manGANime: Future Video Prediction with Conditional GANs

Stephen Huan, Luke Thistlethwaite

Thomas Jefferson High School for Science and Technology
Computer Systems Lab 2020–2021

Abstract

Unsupervised image synthesis has made enormous progress recently, especially with a class of models called generative adversarial networks (GANs). However, work on video generation remains stagnant because of its inherent difficulty. We consider a simplified domain for video generation by generating cartoon video ("anime"), and create a novel dataset for this task. We are then able to generate videos in an autoregressive and self-supervised manner, conditioning GANs on previous video frames as well as the comic book ("manga") pages which inspired the original anime. Source code is available here: https://github.com/stephen-huan/manGANime.

1 Introduction

A class of neural networks called generative adversarial networks (GANs) are now able to generate images of astounding quality and resolution. StyleGAN is able to generate very realistic images of human faces at a resolution of 1024x1024 [1], the quality of which was later improved in StyleGAN2 [2], and finally further developed to use less data with an adaptive discriminator augmentation system, dubbed StyleGAN2-ADA [3].

A natural extension of image generation is *video* generation, to generate multiple images in a coherent sequence. There are varying formulations of video generation, but we will consider the specific problem of *future video prediction*: given a few initial conditioning frames, predict the frames that follow. This provides more structure compared to completely unsupervised video generation, since the model is provided with conditioning frames and therefore "has something to go off of". Another benefit is that a model could be trained with pure self-supervision, since the ground truth video is known.

However, the aforementioned successes in unsupervised *image* generation have not necessarily translated to *video* generation. The current state of the art in future video prediction is comparatively limited; "... there has been relatively little work extensively studying generative adversarial networks for video" [4]. In particular, the DVD-GAN model generates up to 48 frames (4 seconds) at a resolution of 256x256 [5], a much smaller resolution and for a duration too short for most content generation applications.

Motivated by these shortcomings, we propose a simplified domain for video generation which will naturally improve results due to the abstractions inherent to this domain. StyleGAN and DVD-GAN are both most frequently trained on real-world datasets, for example the FFHQ dataset of human faces and the Kinetics-600 dataset of YouTube videos of human actions [1], [5]. We propose instead applying video generation algorithms to cartoon video, specifically a subset of animations produced in Japan called *anime*.

Our reasons for focusing on anime are as follows: first, there is a large volume of unlabeled anime on the internet. A popular online anime steaming service, Crunchyroll, has over 800 shows in its catalog [6], which assuming an average of around 12 episodes per show and 20 minutes per episode, corresponds to over 3,000 hours of video. The variance in visual style is also lower than that of American animation, for example. Secondly, many assumptions existing algorithms make are justified in anime. DVD-GAN, for example, skips every other frame as it can "...generate videos with more complexity without incurring higher computational cost" [5]. Correspondingly, anime is often animated "on twos", with every other frame duplicated, or even on threes or fours. [7]. Vondrick *et al.* [4] use convolutional GANs to model a dynamic foreground separately from a static background, masking them into a final video. Again, anime is actually created with an explicit foreground/background separation, as shown by specialized companies dedicated to producing background animation [8]. Thus, while the real world thwarts assumptions, anime is known to obey these priors *a priori*. Finally, there is a huge demand for anime, as shown by Crunchyroll's over 20 million users and 1 million paid subscribers [6], but a shortage of labor causing animators to work long hours for low pay [9]. Machine learning systems like the ones already used by Sony [10] can automate tedious work, allowing animators to focus on the creative aspects of their work.

Note that anime is not so abstract as to destroy any semblance of the real world, since algorithms on the real world apply to anime. For example, the popular object detection model YOLOv4, trained on the MS COCO dataset consisting of real world images [11] can be directly applied to anime without transfer learning, as shown in Figure 1.

