Generating High Quality Anime Videos with Diffusion

Justin Lim
Stanford University
jlim23@stanford.edu

Winston Shum
Stanford University
wshum@stanford.edu

Jonathan Lee
Stanford University
jezra@stanford.edu

Abstract

Generating high-quality anime videos remains a significant challenge due to the labor-intensive nature of traditional animation processes, as well as complexities involved in maintaining temporal coherence and smooth motion dynamics between scenes. This project aims to improve anime generation through a multi-stage frame generation and refinement pipeline. We first propose leveraging pre-trained models such as SVD, ModelScopeT2V, and I2VGen-XL to generate initial video frames from text and image inputs. Subsequent video frames are then generated using the StreamingT2V framework, which employs autoregressive conditioning to ensure consistency and smooth transitions. The final refinement stage incorporates anime-specific interpolation (AnimeInterp) and super-resolution enhancement (AnimeSR) techniques to output videos specifically for anime studios. Evaluation metrics, including Frechet Video Distance (FVD) and Learned Perceptual Image Patch Similarity (LPIPS), are used to assess the perceptual similarity and quality of the generated videos. Our results show that fine-tuning Stability AI's diffusion model, SVD, yields the best performance, demonstrating the potential of automating the production of high-quality anime videos that adhere closely to the original anime style.

1. Introduction

Japanese animation studios spend extensive hours animating and producing entire anime videos using a limited number of manga frames. Despite technological advancements, overworked animation studios often produce videos with low frame rates due to the labor-intensive process of drawing and designing each frame while meeting tight deadlines. However, recent developments in video understanding and generation within the deep learning domain have aimed to mitigate some of these challenges by focusing on frame generation or enhancing the quality of low-resolution frames. However, the task of generating high-quality anime videos for extended durations remains largely unsolved due to several key challenges, including maintaining temporal coherence between frames, ensuring smooth optical flow, and effectively learning motion dynamics. As such, we aim to contribute to the field by experimenting with frameworks capable of converting single anime frames or context scripts to anime-style videos.

Our project involves the following steps: (1) **Initial Frame Generation** with models capable of image-to-video and/or text-to-video, (2) **Autoregressive Fusion** through feeding the initial frames from step 1 into StreamingT2V [2] to generate longer videos, and (3) **Quality Refinement** by refining the output with anime specific interpolation and resolution-enhancement methods from AnimeInterp [7] and AnimeSR [10] respectively. Using these approaches, we aim to experiment and search for the best pipeline that addresses the challenges of generating high resolution and high framerate anime videos that stay true the supplied context.

1.1. Problem Statement

Our problem statement is as follows: given either image or text, how can we generate an anime-style video that preserves optical flow, physical laws, and temporal coherence, incorporates context from the story, and maintains the animation style?

2. Related Works

2.1. A Survey on Long Video Generation: Challenges, Methods, Prospects

This paper provides a comprehensive overview of techniques for generating long-form videos [3]. It highlights a technique we plan to utilize in our project: the temporal autoregressive method. This method generates frames sequentially, with each frame conditioned on the preceding one, therefore creating a cohesive video that maintains temporal coherence across frames. The paper also discusses the application of diffusion models with autoregressive layers and their efficacy in maintaining temporal consistency and high quality in long video generation. While our videos will not be long (around 2 seconds in length) the underlying architecture and techniques remain highly relevant to us.

2.2. Phenaki: Variable Length Video Generation from Open Domain Textual Descriptions

Presented by Google in 2022, this paper introduces Phenaki, a model capable of realistic video synthesis from sequential textual prompts [8]. This work is particularly relevant to the multimodal aspect of our project, as our dataset includes not only frames of anime manga but also textual descriptions resembling a ”director's script” to guide the model in generating successive frames for animation. A significant advantage of Phenaki is the use of a bidirectional masked transformer, which facilitates the use of smaller sampling steps that disregard strict video sequencing, a feature that is particularly beneficial for anime animation. While Phenaki is unfortunately not open source, we likewise plan to incorporate textual prompts in our finetuning in hopes of generating similar or superior results to Phenaki.

2.3. VideoDrafter: Content-Consistent Multi-Scene Video Generation with LLM

VideoDrafter is a model released in 2024 that can create content-consistent multi-scene videos from a single input prompt. Unlike traditional models that typically focus on single-scene outputs, VideoDrafter effectively handles multi-scene narratives by generating a detailed, scene-by-scene script through an LLM. The scripts are frame-specific, and leverage entity descriptions, and camera movements, which are then used to produce reference images for each entity through a text-to-image model. These images serve as a stable visual anchor throughout the diffusion process, enhancing the consistency and coherence of the generated video scenes [5]. While we do not adapt this exact approach, the paper served as a reference to create our dataset for fine-tuning, which takes an input video and passes key frames into an LLM to create a director's script for the entire scene.

