Cinematic Frame Generation with Diffusion Models

Raghav Ganesh
Stanford University
graghav@stanford.edu

Arjun Karanam
Stanford University
akaranam@stanford.edu

Ronak Malde
Stanford University
rmalde@stanford.edu

1 Introduction

1.1 Motivation

Diffusion models have now become the state of the art in creating high-quality images, audio, etc. A key challenge is now extending the diffusion architecture to more complicated modalities. Our eventual goal is to help video creators make their videos more cinematic, with dramatic lighting, more spectacular landscapes, and a more professional look. We first wish to explore this change of style for images, and then explore the possibilities of extending to videos.

1.2 Related Works

The first task that we aim to tackle in this project is efficient finetuning of the diffusion model. Low Rank Adaptation (LoRA) [4] achieves efficient finetuning by replacing the massive weight matrices with far smaller residual matrices to make finetuning faster. Although LoRA was initially presented for finetuning large language models, it has also shown significant success for image generation diffusion models as well. We will also evaluate various strategies for textual guiding of generation, which include Textual Inversion [3], DreamBooth [7], and ControlNet [8]. As a final extension, we will be exploring conditional generation of videos. One major work in this area is Dreamix [5], which applies temporal consistency by adding a low-resolution temporal input to the model. Another work, called Video-ControlNet [2], builds a depth map that the model must adhere to after conditional generation.

2 Problem Statement

For our project, we seek to first implement the default project, where we will fine-tune the open source Stable Diffusion model to generate more cinematic pictures. We will leave the term "cinematic" to be a subjective term, but generally this would translate to more dramatic lighting, grandiose landscapes, and lens flare effects. Next, we hope to take this Diffusion model and condition it on a new type of input using the architecture described in the ControlNet paper [8]. Finally, we hope to apply these techniques to video generation models with temporal consistency, so that someone can input their existing video, and then our model generates a more cinematic version of the existing video.

2.1 Dataset

Since there is no specific dataset that has cinematic photos/videos and also corresponding captions, we constructed our own dataset of around 50 cinematic images and associated captions. The dataset contained a combination of images found on the internet and images generated by DALLE-3, and consisted of shots of people, objects, and landscapes, with features such as dramatic backdrops and rays of light, creating a cinematic effect. In total, we generated roughly images using a wide variety of subjects and scenes. Additionally, we ensured that the prompts used to create the dataset were not used when evaluating the approaches below (i.e if an image of an astronaut is in the fine-tuning

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

dataset, the models will not be evaluated on their ability to create cinematic astronauts). Some examples of images can be found in figure 1.

(a) An astronaut exploring the moons of jupiter

(b) The siege of Constantinople

(c) A soccer player dribbling a ball

(d) Sunglasses lying on a beach

Figure 1: Four images sampled from the created Cinematic Dataset

3 Approach

Our technical approach is as follows. First, we plan on using LoRA to fine tune a diffusion model to generate images in our cinematic style. To do so, we use an augmented version of the Cinematic Style dataset. Second, we plan to try other methods to improve this generation, such as textual inversion and Dreambooth. Finally, we plan to use our learnings from these two steps and create a model that can take entire videos, and make them more cinematic. In order to do this, we will need to use LoRA to finetune a diffusion model in order to make it conditional - for the video editing task, we don't want to create a new series of images from scratch. Instead, we want to take existing images and edit them. We go into further detail on each of these below:

3.1 Image Methods

3.1.1 Stable Diffusion (Baseline)

Stable Diffusion [6] is a latent diffusion model, which decomposes the overall diffusion training process into two steps in order to reduce the complexity of training. First, a VAE is trained to reduce the RGB image into a more compact latent space representation. Then, a diffusion model is trained on this latent space, rather than the higher dimensional pixel space. This makes training far more efficient and allows the model to learn from the already latent representation. The diffusion model also adds functionality for conditional generation by adding a cross-attention layer to the existing U-Net architecture already in the diffusion model, and then jointly training both under the same objective, shown in Figure 2. Here x is an image, z_t the latent noised image, t the time step, and $c_\theta(y)$ a model that converts a conditioning input y to a conditioning vector, and ϵ_θ is the network to be learned. Also, ε represents the encoder which maps the images x into the latent space z.

$$L_{DM} = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2\right]$$

Figure 2: Training loss for Latent Diffusion Model

3.1.2 Extension 1 - LoRA

First, we will use LoRA [4] to finetune an existing diffusion model to our Cinematic Dataset. Unlike traditional fine-tuning methods, which modify all parameters of the model, LoRA focuses on updating only a small subset of parameters. Specifically, it introduces low-rank matrices A and B that are applied to the weights of the model's layers.