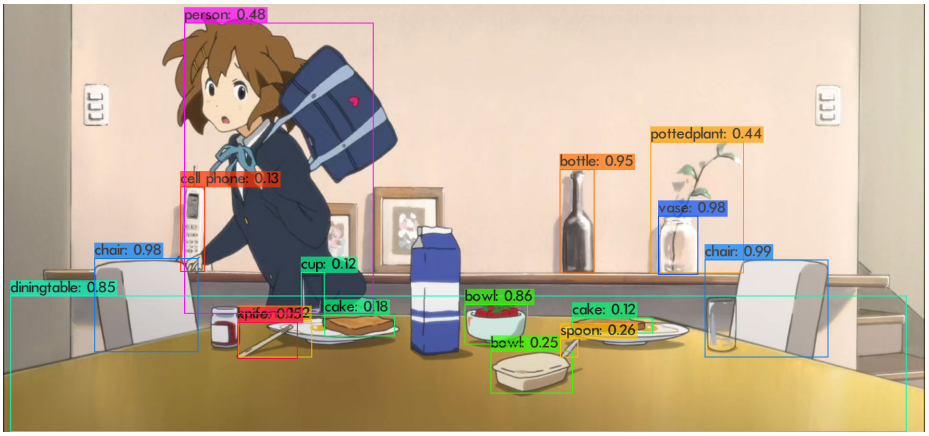


Figure 1: The official implementation of YOLOv4-CSP [11] on an image from *K-On!* [12], threshold of 0.1. See https://youtu.be/i8jCzh9nWc4 for a full demonstration.

In this paper, we show how to efficiently create a dataset of anime, each frame of which is tagged with its corresponding comic book (*manga*) page, a useful conditioning signal justified in section 3. We create such a dataset on *Girls' Last Tour* [13].

We then experiment with conditioning StyleGAN2-ADA [3] on the resulting dataset in order to generate video autoregressively, and with linear interpolation in the latent space of the model to create smooth transitions between two frames.

2 Related Work

Video generation Previous work on applying GANs to video generation include the aforementioned Vondrick *et al.* paper, which also makes use of large amounts of unlabeled video, models the foreground separately from the background, and experiments with adapting the discriminator for class prediction and the generator to future video prediction with a simple conditioning scheme, encoding the initial frame into the latent vector [4]. The current state of the art model on the Kinetics-600 and UCF-101 datasets is the aforementioned Dual Video Discriminator GAN (DVD-GAN), whose key insight is to have two (dual) discriminators, one for measuring the realism of each frame and another for measuring the realism of the video's motion as a whole; the separation of concerns allows each discriminator to act more efficiently [5]. We deviate from these approaches in conditioning an image generator for video generation rather than directly modeling videos as a 3D volume. We also condition upon manga, so the number of frames per manga page determines our target length. Our dataset averages 360 frames per page, skipping every other frame. DVD-GAN, however, generates only at most 48 frames [5].

Attention A technique that has proved very useful in language modeling is attention, intuitively a set of weights guiding what the network pays attention to. These weights are learned with gradient decent as usual. The landmark paper "Attention Is All You Need" introduced the *transformer* architecture, a model which abandons the recurrent and convolutional connections popular in language modeling in favor of pure attention, hence "Attention is All You Need" [14]. [15] then takes this transformer architecture and applies it to video generation, attaining state of the art performance on the BAIR robot pushing dataset with the resulting *video transformer*. We hypothesize that attention is a useful mechanism to guide a model's focus on its conditioning information, especially if its conditioning signal is as structured and sequence-like as manga. We do not, however, use standard autoregressive modeling techniques like recurrent connections or attention, and instead attempt to autoregressively generate video with conditioning alone.

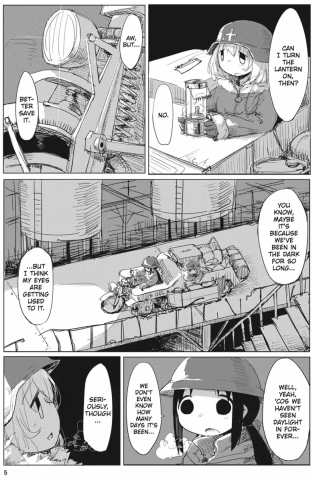
Latent space exploration Jahanian *et al.* experiment with "steering" GAN output by performing walks in the latent space. Their learned walks perform modifications like color changes, rotations, and camera zooms; they find that a linear walk performs as well as a nonlinear walk [16]. We experiment with linear interpolation in the latent space to generate transitions between two images, finding that the resulting video is "smooth", as would be expected from the perceptual path regularization done by StyleGAN [1], [2].

3 Dataset

Before discussing our methods for video generation, we first begin with an analysis to efficiently construct an anime dataset tagged with manga.

3.1 Picking an Anime

We make use of the large amounts of unlabeled video described in the introduction. However, because we focus on applying future video prediction to the domain of anime, we have an yet another advantage that the real world doesn't: source material. Production of anime is expensive, so anime is often animated with reference to a comic book, or *manga*. Certain anime will follow their source manga very precisely, as shown in Figure 2.