2.4. StreamingT2V: Consistent, Dynamic, and Extendable Long Video Generation From Text

StreamingT2V is a framework that utilizes autoregressive conditioning to ensure consistency and smooth transitions for specifically longer video generation (80 - 1200 fps) and a Conditional Attention Module (CAM) to enhance video coherence by conditioning the current frame on extracted features from previous frames [2]. This also uses a diffusion model and maintains temporal consistency. This paper essentially represents a different (but open-source) option in comparison to Phenaki. Before the Streaming T2V Stage, the Initialization Stage creates an initial 16-frames to feed into the Streaming T2V Stage using a pretrained text-to-video model called Modelscope. Their Github also has the option to have this initialization stage use Stable Video Diffusion from Stability AI. Both are referenced below.

2.5. ModelscopeT2V

ModelscopeT2V is the original model that StreamingT2V proposed to create the initial batch of 16 frames for their pipeline. ModelScopeT2V incorporates spatio-temporal blocks to ensure consistent frame generation and smooth movement transitions [9]. It is designed to adapt to different video lengths during both training and inference, making it suitable for a wide variety of image-text and video-text datasets. The primary components of the model include a VQGAN, a text encoder, and a denoising UNet. During training, ModelscopeT2V employs a diffusion process into a latent space to train the UNet. In the inference phase, the text encoder converts the prompt into a text encoding that guides the UNet's denoising process on sampled random noise. The VQGAN then transforms the denoised output to generate the resulting video. During training, ModelscopeT2V uses a diffusion process into a latent space to train the UNet. In the inference phase, the text encoder converts the prompt into a text encoding that guides the UNet's denoising process on sampled random noise. The VQGAN then transforms the denoised output to generate the resulting video.

2.6. Stable Video Diffusion: Scaling Latent Video Diffusion Models to Large Datasets

Stable Video Diffusion (SVD) is another alternative that StreamingT2V uses to create the initial batch of 16 frames for their pipeline. While ModelscopeT2V takes in a prompt to generate the video, SVD uses an input image. The paper describes 3 stages of successful video late diffusion model training: text-to-image pretraining, video pretraining, and high-quality video fine-tuning [1]. The paper uses Stable Diffusion 2.1 as its base model, which is natively a text-to-video model. They finetune this model for image-to-video generation, where the input is now a still frame that conditions the outputted video. To do this, they replace the original text embeddings of the input prompt with a CLIP image embedding of the input image. Then, they concatenate a noise-augmented version of the conditioning image to the input of the base model's UNet.

2.7. I2VGen-XL

I2VGen-XL is a model that we propose for creating the initial 16 frames for the StreamingT2V pipeline. Unlike SVD or ModelscopeT2V, this model can take in both a textual prompt and a source image to create short videos [11]. The framework uses a cascaded strategy split into two stages: a base stage for generating the base video and a refinement stage for improving the video quality. In the base stage, two hierarchical encoders simultaneously capture the high-level semantics and low-level details of the input image and feed the embeddings into a latent diffusion model. This gives the base video more realistic motion and preserves the content of the generated frames. The refinement stage then uses the user's text as a condition for another video diffusion model, allowing it to fix issues like noise, temporal and spatial jitters, and deformations. An important thing to note is the authors mention **that I2VGen-XL specifically struggles with anime.** Thus, we implement StreamingT2V and anime-specific refinement techniques to generate superior anime videos.

2.8. SDEdit

SDEdit is a method published in 2022 for image synthesis and editing that utilizes a stochastic differential equation (SDE) to enhance the realism of input images. It begins by adding noise to an input image, which may include any type of user guidance. Following the noise addition, SDEdit employs the SDE prior to iteratively denoise and refine the image, significantly improving its realism [6]. This process makes it well-suited for integrating as a refinement layer in autoregressive diffusion models for video enhancement, as also noted by the StreamingT2V paper. This serves as a baseline method for image refinement, but the next two related works refer to refinement techniques that are specifically used for animation refinement rather than hyper-realism.

2.9. Deep Animation Video Interpolation in the Wild

This paper explores the application of neural networks for predicting intermediate frames in animated sequences as well as refining them for specifically animated videos through a refinement model called AnimeInterp [7]. There are two key challenges that their model, AnimeInterp, resolves: 1) the lack of texture in cartoons, making frame interpolation difficult, and 2) exaggerated motions that are non-linear and thus more difficult to predict. It uses Segment-Guided Matching to resolve the "lack of textures" issue by exploiting global matching among color pieces that are piecewise coherent and uses Recurrent Flow Refinement to fix the issue of anime's exaggerated motions. In our architecture, we utilize AnimeInterp immediately after the Streaming2TV block.