For a weight matrix W in the model, the LoRA adjustment is applied as follows:

$$W' = W + AB, \tag{1}$$

where W' is the modified weight matrix, and $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times d}$ are the low-rank matrices, with $r \ll d$ representing the rank of the adaptation.

During the fine-tuning process, only these low-rank matrices A and B are updated, while the original model parameters W remain unchanged. This approach significantly reduces the number of trainable parameters, leading to faster and more efficient fine-tuning. In our project, we used LoRA fine tuning with our curated cinematic dataset.

3.1.3 Extension 2 - Textual Inversion

Textual inversion [3] seeks to include a new, learned property into the textual embedding of the model so that it can be referenced in future prompts. This is done by specifying a new token, which we can denote S_*, and a small finetuning dataset in which all images exhibit or contain that property.
Then, the diffusion model is finetuned on this small dataset, where matching prompts are constructed in the form "An image of S_*", or "A rendition of S_*". The finetuning objective is simply to minimize the loss of the entire latent diffusion model, exactly the same as 2, to run optimally on this finetune dataset.

$$v_* = \arg\min_v \mathbb{E}_{z\sim\mathcal{E}(x),y,\epsilon\sim\mathcal{N}(0,1),t}\left[\|\epsilon - \epsilon_\theta(z_t, t, c_\theta(y))\|_2^2\right]$$

Figure 3: Training objective for Textual Inversion

Then, during inference, one has to include S_* in the prompt, and the model will output an image with that property. Figure 4 demonstrates how Textual Inversion can be used for style transfer during inference, which is the task we are focused on. In our project, we used the same finetune dataset of cinematic images, and finetuned the base stable diffusion model on this style.

Figure 4: Textual Inversion for style transfer

3.1.4 Extension 3 - Dream Booth

DreamBooth [7] aims to fine-tune diffusion models by embedding the subject into the output domain of the model, so that it can be generated with a unique identifier token in various different contexts.

This technique aims to avoid language drift, which is a phenomenon when the model associates the name of a class with the specific instance, by using the semantic prior embedded in models with a Class-specific Prior Preservation Loss. This in turn promotes the generation of various different instances of the subject class.

The finetuning objective in the case of DreamBooth, is to minimize the Class-specific Prior Preservation Loss, which is as shown in the figure. Here, $\hat{x_\theta}$ is a pre-trained text-to-image diffusion model with initial noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$. α_t, σ_t, and ω_t control the noise and sample quality, with process time $t \sim \mathcal{U}([0, 1])$. $\mathbf{x}$ is the ground-truth image and $\mathbf{c}$ is a conditioning vector obtained through the text prompt. Additionally, the data generated $\mathbf{x}_{pr} = \hat{\mathbf{x}}(\mathbf{z}_{t_1}, \mathbf{c}_{pr})$ is run on the pre-trained model with random noise $\mathbf{z_{t_1}} \sim \mathcal{N}(0, \mathbf{I})$. The conditioning vector is represented with $\mathbf{c}_{pr} := \Gamma(f(\text{"a [class identifier]" }))$.

$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t \|\hat{\mathbf{x}}_\theta(\alpha_t \mathbf{x} + \sigma_t \boldsymbol{\epsilon}, \mathbf{c}) - \mathbf{x}\|_2^2 + \lambda w_{t'} \|\hat{\mathbf{x}}_\theta(\alpha_{t'} \mathbf{x}_{\text{pr}} + \sigma_{t'} \boldsymbol{\epsilon}', \mathbf{c}_{\text{pr}}) - \mathbf{x}_{\text{pr}}\|_2^2],$$

Figure 5: Training objective for DreamBooth

To use DreamBooth, one must specify an instance prompt prior to training, and provide examples of instances, ideally the same subject. An example of this would be "An sks dog", where sks is the unique identifier. Following this, one can generate images of the subject in various different contexts with prompts such as "an sks dog in the ocean" or "an sks dog in a doghouse".

Figure 7 demonstrates how DreamBooth can be used for this. DreamBooth is fundamentally structured in the opposite manner as our project goals, as the instance contains images of the subject which can then be generated in various different styles. We, on the other hand, have curated a dataset of a specific style, which we would like to apply to various different subjects. Nevertheless, we believed it would be an interesting experiment to see how DreamBooth would perform for our task.