(a) A manga page from *Girls' Last Tour*, [17].

(b) A anime frame from *Girls' Last Tour*, [13].

Figure 2: The correspondence between anime and manga.

It is useful, therefore, to find anime with a high correspondence with its source manga. Formally speaking, we define a *page function* $f : \mathbb{Z} \to \mathbb{Z}$ that maps an anime frame index to its corresponding manga page. By the definition of a function, each element in the domain must be mapped to an element in the co-domain. Since our domain is anime frames, the existence of f implies that each anime frame has a correspondence in manga. This means we will remove frames that have no basis in manga, for example, the "intro" and "outro" songs at the beginning/end of most episodes. We further enforce that f is monotonic, i.e. $x \leq y$ implies $f(x) \leq f(y)$. This implies the anime "goes in the same order" as the manga, which we will see simplifies the construction of f immensely.

Many anime/manga pairs fulfill these conditions; we considered *Girls' Last Tour* [13] and its corresponding manga [17]–[20], *K-On!* [12] and its second season *K-On!!* [21] and their corresponding manga [22]–[25] among others. We decided on *Girls' Last Tour* because of its consistent visual style and authors' familiarity with the material. The following analysis will therefore use the quantitative features of *Girls' Last Tour*.

3.2 Exploiting Monotonicity for Efficient Tagging

Suppose we have an anime/manga pair and wish to construct f, that is, to tag each frame of the anime with its corresponding manga page. Since the average anime has 12 episodes, each episode is around 20 minutes without intros/outros, and we sample at 12 FPS, skipping ever other frame, we will have around 170,000 frames to process, exactly 174,792 total frames for *Girls' Last Tour* and 533 manga pages, which is too much to efficiently label by hand. But because f is monotonic, we do not need to tag each frame; we only need to tag the *boundaries* when f changes. For example, if we know $f(0) = 2$ and 600 is the smallest index such that $f(600) = 3$, then that implies the frame indexes 1–599 must be 2 by monotonicity. Thus, instead of tagging each frame one by one, we can tag in batches determined by the transitions between manga pages, reducing the number of operations from 170,000 (the number of anime frames) to approximately 533 operations (the number of manga pages). This is implemented in a graphical user interface (GUI) shown in Figure 3. However, we still need to find which frames mark a transition between manga pages. Because these critical frames naturally mark a transition in the anime as well, we hypothesize they can be automatically detected by measuring similarity.