2.10. AnimeSr

The original StreamingT2V pipeline ends with a refinement stage that autoregressively applies a high-resolution text-to-short-video model using randomized blending to enhance the long video. Because our goals are to work with animations and not real-world videos, AnimeSr could be used as an alternative model for video refinement. The model combines the efficiency of unidirectional recurrent networks and sliding window approaches to achieve super-resolution [10]. Notably, the authors use an input-rescaling strategy because it eliminates artifacts while not affecting the details of the animation under the proper rescaling factor. Because animated videos consist of smooth segments, lines, and colors unlike real-world videos, downscaling was found to make frames look cleaner. Thus, we utilize AnimeSR as the final portion of our pipeline after the outputted video from StreamingT2V has been further interpolated by AnimeInterp.

2.11. Collaborative Neural Rendering Using Anime Character Sheets

This paper was primarily used as a benchmark sanity check in comparing our LPIPS scores. Their model takes in anime character sheets and generates new anime characters but with various motions [4]. The resulting images are then compared with the original character sheets using LPIPS. This paper was particularly relevant to us because they compute LPIPS between original anime and generated anime.

3. Methods

Our overall proposed methodology for anime video generation involves a multi-step pipeline. (1) We leverage various pretrained models to generate an initial 16 frames from either text, an anchor image, or both. (2) Using the generated frames from each model, we evaluate these short 16-frame videos by calculating FID and FVD scores against the original anime frames. After evaluating, we select the most promising models and fine-tune them for downstream tasks. (3) We take the generated images from the selected fine-tuned models and use them as input to the StreamingT2V framework. (4) Finally, we enhance the output from StreamingT2V using anime-specific refinement techniques. We then conduct our final evaluations using FID and FVD. In essence, we run the initial generated frames from all models through the entire pipeline to obtain final anime videos that we can compare against the originals. Something that is important to note is that some of the model's expect inputs of different resolutions and also output different resolutions than the gold standard resolution of 1280 x 720. Thus, before evaluating we resize the frames, which we detail in the evaluation subsections.

3.1. Generating Initial Video Frames

We first experiment with different pretrained models to generate a short video of 16 frames. We experiment with both image-to-video, text-to-video, and image and text-to-video models that can be fine-tuned for the anime domain using our fine-tuning dataset. The following models were considered:

- **SVD:** Stability AI's Diffusion XT model that primarily support image-to-video generation
- **I2VGen-XL:** created by Alibaba, I2VGen-XL is an image-text-to-video model leveraging cascaded diffusion models
- **ModelscopeT2V:** created by Alibaba, ModelScopeT2V incorporates spatio-temporal blocks on top of the classic text-to-image models by bringing together a VQGAN, a text encoder, and a denoising UNET

3.2. Evaluation Metrics and Initial Evaluation

After generating initial frames from the three pretrained models and our fine-tuned model, we resize the gold standard frames to match the output resolution. For example, SVD outputs 1280 x 704 yet the gold standard frames are 1280 x 720. Once resized, we evaluate these frames using two key metrics. Thus, by standardizing the resolutions, we are able to more accurately compare performance between models.

One metric we will employ is the Frechet Video Distance (FVD) to evaluate the generated videos. FVD measures the similarity between the distribution of videos, which in our case, will compare our generated videos and the gold standard (real anime frames). FVD extends the Fréchet Inception Distance (FID), which is used for images, to the temporal domain of videos. We pass our generated videos through a pretrained neural net, such as a 3D Conv-Net, allowing us to extract temporal features in the video. Then we use the feature representations to calculate the mean and covariances and calculate the distance with the same equation:

$$\text{FVD}(P, Q) = \|\mu_p - \mu_q\|_2^2 + \text{Tr}(\Sigma_p + \Sigma_q - 2(\Sigma_p\Sigma_q)^{1/2})$$

where:

- μ_P and Σ_P are the mean and covariance of the feature representations of real videos.
- μ_Q and Σ_Q are the mean and covariance of the feature representations of generated videos.
- $\|\mu_P - \mu_Q\|_2^2$ is the squared Euclidean distance between the means.
- Tr denotes the trace of a matrix.
- $(\Sigma_P\Sigma_Q)^{1/2}$ is the matrix square root of the product of the covariance matrices.

