. We respond by developing a novel method of adding conditional control to the output of a text-to-video model to further stabilize background and foreground objects.
- We realize this has already been done by the Control-A-Video and Neural Layered Atlas techniques. We respond by developing novel improvements that make training consistent text-to-image models more efficiently by borrowing ideas from the Uni-ControlNet method.
- As the deadline came closer, we realized we no longer had the time to understand all related literature. We decided to conduct experiments with our baseline techniques instead.

- However, based on feedback we were able to produce some additional results using both some suggested methods as well as methods we did not have time for prior. These new additions can be found highlighted in the results section below.

Due to these difficulties, our paper evolved to become more of a report of our process in investigating the various state of the art methods as well as details about all of the approaches we attempted based on this knowledge. Overall, while we acknowledge that this paper does not demonstrate amazing quantitative results, we believe that the insights gained from our investigation are valuable and do demonstrate several potential novel approaches to video generation.

### 1.3 Experimentation and Approaches

We received several comments questioning the rationale of some of our approaches and experiments. While we believe we have already explained our choices for choosing methods like Uni-ControlNet and NLA, we acknowledge that due to the organization of the paper as mentioned earlier it may have been unclear where these explanations are. So, we have made sure to more clearly state motivations and emphasize our experimentation choices.

Additionally, many comments suggested that we test some of the approaches we listed in further works such as optical flow and frame interpolation. While we did test some of these methods prior to the initial submission (but omitted results do to lack therof), in response to this feedback we ran additional experiments testing different combinations of new methods. The two most promising results are highlighted in a new portion of the results section.

Next, we received several comments inquiring about our rationale for choosing Neural Layered Atlases as an approach, as well as questioning why the experiment failed. First, as mentioned in the related works section as well as our approaches section, we picked NLA as a potential solution due to its proven ability to preserve high spatio-temporal consistency in video editing applications. A primary problem we identified in current video generation is the lack of consistency of output especially when there are occluded objects present. The NLA technique excels in this application specifically, because by separating the objects in the video into independent layer representations, the model is able to alter the motion of specific objects without changing that of others unrealistically. As for a potential further explanation for the failure of our experimentation with NLA, we still believe it is mostly due to computational efficiency. Given sufficient compute, we would implement NLA video generation in a similar way to the Uni-ControlNet implementation we used. Create a simplified version of a neural atlas based off of the initial frame, then generate subsequent frames while minimizing deviation from previous atlas layers.

Finally, to address the lack of quantifiable or tangible results, we believe that unfortunately our chosen area of video generation is very challenging to achieve quantifiable good results in primarily due to constraints such as computing power. We did consider standard metrics such as FVD and CLIP scores. However, we realize that these results likely would not be significant, since throughout our experimentation we have not been able to produce videos longer than 5 seconds or over 30 frames per second. As such, low scores on metrics would not be very accurate to the actual quality of our results since it would be uncertain whether the cause was just due to our lack of resources anm time.

# Investigation of Methods to Improve Generative Video Consistency and Control

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

With the rise of Generative AI, recent improvements in diffusion techniques have been developed to generate art contextually accurate to user input. However, video output from Stable Diffusion models still seem to have sudden differences between neighboring frames and can be easily differentiated from videos taken in real life at times. Subjects and background environments in generated videos are prone to suddenly shifting appearance, making the video more identifiable as a result of AI generation. In particular, we found that even state-of-the art video generation and editing models struggled when occlusion was present. We propose a project to find a solution to improve the smoothness and consistency of video generation network output by using various approaches such as ControlNet and neural layered atlases. Additionally, we intend to combine newer concepts like Uni-ControlNet with existing text to video models in order to enable even better control of video results.

## 2 Introduction

Using diffusion models like Stable Diffusion, users can create videos from input text, images, or videos. Each frame of the output video is generated independently using the diffusion model, which results in slight differences between frames. Current methods to improve the smoothness of diffusion generated videos rely on techniques focusing on pixel motion information [7] or Recurrent All-Pairs Field Transforms (RAFT) for optical flow [13].

These methods show good results when applied to generate videos, but are inefficient because they often require per-pixel calculations and additional processing. Current generative video systems already struggle from computational limits; e.g. generating a 10 second video at just 20 fps would involve using diffusion text-to-image to create over 200 images. Even top open source text to video models like VideoCrafter [3] took over 10 minutes to generate a 2 second video (running on one RTX A4000 GPU). Earlier works like ControlNet [15] have been proven to allow much finer control over text-to-image generation, and don't add much to the runtime complexity since they can be pre-trained for various controls (like edges or depth for example). We hope to use implementations of pre-trained ControlNets such as Uni-ControlNet to allow better control over the various objects of a scene without calculating pixel motion every frame.