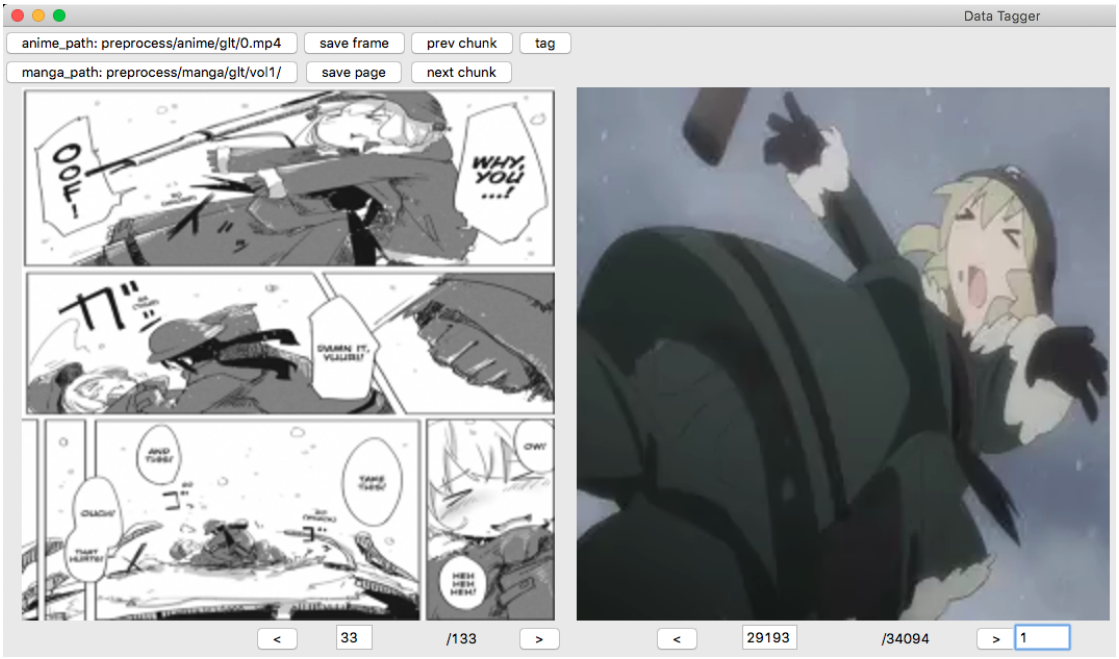


Figure 3: Python `tkinter` GUI for efficient manual data tagging. The interface permits navigation with the left/right arrows as well as jumping to an arbitrary index. Pressing the "tag" button implicitly tags the frames up to the current frame. The "next chunk" button automatically moves to the next transition, determined by a low similarity between adjacent frames. `pims` is used for lazy loading of the underlying image data.

3.3 Measuring Frame Similarity

We want to quantify a visual transition between two frames. We can do this by computing a similarity metric between adjacent frames, and if it is lower than a certain cutoff, declare that a transition must have happened between these two frames. Two vectors are similar if they point in the same direction, which can be measured with the *cosine similarity*:

$$\rho(\boldsymbol{u}, \boldsymbol{v}) = \cos\alpha = \frac{\boldsymbol{u} \cdot \boldsymbol{v}}{\|\boldsymbol{u}\|\|\boldsymbol{v}\|}$$

This metric varies between -1 and 1, and a higher value implies greater similarity. We intuitively generalize this dot product similarity to the element-wise product of two tensors, which for two matrices A, B can be written as follows:

$$\begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2 & \dots & \boldsymbol{a}_n\end{bmatrix} \cdot \begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \dots & \boldsymbol{b}_n\end{bmatrix} = \boldsymbol{a}_1 \cdot \boldsymbol{b}_1 + \boldsymbol{a}_2 \cdot \boldsymbol{b}_2 + \cdots + \boldsymbol{a}_n \cdot \boldsymbol{b}_n = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{ij}B_{ij}$$

Our similarity is then $\rho(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$ where $\|X\| = \sqrt{X \cdot X}$ as usual.

For our particular anime, we found a cutoff of 0.8 worked well. If the user presses the "next chunk" button in the GUI, Figure 3, then the program computes the closest frame such that the similarity between it and the next frame is less than 0.8. We naturally suffer from the *precision-recall trade-off*, since if the cutoff is higher we miss transitions and if the cutoff is lower we encounter spurious transitions. Future work may take into account more sophisticated measures of similarity, for example, by measuring the difference in deep VGG16 embeddings which is shown to align with human perception of similarity [26], or by taking into account changes in optical flow [27].

3.4 Relative Number of Monotonic Functions

Another approach to tagging data might be to train a neural network to approximate the page function f. Monotonicity still is helpful, as it restricts the number of possible functions for f. Suppose we have n anime frames and k manga pages. If f was unconstrained, we'd have n choices of k possibilities, k^n possibilities. However, f is constrained to be monotonic. A function is monotonic if and only if its outputs are sorted when read from increasing order of its inputs (this follows from the definition of sorted being essentially identical to the definition of a discrete monotonic function). A list has a unique sorted representation, so a monotonic function is entirely determined by the *values* of its outputs and not the ordering. For example, if we want to find a monotonic function from $\{1, 2, 3\}$ to $\{4, 5, 6\}$ and our output values are 4, 5, and 4, we must assign $f(1) = 4$, $f(2) = 4$, and $f(3) = 5$. Thus, the number of monotonic functions from n values to k values is the number of ways to put n unlabeled balls into k labeled boxes. By the classic "stars and bars" argument, the number of ways is $\binom{n+k-1}{n}$. For $n = 174792$ and $k = 533$, there are over 10^{1225} less monotonic functions than unconstrained functions, so enforcing monotonicity drastically reduces the functional space.

4 Methods

Background A generative adversarial network (GAN) is composed of a generator G and a discriminator D [28]. Intuitively, the generator attempts to generate realistic output while the discriminator attempts to discern between real and generated images [28]. Since the generator is trying to fool the discriminator while the discriminator is trying to catch the generator, the two are locked in competition, hence "adversarial networks". In order to allow the generator to generate varied output, a *latent code* or *latent vector* $\boldsymbol{z}$ is provided to the generator whose elements are sampled i.i.d. from the standard normal distribution with mean 0 and standard deviation 1, $\boldsymbol{z} \sim \mathcal{N}(0, 1)$.