Another potential evaluation metric is the Learned Perceptual Image Patch Similarity (LPIPS), which calculates perceptual similarity between two images. We compute the similarity between the activations of two image patches when passed through a pretrained network like VGG. LPIPS has been shown to correlate well with human judgments of image similarity and takes into account image texture, structure, and semantics. This makes it a great choice for evaluating our generated video frames. The calculation typically uses a weighted L2 norm between the extracted feature maps from multiple layers in the neural network. A low LPIPS score means that image patches are perceptual similarFormally, the LPIPS score between two images, x and y, is calculated as follows:

$$\text{LPIPS}(x, y) = \sum_l w_l \cdot \frac{1}{H_l W_l} \sum_{h,w} \left\| \hat{\phi}_l(x)_{hw} - \hat{\phi}_l(y)_{hw} \right\|_2^2$$

where:

- l indexes the layers of the neural network.
- w_l are the learned weights for each layer.
- H_l and W_l are the height and width of the feature maps at layer l.
- $\phi_l(x)$ and $\phi_l(y)$ are the feature maps of images x and y at layer l.
- $\hat{\phi}_l(x)$ and $\hat{\phi}_l(y)$ represent the normalized feature maps

These metrics will provide a quantitative measure of how closely our generated frames and videos match the quality and characteristics of the gold standard frames in our dataset, ensuring that the generated content maintains high fidelity to the original anime style.

While other potential evaluation metrics were considered, we chose to focus on FVD and LPIPS since they are often seen as the standard metrics for evaluating generative models when performing video-generation or image-generation tasks.

3.3. Finetuning Initial Models

Based on our FID and FVD calculations in addition to qualitative analysis, we choose the most promising models and finetune them for short video diffusion with our small anime dataset. We then re-evaluate the FID and FVD calculations and use the fine-tuned model as the basis for StreamingT2V.

3.4. StreamingT2V Framework

StreamingT2V, a framework first proposed by Picsart AI Research, uses an autoregressive approach for long video generation for 80, 240, 600, 1200 or more frames with smooth transitions. The framework builds upon a pretrained text-to-video model by adding two key components.

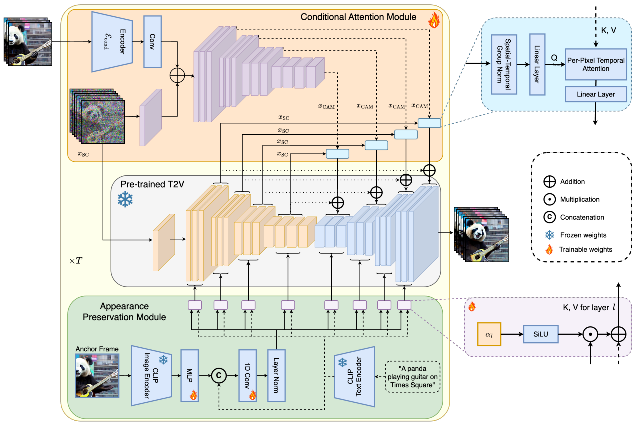

Figure 1. Architecture of StreamingT2V generation block

- **Conditional Attention Module (CAM):** the short term memory block conditions the current video generation on features extracted from previous chunks using an attentional mechanism, helping facilitate smooth chunk transitions

- **Appearance Preservation Module (APM):** This long-term memory block extracts high-level scene and object features from the initial video chunk, preventing the model from forgetting the initial scene and helps ground the model throughout the generation of the video

In our proposed methodology, we take the original 16 frames generated by the fine-tuned pretrained models along with the prompt as the inputs into the StreamingT2V framework, allowing us to autorgressively generate longer videos with greater frame lengths.

The specific architecture of the StreamingT2V framework is shown in Figure 1.

3.5. Long-Video Refinement

The original StreamingT2V paper uses a randomized blending technique alongside a text-to-video model like MS-Vid2Vid-XL. A diagram of the StreamingT2V Refinement stage can be shown in Figure 2 Our methodology proposes the following enhancements based on previous literature on anime video refinement.

- **AnimeInterp:** We aim to leverage an interpolation technique for anime as described in the AnimeInterp paper to generate high-quality intermediate frames, enhancing the smoothness of the final animation

- **AnimeSR:** Finally, after we interpolate frames at a lower resolution, we use a final refinement technique in the form of AnimeSR's resolution enhancer specifically designed for anime, improving video quality on the long video frames