Additionally, several state-of-the-art generative video editing frameworks have employed techniques to effectively separate the objects and background of a scene. This also involves pre-trained models that break apart the frames of a video, and demonstrate strong ability to preserve spatio-temporal consistency across the video despite edits to the appearance of the frames.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

Taking inspiration from these previous methods, we aim incorporate them in video generation architecture itself. In applying these methods in this novel way, we hope to further improve control and consistency of AI-generated videos while improving efficiency at the same time.

# 3 Related Works

**Tune-A-Video** is an approach that creates a T2V model by fine-tuning an existing T2I diffusion model [14]. This method continues to train pretrained T2I models with an additional spatial-temporal attention mechanism using structural guidance. As a result, videos generated from the final result can have temporal consistency without having to train a model from a large video dataset. **Control-A-Video** expands upon Tune-A-Video by adding control to the outputs generated by T2V models [4]. This technique aims to generate videos based on a sequence of control maps (depths, soft-edges, normals, segmentation masks, etc). Furthermore, this model uses two motion-adaptive noise initialization strategies to incorporate motion priors and a first-frame conditioned controller to include content priors. Resulting videos show an increased foreground and background consistency between frames.

**ControlNet** is a neural network architecture that enables the addition of conditioning controls to existing text-to-image diffusion models [15]. They can be trained on various spatial conditions such as Canny edges, segmentation maps, depths, and more. The ControlNet structure is applied to each encoder level in the U-net of the diffusion model to control the output of the model. **Uni-ControlNet** expands upon the ControlNet by handling different conditions within one single model and by supporting composable control [16]. This method utilizes a multi-scale condition injection strategy instead of injecting singular conditions directly into input noise. This technique proves superior to using N different ControlNets on N separate conditions since the Uni-ControlNet only uses local and global adapters, reducing the number of times the model needs to be fine-tuned to a constant value of two.

**Neural Layered Atlases** (NLA) is a method that unwraps an input video into a set of layered 2D atlases [10]. Each atlas is a representation of the appearance of an object or background throughout an entire video. This technique uses coordinate-based MLPs to map pixels into an atlas space, which are then optimized against reconstruction and regularization losses in order to preserve the integrity/realism of the video. By separating objects into separate interpretable layers, NLA allows users to make edits to the entire video by simply altering a single atlas. This method has been shown to improve temporal consistency as well as occlusion performance in video editing. **Text2LIVE** and **StableVideo** are both video editing frameworks that leverage pre-trained neural layered atlases of videos in order to produce consistent edits. While Text2LIVE uses only NLA to ensure consistency [1], StableVideo takes it one step further and utilizes an inter-frame propogation mechanism based on ControlNet in order to further preserve object consistency [2]. However, both of these frameworks require separately trained NLAs and an existing video.

# 4 Methodology

## 4.1 Preliminaries

**Denoising Diffusion Probabilistic Models (DDPMs)** is a generative technique that is trained on reversing a fixed forward Markov Chain $x_1, ..., x_T$ [6]. Assuming an image data distribution $x_0 \sim q(x_0)$, the Markov transition $q(x_t|x_{t-1})$ is defined as a Gaussian distribution:

$$q(x_t|x_{t-1}) \sim \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbb{I}), \quad t = 1, ..., T \tag{1}$$

where $\beta_t \in (0, 1)$ is the variance schedule. As a consequence:

$$q(x_t|x_0) \sim \mathcal{N}(\sqrt{(1-\beta_t)...(1-\beta_1)}x_0, (\beta_t)...(\beta_1)\mathbb{I}), \quad t = 1, ..., T \tag{2}$$

$$q(x_{t-1}|x_t, x_0) \sim \mathcal{N}(\hat{\mu_t}(x_t, x_0), \hat{\beta_t}\mathbb{I}), \quad t = 1, ..., T \tag{3}$$

where

$$\hat{\mu}_t(x_t, x_0) = \frac{1}{\sqrt{1-\beta_t}} \left( x_t - \frac{\beta_t}{\sqrt{1-((1-\beta_1)...(1-\beta_t))}} \epsilon \right) \tag{4}$$