The StyleGAN papers [1], [2] modify this basic setup by introducing a *mapping network* $f : \mathcal{Z} \to \mathcal{W}$, which takes in a standard latent vector $\boldsymbol{z}$ and produces $\boldsymbol{w} \in \mathcal{W}$. Intuitively, this mapping network allows $\boldsymbol{w}$ to be "disentangled" from the characteristics of the training distribution, allowing for easier generation [1].

StyleGAN We first directly train StyleGAN on our dataset, specifically StyleGAN2 with adaptive discriminator augmentation (ADA), an adaptation that allows for better training on low volume datasets by automatically augmenting the dataset with simple geometric transformations like rotations, translations, and color changes [3].

Future Video Prediction With an unconditional model trained, we condition the model for future video prediction similar to [4] by inserting image data into the latent vector. Unlike [4], we fully *replace* the latent with our image data. The dimensionality of the latent $\boldsymbol{z}$ is 512 while we need to fit two 256x256 images, so we greyscale both images and then apply a 16x16 average pool to reduce each image to $16^2 = 256$ parameters, then concatenate the two flattened images to form the vector $\boldsymbol{x}$. Finally, we transform $\boldsymbol{x}$ into $\boldsymbol{z} = 2\boldsymbol{x} - 1$ in order to center the image $\boldsymbol{x}$ (with pixel values between 0 and 1) into a range of -1 to 1, centered at 0. This latent is then passed to the generator, which generates videos autoregressively according to the following recurrence:

$$F_{n+1} = G(F_n; M(n))$$

where F_n is the nth frame in the sequence and $M(n)$ is the page function giving the manga page associated with the nth frame. F_0 is the initial conditioning frame. We then optimize G according to the objective of mean squared loss against the known ground truth video frames and with gradient decent as usual.

Latent Exploration Lastly, we explore the latent space of the learned model. We first try to reconstruct an image by finding its corresponding latent, i.e. given an image X find a latent vector $\boldsymbol{w}$ such that $G(\boldsymbol{w})$ is as close to X as possible. This is achieved with the projection technique described in [2], which attempts to minimize the perceptual distance as measured by VGG16 embeddings [2], [26]. Next, given two latents $\boldsymbol{w}_0$ and $\boldsymbol{w}_1$, we can linearly interpolate between the two with $\boldsymbol{w}(t) = \boldsymbol{w}_0 + t(\boldsymbol{w}_1 - \boldsymbol{w}_0)$, $t \in [0, 1]$. Both techniques are done in $\mathcal{W}$ rather than $\mathcal{Z}$ for the decoupling described above.

5 Experiments

As mentioned in section 3, we primarily work on the anime *Girls' Last Tour* [13]. We removed the intros and outros near the beginning and end of each episode, each is around 1 minute and 30 seconds long for a total of 3 minutes removed per episode, then scaled each frame to 256x256. We then extracted every other frame, or 12 frames per second.

5.1 Training StyleGAN

We trained the unconditional StyleGAN2-ADA [3] on the resulting image dataset of 174,792 anime frames, training details are discussed in the appendix, subsection 7.1.

(a) Generated images. (b) Real images.

Figure 4: Images generated by StyleGAN2-ADA [3] compared to ground truth images from *Girls' Last Tour* [13]. Although certain images are strikingly similar (top left of both, middle generated image against the bottom left real image), the generated images show clear distortions and are discernible from the real images.

The final model achieves a Fréchet inception distance (FID) of 25.722. FID is a common metric of generated image quality; we measure FID on 50k generated images against all training images like [3]. This is relatively high compared to the FID of 3.88 achieved on the 70k image FFHQ dataset (human faces) by [3] (lower is better). They use x-flips to amplify FFHQ to 140k images, but we also apply x-flips to our 170k image dataset so the difference is not of data volume. We suspect that while the images in FFHQ and similar datasets can be thought of as independently and identically distributed (i.i.d.) since they are sampled from the internet, our images are taken from video frames and are therefore highly correlated with each other. Thus, although we have 170k images, the diversity of our images is much lower compared to FFHQ which may explain the poor qualitative results and poor quantitative FID.

5.2 Conditioning StyleGAN

We encode an initial frame and its manga page into the latent vector. We then train our unconditional model to predict successive video frames with gradient decent given the ground truth video; training details are described in the appendix, subsection 7.2.