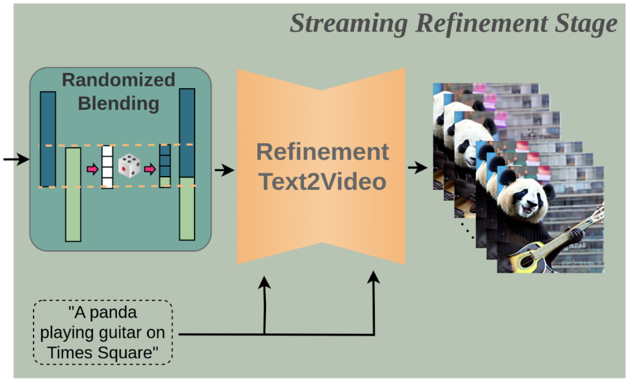
Streaming Refinement Stage
Randomized Blending
Refinement Text2Video
"A panda playing guitar on Times Square"

Figure 2. StreamingT2V refinement block

3.6. Final Evaluation: Outputted Videos

For our final evaluation, we take each initial model and feed it through the entire pipeline and compare their outputted videos to our gold standard (original) anime frames using FID and FVD. Thus, the baseline method can be seen as the outputted videos of the pre-trained, non-fine-tuned models after being fed through the StreamingT2V and anime refinement blocks of the model. These videos are evaluated against the fine-tuned model. An overview of our general architecture can be seen in Figure 3

4. Dataset

To our knowledge, there are no publicly available datasets for fine-tuning video generation models for the specific task of anime video generation. As such, we curate our own datasets by collecting online anime videos and subsequently process the data and split them into training and testing sets. Our pre-processing steps include compressing each anime video to 12 FPS, then segmenting them into around 10 second chunks.

Our training set consists of around 1400 anime videos, each being 25 frames long. We have a separate testing set consisting of a few short videos from another anime, each standardized to the model's desired input resolution so that the pixels and features match.

4.1. Training Set

The training set was curated for fine-tuning the pre-trained models. We originally created a training set to fine-tune I2VGen-XL, which is trained simultaneously on image-text pairs and video-text pairs. The image-text pairs are generated by taking random frames from the 10-second videos and passing them to GPT-4o for image captioning. We then create a text file where each line represents a link between an image file name and the corresponding caption. To create the video-text pairs, we use a similar approach, where we stream every 6th frame of a video to GPT-4o and

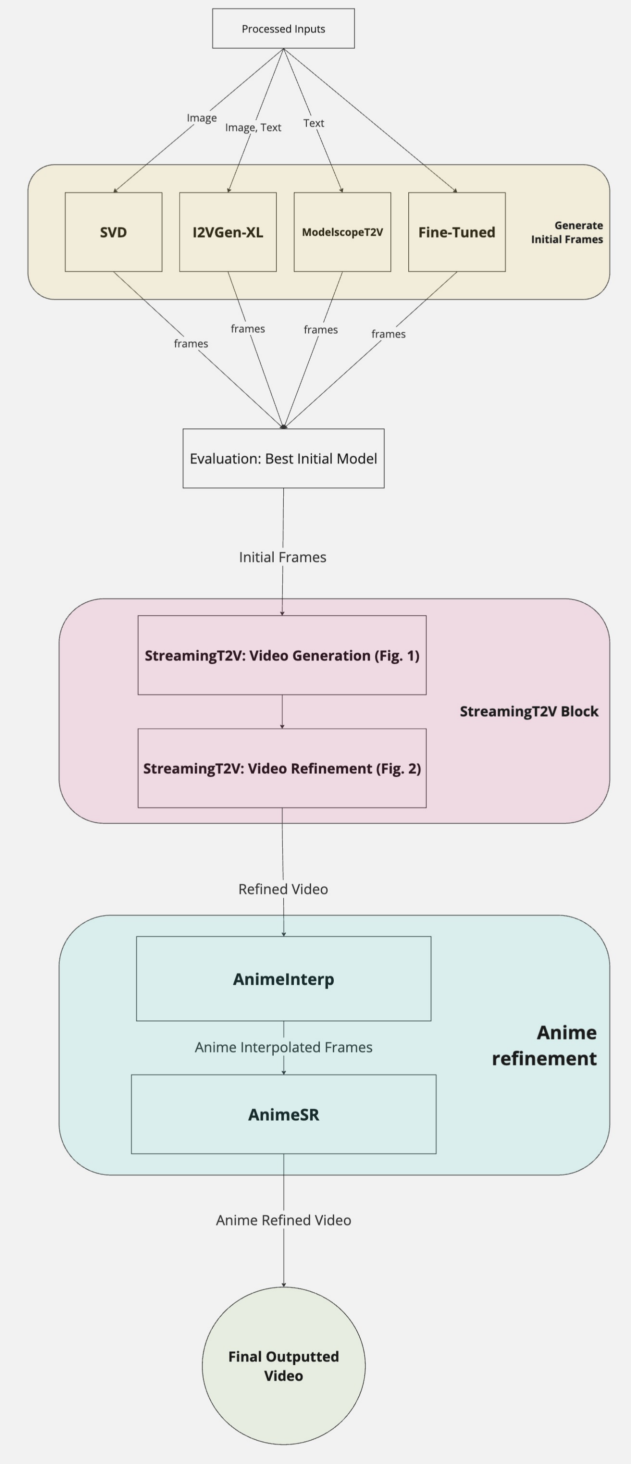

Figure 3. Overall Model Architecture

ask it to create a ”director’s script” of the video. We choose every 6th frame because it represents a half second of video time and it greatly reduces the amount of data that GPT-4o must process without sacrificing text quality.