and

$$\hat{\beta}_t = \frac{1-((1-\beta_1)..(1-\beta_{t-1}))}{1-((1-\beta_1)..(1-\beta_t))} \beta_t \tag{5}$$

and

$$\epsilon \sim \mathcal{N}(0, \mathbb{I}) \tag{6}$$

due to the Markov Property and Bayes' Rule. DDPMs accomplish this reversal at each step with a transition defined as:

$$p_\theta(x_{t-1}, x_t) \sim \mathcal{N}(\mu_\theta(x_t, t), \delta_t^2 \mathbb{I}), \quad t = 1, ..., T \tag{7}$$

where $\mu_\theta$ is the denoising autoencoder, with learnable parameters $\theta$, trained so that the reverse process is as close to possible to the forward process with the objective:

$$\min_\theta \; \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,\mathbb{I}), t} \left[ \frac{1}{2\delta_t^2} ||\hat{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)||^2 \right] \tag{8}$$

as derived by maximizing the variational lower bound of the log-likelihood optimization which has a term representing the KL-divergence between Gaussian distributions.

**Latent Diffusion Models (LDMs)** are newly introduced variants of DDPMs that use the same technique as described above within the latent space of an autoencoder [12]. The concept of LDMs is best described as two-fold. The first of which is an autoencoder optimized to minimize patch-wise loss on a large dataset of images with an encoder to compress input images into latent space and a decoder to reconstruct latent variables back to the approximate original input. The second part is a DDPM trained to denoise added to sampled values in the latent space.

### 4.2 Preliminary Work

In order to understand the state-of-the-art space for the area of video generation, we began by collecting the top text/image to video generation and video editing models to our knowledge (that offered public code bases). Initially, inspired by discussions about transformers and attention methods like (SWIN [9]), we aimed to improve consistency by augmenting the attention modules of the video models. In order to preserve a realistic transition between frames, we attempted to apply a shifted-window attention method between adjacent frames. However, a deeper dive into the code of these models revealed that some projects like Tune-A-Video and Control-A-Video had already implemented novel spatio-temporal attention modules that add time as a third dimension to create even more consistent video frames. Realizing this, we pivoted to finding novel applications or combinations of some of the most well-performing methods, aiming to achieve the benefits of all of these methods within one model.

### 4.3 Approaches

Throughout our investigation of novel methods to further improve video generation, we converged on two primary approaches. First, we experimented with applying the foundational ControlNet method (which adds controls to text-to-image generation) to video generation. This involved extensive testing of existing works that use this method in order to understand how the ControlNet architecture fits into the process of video generation. Second, we explored existing video editing models that generally aim to isolate various objects in a scene in order to preserve spatio-temporal consistency throughout the editing process. We tested various ways of incorporating these techniques into the video generation process itself in order to create a self-contained video generation framework that can achieve high consistency and control without post processing or external plug-ins.

**Uni-ControlNet for video generation**

The first off-the-shelf model we experimented with was Tune-A-Video [14]. This work fine tuned existing text-to-image models in order to convert them into text-to-video. Notably, Tune-A-Video utilizes a 3D U-Net that allows for more consistent results through applying a spatio-temporal attention mechanism. We first aimed to modify the Tune-A-Video pipeline in order to integrate existing ControlNet implementations, and therefore benefit from the consistency of Tune-A-Video while improving the control provided by ControlNet.

Through further literature review, we discovered Control-A-Video, which already incorporates ControlNet-like controls into a text-to-video generation model. They utilize a 3D ControlNet pipeline to apply the control maps across not only the individual frames but also across time. Thus, instead of trying to recreate Control-A-Video using Tune-A-Video we decided to instead try to improve upon Control-A-Video by leveraging the multi-control ability of Uni-ControlNet [16]. Uni-ControlNet enables composable multi-control by using local and global adapter representations to add many controls without impacting the efficiency of the process significantly. Using this method, we can apply multiple controls that improve consistency and realism at the same time. For example, we can use both Canny edge maps and Midas depth maps at the same time to ensure objects in the generated video not only follow a realistic movement pattern (using edges) but also don't morph into the background (depth).

We choose to focus our approach on incorporating Uni-ControlNet because of its potential to improve both consistency and quality simultaneously without increasing complexity significantly.