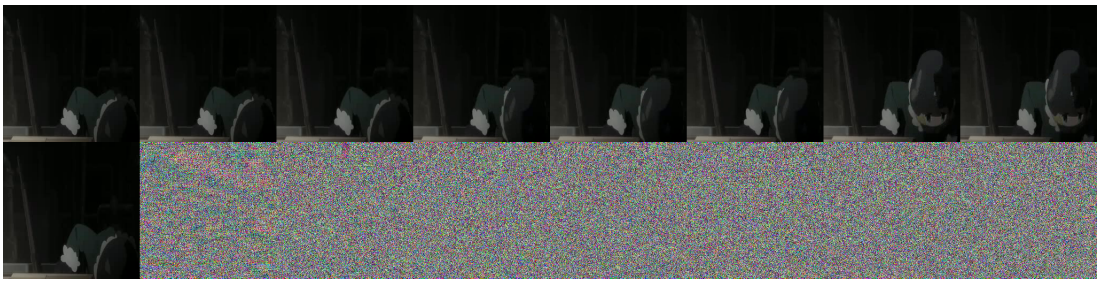


Figure 5: StyleGAN2-ADA [3] applied to future video prediction with conditioning. Top row is the ground truth 8 frames and bottom row is generated output. We notice that the output is degenerate noise, which we suspect indicates a capacity issue.

We find that the generator outputs noise no matter the training duration. We suspect that this indicates a capacity issue, as $\boldsymbol{z}$'s dimensionality of 512 is too small to effectively encode the initial frame and manga page. As mentioned in [5], there is trade-off between unconditional video synthesis and future video prediction: while future video prediction provides the ground truth objects and backgrounds which the generator is able to copy in future frames, helping FID, at the same time the generator is not able to "choose" what to generate and must continue what is given. Since $\boldsymbol{z}$ is not large enough, the generator loses the only benefit of conditioning and is therefore likely dominated by the discriminator, which furthermore benefits from conditioning as it has more information.

5.3 Latent Interpolation

We first attempt to reconstruct the ground truth video by finding the corresponding latent vector for each frame. This is shown in the middle row of Figure 6. We can see that certain frames are reconstructed well (for example, frames 3, 4, and 7) but other are not represented at all. We suspect only images frequently occurring can be convincingly represented by a corresponding latent vector. For linear interpolation, we start from the frame two seconds into the video and end one second from the end in order to prevent a completely black generated image. We then linearly interpolate from the start latent to the end. We find that the resulting video is perceptually "smooth"; each frame transitions into the next without jarring changes. This is to be expected, since the perceptual path length metric discussed in [1], [2] incentives smooth transitions. In particular, it directly minimizes the perceptual difference between adjacent interpolations in the latent space as measured with VGG16 embeddings [1], [26]. The path length regularization introduced in [2] also contributes. Because the shortest path between two points is a line, a model must map such a path into a shortest path on the image manifold, or a geodesic [2].

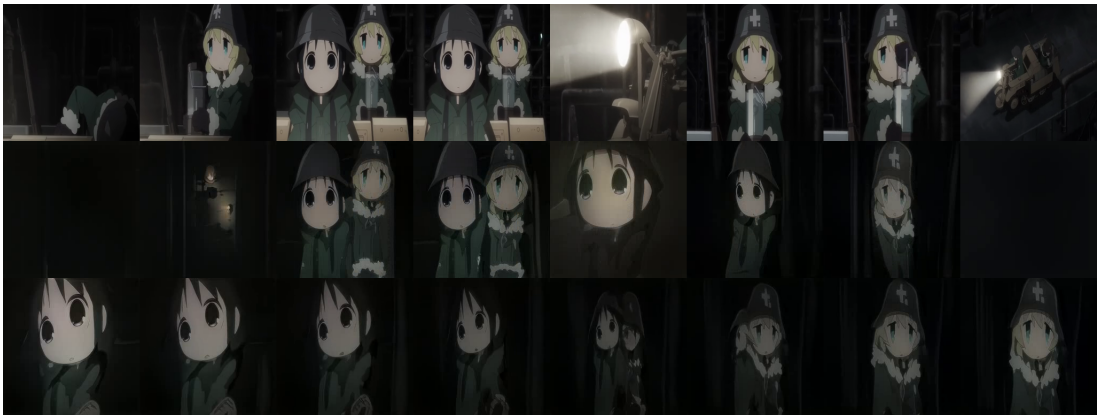


Figure 6: StyleGAN2-ADA [3] used for latent interpolation. Top row is the ground truth 8 frames sampled evenly over a video with duration ≈ 15 seconds, middle row is a reconstruction of each frame by finding its corresponding latent vector, and the bottom row is a linear interpolation from the start vector to the end vector. For the resulting videos and more examples, see https://youtu.be/4J9BBHX-uNg.