Each model expects a different format for the dataset. I2VGen-XL extracts the first frame of a video that is paired with a text-prompt to generate videos. It also takes in image-text pairs for extra training. Thus, during the fine-tuning process, I2VGen-XL receives data like so: (1) Video-Text: (video_path|||director_script), and (2) Image-Text: (image_path,image_caption).

ModelscopeT2V takes in only text input, so we would only pass in image captions. SVD expects a directory containing video folders, where each video folder is a folder of 25 images corresponding to the first 25 frames of the original video. Thus, for SVD our fine-tuning set contained folders of these initial frames (no text).

4.2. Test Set

The test set was constructed and processed in the same manner as the training dataset; raw videos were downsampled and segmented into smaller chunks, then had their first frame captioned by GPT-4o. This results in 20 videos as the gold output and 20 image-text pairs. At test time, we pass either the image corresponding to the first frame of a video, the text prompt, or both (exclusive to I2VGen-XL) to the four pretrained video-generation models that we selected: (1) SVD: image-to-video, (2) ModelscopeT2V: text-to-video, (3) I2VGen-XL: text+image-video.

5. Experiments/Results/Discussion

5.1. Initial Evaluation

Our initial evaluation involved generating an initial 16 frames from the three baseline models. SVD was given a single anchor image from an anime as input, ModelScopeT2V was given an anime image caption, and I2VGen-XL was given both an anime anchor image and a director’s script. The outputted FVD and LPIPS scores between the outputted video frames of each model and the gold standard anime frames can be seen below.

Evaluation Metrics		
Model Name	FVD	LPIPS
SVD	**2523640**	**0.1434**
ModelScope	3849737	0.3862
I2VGen-XL	6571285	0.2916

The FVD scores are unusually high, but regardless, the baseline SVD model had the best (lowest) FVD and LPIPS scores. Accordingly, we decided to fine-tune SVD since it performed the best. This was surprising as we initially hypothesized that I2VGen-XL would perform the best because of its capability to integrate both image and textual inputs. However, it is worth noting that the poor performance on our anime images is consistent with observations in the original paper regarding the model’s difficulties with animated content. Qualitatively, we noticed that the video frames generated by I2VGen-XL would deteriorate, culminating in frames that turned completely white. We tried to fine-tune I2VGen-XL, but the results largely stayed the same. Thus, we proceeded to fine-tune SVD given it had the best initial performance of the three models.

5.2. Fine-tuning SVD:

To fine-tune SVD, we trained it on our custom anime dataset containing 1470 images, where each image represents the first frame of a 10-second anime video. We output 25 frames during training as this was the standard set by Stability AI in their paper introducing SVD [1]. During the fine-tuning process of SVD, we encountered a tendency towards overfitting when a high number of training steps were applied. This likely stemmed from the limited number of initial frames (16) that we are generating, which likely prompted the model to minimize training loss by reducing motion between frames. Consequently, the produced videos displayed minimal-to-zero movement. When using an insufficient number of fine-tuning steps, the outputted videos were highly unstable with abrupt and indecipherable motion.

Thus, to optimize the fine-tuning, we experimented with 1,200 and 2,000 training steps, denoted as SVD-1200 and SVD-2000, respectively. We believe that steps between 1200 and 2000 would strike an optimal balance between maintaining dynamic motion and avoiding the generation of chaotic video sequences. The results of the fine-tuning steps in comparison to baseline SVD can be seen below.