**Neural Layered Atlases (NLA)**

Through our investigation of the video editing system StableVideo [2], we learned about the concept of Neural Layered Atlases [10]. NLA can be pre-trained on the frames of a video in order to create atlas representations of the background and each foreground object respectively. NLA uses multi-layer perceptron networks to map pixels from pixel-space into atlas space. The atlas serves as a single representation of a "layer" or an object over all frames in the video. Particularly of interest to us, the training of an NLA representation uses a rigidity loss term to encourage pixel mappings to be locally rigid in 2D atlas space (through a Jacobian matrix of mapping M at each pixel p):

$$J_M = [M(p_x) - M(p)M(p_y) - M(p)] \in \mathbb{R}^{2x2} \tag{9}$$

where

$$p_x = (x + \Delta, y, t), p_y = (x, y + \Delta, t) \tag{10}$$

This rigidity in atlas space encourages better spatio-temporal consistency when frames are changed through the editing process, which allows changes to a single atlas layer to be automatically applied to the entire video. Inspired by this, we hope to incorporate atlas rigidity loss between frames during generation itself. By creating an atlas representation on the first frame, we could enforce a similar rigidity loss function to subsequent frames in order to discourage unrealistic perturbations.

# 5 Experimental Results

## 5.1 Tune-A-Video Testing

First, we tested the base functionality of Tune-A-Video in order to better understand the framework. We utilized the model to generate videos of various characters skiing in different art styles as shown in Figure 1. Then, we created a ComfyUI workflow that generates control maps for each frame of these videos, which resulted in Figure 2.

(a) Results after 100 epochs.

(b) Results after 500 epochs.

Figure 1: After feeding Tune-A-Video with a video of a man skiing, we generate videos of various characters skiing in different artstyles using different prompts. The prompts in order from left to right are: "mickey mouse is skiing on the snow", "spider man is skiing on the beach, cartoon style", "wonder woman, wearing a cowboy hat, is skiing", and "a man, wearing pink clothes, is skiing at sunset"

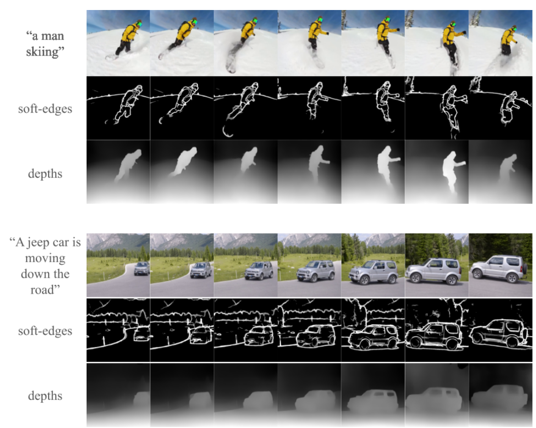



Figure 2: We generate control maps such as soft-edges and depths from sample videos to better control the Tune-A-Video model.

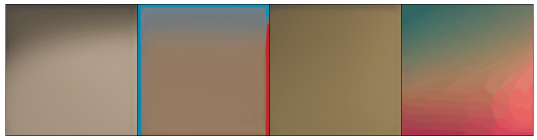
### 5.2 Additional results with Tune-A-Video (using SD 2.1)

During our testing of Tune-A-Video, we decided to try using Stable Diffusion 2.1 (as opposed to 2.0 which we used for our earlier results). However, when we ran the exact same prompt experiment of different characters skiing, we got extremely blurry results (see Figure 3).

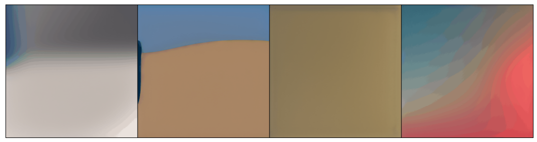
This was an interesting and unexpected result, because we could not understand why the new Stable Diffusion would cause this to fail. However, through further investigation we noticed that SDv2.1 has a new depth-guided control system that generates images based off the Midas depth map of the image input [11]. This component may make it harder for Tune-A-Video to fine-tune SDv2.1 to create a text-to-video model since it does not come with SDv2.1 in the form of a separate ControlNet. We also tried using the UNet3DConditionModel class from the latest version of the HuggingFace diffusers library, but ran into issues since the base diffusion model of SDv2.1 only fits to 2D U-Nets. It doesn’t have the extra layer to support the temporal dimension.