6 Conclusion

Dismayed at the state of video generation, we propose the simplified domain of generating anime. We discuss important features of anime and manga for efficient generation of a large-scale dataset from unlabeled video on the internet. We then experiment with adjusting StyleGAN [3] for future video prediction with conditioning and with linear interpolation in the latent space. Although our results are poor, we believe the avenues of research laid out in this work can be improved in future experiments. One possible approach may be to adjust DVD-GAN [5] to use a foreground/background mask [4] and then upscale the video with a deep super-resolution algorithm like waifu2x or dandere2x.

Acknowledgments This project was done in TJHSST's Computer Systems Lab. We thank Dr. Torbert and Dr. Gabor, our mentors for this research project.

References

[1] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *CoRR*, vol. abs/1812.04948, 2018. arXiv: 1812.04948. [Online]. Available: http://arxiv.org/abs/1812.04948.

[2] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," *CoRR*, vol. abs/1912.04958, 2019. arXiv: 1912.04958. [Online]. Available: http://arxiv.org/abs/1912.04958.

[3] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, *Training generative adversarial networks with limited data*, 2020. arXiv: 2006.06676 [cs.CV].

[4] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," *CoRR*, vol. abs/1609.02612, 2016. arXiv: 1609.02612. [Online]. Available: http://arxiv.org/abs/1609.02612.

[5] A. Clark, J. Donahue, and K. Simonyan, "Efficient video generation on complex datasets," *CoRR*, vol. abs/1907.06571, 2019. arXiv: 1907.06571. [Online]. Available: http://arxiv.org/abs/1907.06571.

[6] "Largest anime streaming service Crunchyroll surpasses one million paid subscribers." (2017), [Online]. Available: https://www.crunchyroll.com/anime-press-release/2017/02/09-1/largest-anime-streaming-service-crunchyroll-surpasses-one-million-paid-subscribers (visited on 04/23/2021).

[7] "Frame rate." (2021), [Online]. Available: https://en.wikipedia.org/wiki/Frame_rate (visited on 04/23/2021).

[8] "Background art design company." (2020), [Online]. Available: https://www.kusanagi.co.jp/ (visited on 04/23/2021).

[9] E. Margolis. "The dark side of Japan's anime industry." (2019), [Online]. Available: https://www.vox.com/culture/2019/7/2/20677237/anime-industry-japan-artists-pay-labor-abuse-neon-genesis-evangelion-netflix (visited on 04/23/2021).

[10] G. Boucher. "Sony gets inventive, seeks patents for 'Spider-Man: Into The Spider-Verse' animation tech." (2018), [Online]. Available: https://deadline.com/2018/12/sony-gets-inventive-seeks-patents-for-spider-man-into-the-spider-verse-animation-tech-1202518373/ (visited on 04/23/2021).

[11] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," *arXiv preprint arXiv:2011.08036*, 2020.

[12] N. Yamada, *K-On!* 2009. [Online]. Available: https://myanimelist.net/anime/5680/K-On (visited on 04/23/2021).

[13] T. Ozaki, *Girls' Last Tour*, 2017. [Online]. Available: https://myanimelist.net/anime/35838/Shoujo_Shuumatsu_Ryokou (visited on 04/23/2021).

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[15] D. Weissenborn, O. Täckström, and J. Uszkoreit, "Scaling autoregressive video models," *CoRR*, vol. abs/1906.02634, 2019. arXiv: 1906.02634. [Online]. Available: http://arxiv.org/abs/1906.02634.

[16] A. Jahanian, L. Chai, and P. Isola, "On the "steerability" of generative adversarial networks," *CoRR*, vol. abs/1907.07171, 2019. arXiv: 1907.07171. [Online]. Available: http://arxiv.org/abs/1907.07171.

[17] Tsukumizu, *Girls' Last Tour, Vol. 1*. New York: Yen Press, 2017, ISBN: 9780316470636.

[18] ——, *Girls' Last Tour, Vol. 2*. New York: Yen Press, 2017, ISBN: 9780316470643.