Evaluation Metrics		
Model Name	FVD	LPIPS
SVD	2523640	0.1434
SVD-1200	**1065506**	0.1254
SVD-2000	1482391	**0.1205**

5.3. Final Evaluation

Our final evaluation computes the FVD and LPIPS scores between our five models (SVD, SVD-1200, SVD-2000, ModelScopeT2V, I2VGen-XL) and the gold standard anime frames. Before computing, we resize the gold standard anime frames to match the resolution of the output from the models. The results of each baseline model and our fine-tuned SVD models, after being fed through our entire pipeline which includes the StreamingT2V, AnimeInterp (interpolation block), and AnimeSR (resolution refinement) stages, can be seen below:

Evaluation Metrics After Full Pipeline		
Model Name	FVD	LPIPS
SVD	3902156	0.1755
SVD-1200	1409045	0.1915
SVD-2000	**866915**	**0.1392**
ModelScope	9176460	0.4002
I2VGen-XL	60920012	0.4248

From these results, we see that SVD-2000 outperformed other models in both the FVD and LPIPS metrics. This contrasts with our observations made during the fine-tuning stage, where SVD-1200 was able to create videos with increased motion that was more realistic. We suspect that this is due to the autoregressive nature of the StreamingT2V block. Specifically, StreamingT2V introduced more motion in the subsequent frames, even if the initial set of 16 frames exhibits minimal movement. As a result, by having a more consistent initial batch of 16 frames, StreamingT2V is able to generate videos with less artifacting and smoother motions compared to using other initial models. Effectively, deviations in the initial 16 frames from the first frame are amplified in the subsequent frames generated by StreamingT2V. This effect is further illustrated by the significant increase in the results for the pipelines based on ModelScope and I2VGen-XL, where both the FVD and LPIPs scores deteriorated drastically. This highlights the significance of the consistency and stability of the first 16 frames on the quality of the subsequent frames generated by the autoregressive StreamingT2V block. Qualitatively, this can also be demonstrated through the figures 4,5,6,7:

6. Conclusion/Future Work

Overall, our project aims to solve the task of anime video generation from a single image-text pair. In evaluating our proposed methodology, our results highlight that SVD-2000 was able to generate anime videos that most closely resembled real anime videos. The choice of model to generate the initial frames, before being fed through the rest of the pipeline, had a strong influence on the outputted videos, with SVD (both before and after fine-tuning) showing the best results by far.

These results indicate the importance of initial frame generation, especially for complex architectures that have multiple stages of frame generation. Furthermore, the higher FVD and LPIPS scores for the baseline, non-fine tuned models that still went through the StreamingT2V and refinement stages indicate that strong interpolation and refinement cannot adequately improve generation if the initial frames are too poor.

Our current study is constrained by several limitations. First, both the fine-tuning and testing dataset used was self-curated, comprising of a limited number of scraped anime videos, captions, and images. Future improvements could involve experimenting with larger and more diverse datasets that could potentially increase model performance.

In addition, we were limited in our research due to limited access to computational resources, with only a single NVIDIA A100 GPU (80GB) available for fine-tuning and inference across all models and frameworks. Future work could benefit significantly from leveraging larger computational units, thereby allowing more extensive fine-tuning