### 5.3 Uni-ControlNet for Video Generation

In order to leverage incorporate Uni-ControlNet into video generation, we took inspiration from the inference module of VideoCrafter and modified the Uni-ControlNet code to create the frames of a video. Our first approach utilized the edge and depth maps of an existing video (a car turning on a

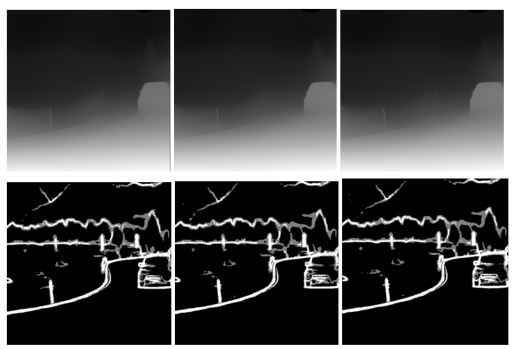


(a) Results after 100 epochs.



(b) Results after 500 epochs.

Figure 3: After fine-tuning a SDv2.1 model with Tune-A-Video, we generate videos of characters skiing in different styles with different prompts. The prompts from left to right are: "mickey mouse is skiing on the snow", "spider man is skiing on the beach, cartoon style", "wonder woman, wearing a cowboy hat, is skiing", "a man, wearing pink clothes, is skiing at sunset"



(a) "Control maps"

Figure 4: Edge and depth maps from car video.

road) in the form of a list of maps for each frame in the video. Then, during the generation of each subsequent frame, we use Uni-ControlNet to control the image generated using the two control maps of the corresponding original frame as shown in the following code: Appendix A. A sample of the edge and depth maps are shown in Figure 4

Using these control maps, we successfully generated a video using the prompt "jeep driving in the snow", which resulted in the following video Figure 5(three frames shown):

We were able to generate an output that adheres to not only the edges but also the depth of the original controls. This created a smooth motion of the car, although the background still had some significant

Figure 5: Video created using both edge/depth maps of existing frames

Figure 6: Video created using "true" generation

Figure 7: Video generated using edge map interpolation

fluctuations. While the ControlNet can control major features of the frames, we expect that this is because the network is not detailed enough to prevent small fluctuations between the edges. Therefore, the background still suffers from the same issue of inconsistency. We believe that in the future if we combine this framework with some methods like control interpolation or optical flow, we can create a smoother result while maintaining this improved control.

Next, in order to allow for a truly new video to be created, we first use the controls (edges and depth) of the previous frame to generate each subsequent frame. This created a result that did demonstrate decent consistency, but because each subsequent frame was restricted to the controls of the previous, there was no movement of the car in the video as shown in Figure 6

In order to try and facilitate more variation between frames while keeping consistency, many combinations of controls. The following two proved to be the most promising.

First, we use the edge maps of two consecutive frames and calculate an interpolation between them. With this method we hope to allow the subject to have slight movement, without sacrificing the smoothness of the frames. This produced a result that showed noticeably improved motion, using the prompt "dog running in the snow" in 7.

Second, we use a combination of content maps and optical flow. We remove any depth or edge controls and use the content of the previous frame as the content control for the next frame to allow more movement. To mitigate unrealistic perturbations between frames we apply optical flow from the OpenCV Python package to estimate realistic changes per pixel. Results show potential with very dynamic movement while maintaining some consistency. The figure below shows generation using the prompt "car driving on a road". 8

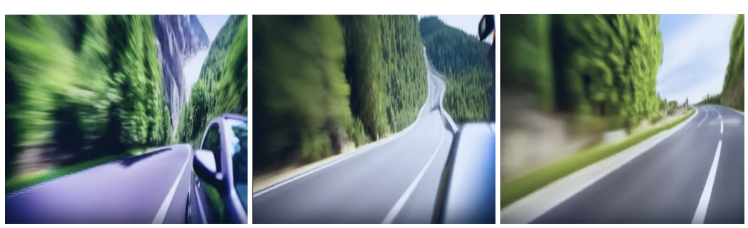


Figure 8: Video generated using content control and optical flow

### 5.4 NLA for Video Generation

As mentioned earlier, we intended to train atlas layers based on initial frames then use rigidity loss to encourage consistency of subsequent frames. However, we were unable to produce any significant results due to computational issues. Atlas layer representations typically require long pre-training times on entire videos, and doing so during runtime in our testing was too much for a single GPU to handle. We attempted to use the DistributedDataParallel wrapper to split the model across two GPUs (see appendix for code), but faced too many difficulties in the implementation to continue.