Figure 6: Dreambooth for subject generation in new contexts

3.1.5 Extension 4 - ControlNet

ControlNet [8] allows for more control when generating images with diffusion models by also conditioning the image generation output by some conditioning control, such as a canny edge image, human pose, or depth map. This allows for the generation of diverse output images that still still retain several higher-level structural elements of the input image it is conditioned on.

The learning objective for ControlNet builds upon the loss function for diffusion models in general, and is as follows. Here, $\mathbf{z}_0$ is the input image, $\mathbf{z}_t$ is the noisy image, t represents the number of times noise is added. $\mathbf{c}_t$ is the text prompt, $\mathbf{c}_f$ is a task specific condition, ϵ_θ is the network to be learned, and $\mathbf{t}$ is the current timestep.

For our project, we used ControlNet to apply a cinematic style transfer onto our input images, by using depth map conditioned ControlNet paired with prompts encouraging cinematic image generation, such as "epic, cinematic, vivid, detailed." Figure 8 demonstrates an example of how ControlNet can be used.

$$\mathcal{L} = \mathbb{E}_{\boldsymbol{z}_0, \boldsymbol{t}, \boldsymbol{c}_t, \boldsymbol{c}_\mathrm{f}, \epsilon \sim \mathcal{N}(0,1)} \Big[\| \epsilon - \epsilon_\theta(\boldsymbol{z}_t, \boldsymbol{t}, \boldsymbol{c}_t, \boldsymbol{c}_\mathrm{f})) \|_2^2 \Big],$$

Figure 7: Training objective for ControlNet

Figure 8: ControlNet for style and context transfer

3.2 Video Methods

3.2.1 ControlNet per Frame (Video Baseline)

For our video generation baseline, we applied ControlNet on a per frame basis to a reference video. This method generates frames which look quite high quality individually, but quite jarring when viewed as a sequence of frames as a video. This is due to the the lack of temporal consistency among frames, as the t^{th} generated frame is not generated conditioned on the $(t-1)^{th}$ generated frame or any of its predecessors. Figure 9 exemplifies what two consecutive frames generated via this method can look like.

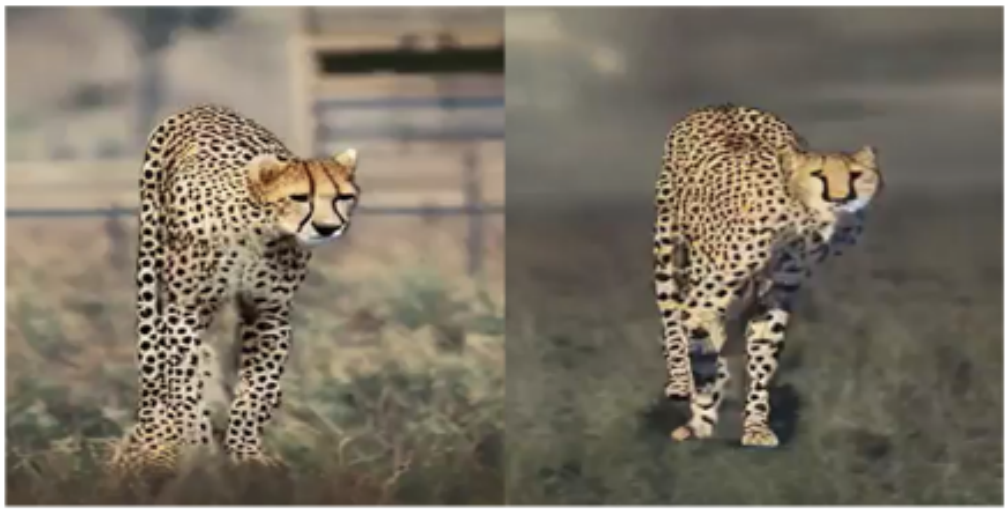

Figure 9: ControlNet Generated Frames

3.2.2 Control a Video

Control-a-video [1] extends the functionality of ControlNet to generate entire videos that are temporally consistent. It employs several methods in order to do this. First, it appends temporal layers after each 2D operation. For example, after each 2D convolution, it adds a 1d temporal convolution layer, and after every attention layer, it adds an additional temporal attention layer. Second, the paper formulates a Spatial-Temporal Self-Attention mechanism placed at the end of the model, which is

formulated with the equations in figure 10. Here, $\bar{v}_i$ denotes the token sequence of frame i, and $[\bar{v}_0, ..., \bar{v}_{N-1}]$ denotes the concatenation of the N frames. Then, they concatenate features K, V of N frames so that each position has a global perception of all video frames and tends to generate more consistent results.