[19] ——, *Girls' Last Tour, Vol. 3*. New York: Yen Press, 2017, ISBN: 9780316470681.

[20] ——, *Girls' Last Tour, Vol. 4*. New York: Yen Press, 2018, ISBN: 9780316415989.

[21] N. Yamada, *K-on!!* 2010. [Online]. Available: https://myanimelist.net/anime/7791/K-On (visited on 04/23/2021).

[22] kakifly, *K-ON!, Vol. 1*. Yen Press, 2014, ISBN: 9780316409780.

[23] ——, *K-ON!, Vol. 2*. Yen Press, 2014, ISBN: 9780316409797.

[24] ——, *K-ON!, Vol. 3*. Yen Press, 2014, ISBN: 9780316409803.

[25] ——, *K-ON!, Vol. 4*. Yen Press, 2014, ISBN: 9780316409827.

[26] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *CoRR*, vol. abs/1801.03924, 2018. arXiv: 1801.03924. [Online]. Available: http://arxiv.org/abs/1801.03924.

[27] A. Mahendran, J. Thewlis, and A. Vedaldi, "Cross pixel optical flow similarity for self-supervised learning," *CoRR*, vol. abs/1807.05636, 2018. arXiv: 1807.05636. [Online]. Available: http://arxiv.org/abs/1807.05636.

[28] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

7 Appendix

7.1 StyleGAN Training

[2] mentions that the R_1 parameter γ significantly depends on the dataset. We experiment with a wide variety of choices for γ as shown in Figure 7.

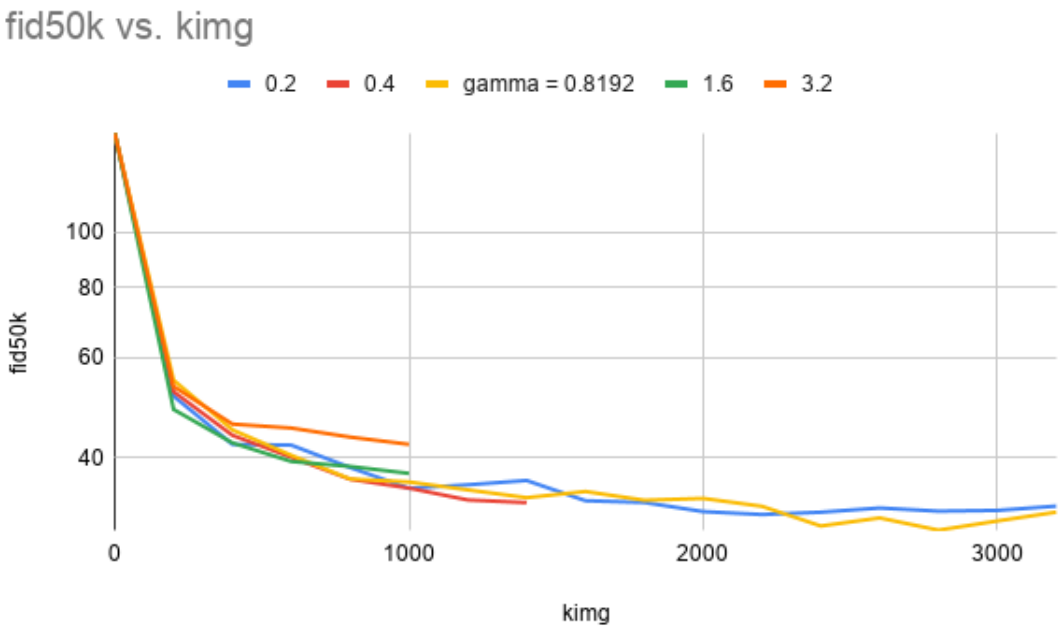


Figure 7: FID over a wide variety of γ choices.

Our final model uses the automatically determined γ of 0.8192 from the resolution of the dataset [3], x-flips, which are shown to improve FID in certain cases [3], and transfer learning from a pre-trained model on the FFHQ 256x256 dataset since transfer learning success seems to depend more on the diversity of the source dataset rather than the similarity between the two datasets [3]. After 6200 kimg (thousands of image shown to the discriminator) the model achieves a FID50k of 25.722, at which point the FID starts to rise so training was terminated.

7.2 Conditioning StyleGAN

We use mean squared loss along with the Adam optimizer with a learning rate of $\alpha = 0.001$. We use a batch size of 16, and ran the model for 4000 batches before stopping.