## 6 Further Work & Conclusions

### 6.1 Conclusions

Overall, through our investigation into the drawbacks and advantages of different video generation and editing approaches, we identified several key weaknesses and areas in need of improvement. First, since video generation relies on diffusion-based models to create each frame, it becomes difficult to control exactly how the entire video looks visually. In particular, we want to make sure that input controls like depth, edges, and text will be reflected consistently throughout the entire video. In order to improve performance in this regard, we showed the potential of using the Uni-ControlNet framework to allow for accurate and composable control of video generation. Next, we acknowledged the challenge of using pre-trained video editing methods like Neural Atlas Layers for improving consistency of video generation. While these methods work very well when editing an existing video, because we are creating new frames in real time, we don't have access to the over-arching pixel knowledge across the entire video that we need to calculate more consistent and rigid representations. Finally, we identified some drawbacks of the state-of-the art Stable Diffusion version 2.1 in it's lack of a 3-dimensional condition model.

### 6.2 Further Work

As we look to the future, we seek to further improve the quality of our Uni-ControlNet-based generation results by implementing 3D spatio-temporal attention architecture as well as other methods to allow for more flexibility between frames. We would also like to continue to experiment with the viability of using Neural Layered Atlases to generate video, perhaps with access to stronger computing power or better implemented multi-GPU models. Additionally, with the exciting announcement of new video generation models like SORA [8] and Stable Diffusion 3 [5], we hope to investigate and learn how these works might address some of the aforementioned shortcomings of current video generation methods. SORA utilizes latent spacetime patches along with a diffusion transformer block and conditioning using GPT-4 and CLIP in order to generates its video frames. We would be excited to investigate how these spacetime patches function to improve the spatio-temporal consistency of the resulting video. Stable Diffusion 3 also utilizes a diffusion transformer but foregoes the patching method, so it would be interesting to compare the results of these two models to determine whether the spacetime patching has a significant impact.

## References


[1] Omer Bar-Tal, Dolev Ofri-Amar, Rafail Fridman, Yoni Kasten, and Tali Dekel. Text2live: Text-driven layered image and video editing, 2022.

[2] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

[3] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter1: Open diffusion models for high-quality video generation, 2023.

[4] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-a-video: Controllable text-to-video generation with diffusion models, 2023.

[5] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, Kyle Lacey, Alex Goodwin, Yannik Marek, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis, 2024.

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[7] Zhihao Hu and Dong Xu. Videocontrolnet: A motion-guided video-to-video translation framework by using diffusion model with controlnet, 2023.

[8] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, Lifang He, and Lichao Sun. Sora: A review on background, technology, limitations, and opportunities of large vision models, 2024.

[9] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[10] Erika Lu, Forrester Cole, Tali Dekel, Weidi Xie, Andrew Zisserman, David Salesin, William T. Freeman, and Michael Rubinstein. Layered neural rendering for retiming people in video, 2021.

[11] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer, 2020.

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

[13] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow, 2020.

[14] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, 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, 2023.

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

[16] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-controlnet: All-in-one control to text-to-image diffusion models, 2023.

# A Appendix / supplemental material

## Implementations

### Initial Uni-ControlNet video pseudo code

```
edge_images = edges of every frame in the control video
depth_images = depth map of every frame in the control video

frames = []
initialize seed

for i in range(num_frames):
    samples = generate sample using DDIM sampler constrained by edge and depth map
    seed += i
    frame = samples[0]
    frames.append(frame)

frames = torch.stack(frames, dim=0)   // stack frames into video

```

### Uni-ControlNet video modified for "true" generation

```
prev_frame = initial_frame
initialize seed
frames = []

for i in range(num_frames):
    get canny edges of previous frame
    get midas depth of previous frame

    samples = generate sample using DDIM sampler constrained by edge and depth map
    seed += i
    frame = samples[0]
    frames.append(frame)
    prev_frame = frame

frames = torch.stack(frames, dim=0)   // stack frames into video
```