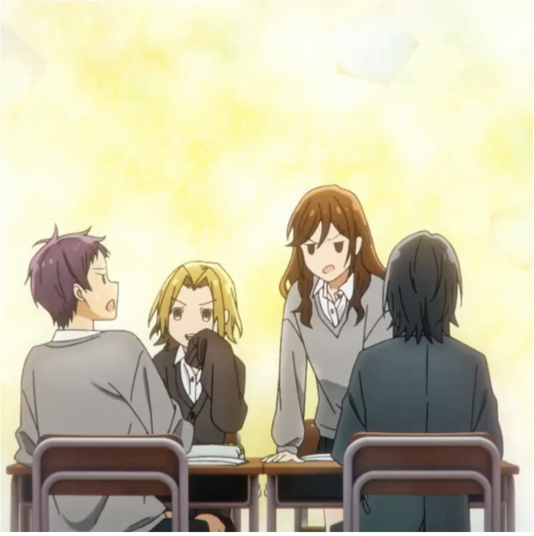

Figure 4. Frame 1 of a video

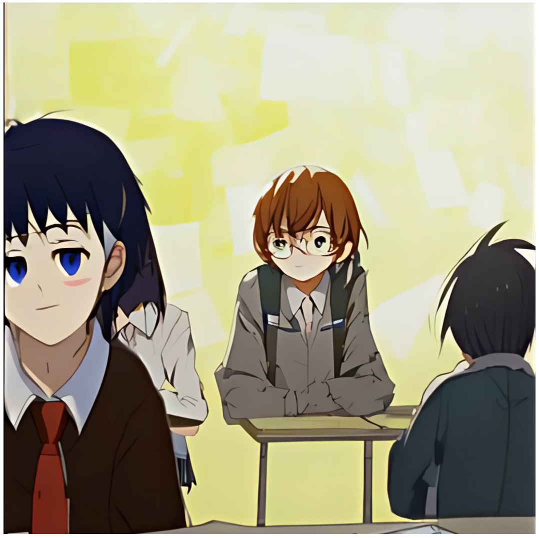

Figure 5. Final frame outputted by full pipeline using SVD basline

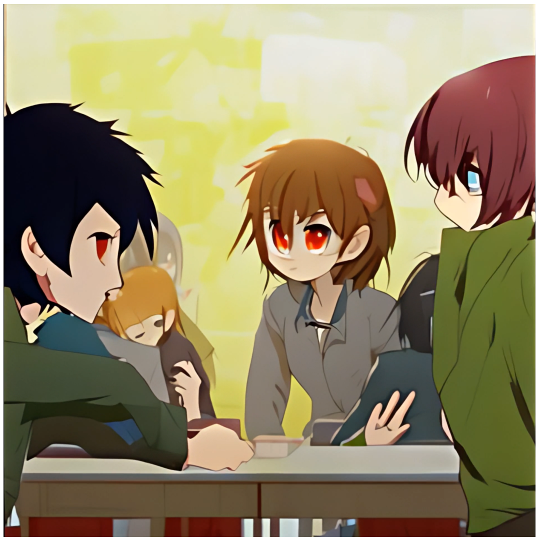

Figure 6. Final frame outputted by full pipeline using SVD-1200

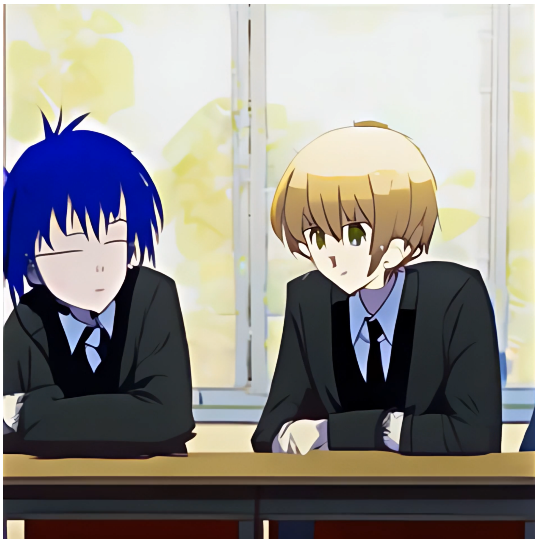

Figure 7. Final frame outputted by full pipeline using SVD-2000

and testing across a larger dataset. Finally, given that generating long videos with existing models remains a computationally intensive task, we were forced to reduce the length of the generated videos for the purpose of evaluation.

7. Contributions

Justin performed data collection and preprocessing. He also helped fine-tune and run inference on all models. Winston was in charge of fine-tuning and running inference on the first phase of models and StreamingT2V. Jonathan was in charge of setting up the refinement models (AnimeSR and AnimeInterp). He was also in charge of the majority of the paper.

References

[1] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, V. Jampani, and R. Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

[2] R. Henschel, L. Khachatryan, D. Hayrapetyan, H. Poghosyan, V. Tadevosyan, Z. Wang, S. Navasardyan, and H. Shi. Streamingt2v: Consistent, dynamic, and extendable long video generation from text, 2024.

[3] C. Li, D. Huang, Z. Lu, Y. Xiao, Q. Pei, and L. Bai. A survey on long video generation: Challenges, methods, and prospects, 2024.

[4] Z. Lin, A. Huang, and Z. Huang. Collaborative neural rendering using anime character sheets, 2023.

[5] F. Long, Z. Qiu, T. Yao, and T. Mei. Videodrafter: Content-consistent multi-scene video generation with llm, 2024.

[6] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations, 2022.

[7] L. Siyao, S. Zhao, W. Yu, W. Sun, D. Metaxas, C. C. Loy, and Z. Liu. Deep animation video interpolation in the wild. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2021.

[8] R. Villegas, M. Babaeizadeh, P.-J. Kindermans, H. Moraldo, H. Zhang, M. T. Saffar, S. Castro, J. Kunze, and D. Erhan. Phenaki: Variable length video generation from open domain textual descriptions. In *International Conference on Learning Representations*, 2023.

[9] J. Wang, H. Yuan, D. Chen, Y. Zhang, X. Wang, and S. Zhang. Modelscope text-to-video technical report, 2023.

[10] Y. Wu, X. Wang, G. Li, and Y. Shan. Animesr: Learning real-world super-resolution models for animation videos, 2022.

[11] S. Zhang, J. Wang, Y. Zhang, K. Zhao, H. Yuan, Z. Qin, X. Wang, D. Zhao, and J. Zhou. I2vgen-xl: High-quality image-to-video synthesis via cascaded diffusion models, 2023.