$$SelfAttn(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d}})V, \quad (1)$$

$$Q = W^Q \bar{v}_i, K = W^K[\bar{v}_0, ..., \bar{v}_{N-1}], V = W^V[\bar{v}_0, ..., \bar{v}_{N-1}] \quad (2)$$

Figure 10: Control-a-video Spatial-Temporal Self-Attention mechanism

The third contribution that Control-a-video makes is it conditionally generates the noise vectors to generate temporally consistent results, rather than randomly initializing the noise with each frame. They employ residual-based and flow-based noise priors in order to generate noise for the next frame, which leads to smoother generation across time, since the latent representation of each frame is so closely related.

The following figure shows two frames generated with Control a Video.

Figure 11: Control a Video Generated Frames

4 Results

4.1 Baseline

As mentioned above, we had two separate baselines, one for our image methods, and another for our video methods. For both baselines, we used Stable Diffusion in its base form, as it is the state of the art when it comes to text-to-image models. As we will explain later, these image/video outputs weren't evaluated in isolation. Instead, each of our extensions was evaluated in comparison to the base Stable Diffusion output.

4.2 Experimental Procedure

All models were finetuned on an NVIDIA V100 GPU on GCP, and finetuned upon Runway ML's Stable Diffusion 1.5 model. For ControlNet, we used the ControlNet Depth model from Lymin Zhang

hosted on Hugging Face. The models were finetuned using mixed precision (fp16), 8000 train steps, and a learning rate of $1e-4$, and trained to generate images that were 512×512 pixels.

4.3 Evaluation Metrics

Since our end goal is very subjective (i.e what does it mean to be a cinematic image) and one that is hard to immediately quantify, we decided to leverage human preferences when evaluating our results. Our primary metric was a pairwise rating.

Pairwise Rating: For this metric, we created a ranking environment where a user is presented with a pair of images, where one image would be from Stable Diffusion, and the other from on of the models. For 100 different samples for each model, we each compared in a pairwise fashion which image was more cinematic. This number was aggregated and broken down by model.

Content Similarity: We also realized that content consistency was a major factor, especially when considering videos where temporal consistency is key. To evaluate this, we used a Vision Language Model (ViT-GPT2) to caption the content of both the initial image and the "Cinematic Image", and then compared their captions using a Cosine similarity distance from a sentence embedding model (all-MiniLM-L6-v2). Ideally, an edited image would not have its content changed.

Temporal Consistency: To evaluate the temporal consistency of the videos we are generating we are using the metrics Peak Signal-to-Noise Ratio (PSNR) and Temporal Structural Similarity Index (T-SSIM). These are metrics widely used in the space of diffusion based video generation to evaluate the quality of the temporal consistency of generated videos. PSNR indicates how close a frame is to its previous frame, relative to the notion that subsequent frames have some level of distortion or noise added to them. T-SSIM determines the similarity of two adjacent frames by considering the image luminance, contrast, and structure. Higher PSNR and T-SSIM values are indicative of higher temporal consistency.