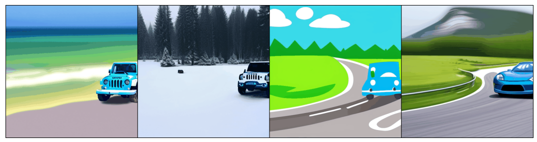
**Interpolating between edge maps**

```
def interpolate_edge_maps(prev_map, curr_map, curr_idx, n_frames):
    t = curr_idx / (n_frames - 1)  # Interpolation factor between 0 and 1
    interpolated_edge_map = cv2.addWeighted(prev_map, 1 -
t, curr_map, t, 0)
    return interpolated_edge_map

use interpolated edge map to guide next frame
```

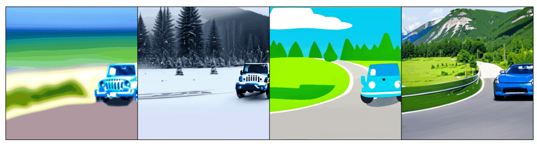
**Using DistributedDataParallel**

```
    def forward(self, x):
        def stem(x):
            for conv, bn in [(self.conv1, self.bn1), (self.conv2, self.bn2), (self.conv3, self.bn
                x = self.relu(bn(conv(x)))
            x = self.avgpool(x)
            return x

        x = x.type(self.conv1.weight.dtype)
        x = stem(x)

        # Split the model across two GPUs
        x1 = x[:, :, :x.shape[2] // 2, :].contiguous().to('cuda:0')
        x2 = x[:, :, x.shape[2] // 2:, :].contiguous().to('cuda:1')

        x1 = self.layer1(x1)
        x1 = self.layer2(x1)
        x2 = self.layer3(x2)
        x2 = self.layer4(x2)

        # Concatenate the results from the two GPUs
        x = torch.cat((x1, x2), dim=2)
        x = self.attnpool(x)

        return x
```

# Additional Figures



(a) Results after 100 epochs.



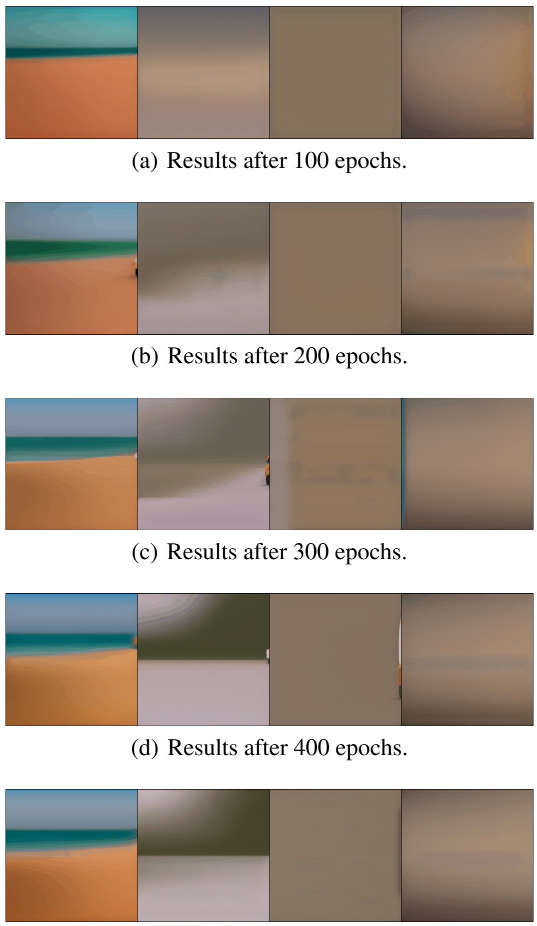
(b) Results after 200 epochs.

(c) Results after 300 epochs.

(d) Results after 400 epochs.

(e) Results after 500 epochs.

Figure 9: After feeding Tune-A-Video with a video of a jeep car turning, we generate videos of various cars turning in different artstyles and backgrounds using different prompts. The prompts in order from left to right are: "a jeep car is moving on the beach", "a jeep car is moving on the snow", "a jeep car is moving on the road, cartoon style", and "a sports car is moving on the road"



(a) Results after 100 epochs.

(b) Results after 200 epochs.

(c) Results after 300 epochs.

(d) Results after 400 epochs.

(e) Results after 500 epochs.

Figure 10: After fine-tuning a SDv2.1 model with Tune-A-Video, we generate videos of various cars turning in different artstyles and backgrounds using different prompts. The prompts in order from left to right are: "a jeep car is moving on the beach", "a jeep car is moving on the snow", "a jeep car is moving on the road, cartoon style", and "a sports car is moving on the road"