4.4 Results

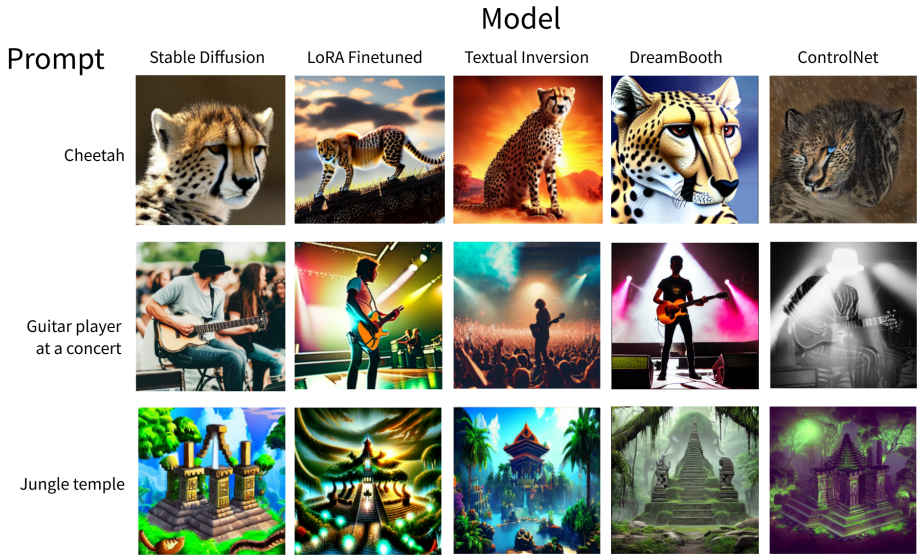

Figure 12: ControlNet for style and context transfer

	LoRA Finetune	Textual Inversion	Dream Booth	ControlNet
Pairwise Rating	89.2%	96.6%	52.4 %	33.0%
Content Similarity	0.79	0.78	0.88	0.43

Table 1: Quantitative Results of Image Methods

	Video Baseline	Control a Video
Pairwise Rating	49.7%	67.4%
Content Similarity	0.68	0.87

Table 2: Quantitative Results of Video Methods

	PSNR	T-SSIM
Input Video	38.5 ± 2.93	0.96 ± 0.026
Per Frame Generation	27.98 ± 0.11	0.26 ± 0.095
Control a Video	51.49 ± 14.36	0.89 ± 0.16

Table 3: Temporal Consistency of Videos

5 Analysis

First we can look at the image methods.

The default Stable Diffusion model consistently generates images which fit the prompt criteria, but are quite varied when it comes to style. Some images have a mobile game aesthetic while other generated images look as though they are a stock image or a taken on a smartphone camera. These images also lack common and consistent cinematic attributes.

The LoRA fine-tuned model consistently produced cinematic images, which are also quite inline with the fine-tuning dataset provided. The images generated and the dataset images all have vibrant colors, "epic" lighting, and other identifiable cinematic attributes.

Textual inversion also provides quite impressive images, with the images consistently excelling in terms of their cinematic quality. The images generated by textual inversion also seem more lifelike, especially when looking at the cheetah and jungle temple, when compared with the images generated from the default Stable Diffusion or LORA finetuned models.

The DreamBooth model generated images that fit the prompt, but not all were especially cinematic in our perspectives. For example, the cheetah image looks far less cinematic and life-like than even the default Stable Diffusion model but the guitarist image looks far more cinematic than the default model.

Finally, the ControlNet model had the worst results out of all of the models provided, due to how poor some of the images generated were both in terms of visual structure and cinematic quality. For example, the cheetah image is quite deformed, and the guitarist image is partially unrecognizable.

Next are the video methods. The video baseline method, since it was using ControlNet, got similar results as the ControlNet for a single image, where the output was not cinematic but it did hold true to the original input. However, the baseline lacked temporal consistency, generating jarring images back to back. This was much better in Control-a-Video, that had improved temporal consistency and slightly more cinematic outputs, but still had room for improvement. For example, a generated cheetah in several of the generated frames was still deformed to a degree, even if the general shape was mostly there. This was likely largely a result of a poor canny edge/depth map given to ControlNet, as these issues are also present in single frame ControlNet as well.

6 Conclusion

Through our work, we found that Textual Inversion worked best in our goal to generate cinematic images, given our prompts and the fine tuning dataset which we supplied. Other methods such as LORA fine-tuning and DreamBooth also provided impressive generated images, but images generated Textual Inversion remained most cinematic as seen through our results and analysis. With regards to our video generation pipelines, we found Control a Video to be an effective technique of generating temporally consistent videos given a input video, and one which far outperformed our baseline method. Throughout our experimentation, we did find that the finetuning process was very sennsitive

to hyperparameters. Thus, the models could perform better in several given substantial tuning of parameters like the learning rate, early stopping, and optimizer settings.

6.1 Future Work

We see lots of potential for this project going forward. Currently, we have an approach that can edit videos in a temporally consistent manner. However, this is done across one scene. How might our method adapt to, say, a movie, where there are multiple scenes? Here we would want intra-scene coherence, but not necessarily inter-scene coherence. Another potential extension is to look into is expanding our dataset. This could be done in an automated way by creating pipeline in which Dall-E 3 automatically generates images, and a multimodal vision model is used to caption the images. Finally, we could explore how to combine some of our finetuned models with the existing Control-a-video pipeline, in order to leverage the strengths of different techniques.

References

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

[2] 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.

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

[4] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685, 2021.

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

[6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021.

[7] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2023.

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