GANterpretations

Pablo Samuel Castro
Google Research, Brain Team
psc@google.com

Since the introduction of Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] there has been a regular stream of both technical advances (e.g., Arjovsky et al. [2017]) and creative uses of these generative models (e.g., [Karras et al., 2019, Zhu et al., 2017, Jin et al., 2017]). In this work we propose an approach for using the power of GANs to automatically generate videos to accompany audio recordings by aligning to spectral properties of the recording. This allows musicians to explore new forms of multi-modal creative expression, where musical performance can induce an AI-generated musical video that is guided by said performance, as well as a medium for creating a visual narrative to follow a storyline (similar to what was proposed by Frosst and Kereliuk [2019]).

The ability of GANs to generate a seemingly infinite amount of images is due largely to the fact that training them involves learning a *latent space* $Z \in \mathbb{R}^d$, where each point $z \in Z$ generates a unique image $G(z)$, where G is the *generator* of the GAN. When trained properly, these latent spaces are learned in a structured manner, where nearby points generate similar images. This enables smooth interpolations between two distinct images (points) $z_1, z_2 \in Z$ via their linear combination $G((1 - \alpha)z_1 + \alpha z_2)$ for some $\alpha \in [0, 1]$. For our work we make use of the BigGAN family of models [Brock et al., 2019], which are class-conditional generative models. In high-level terms, this means that the latent space is of the form $Z \times C$, where $Z \in \mathbb{R}^d$ and C is a finite set of possible categories (such as *jellyfish*, *bike*, and *boa*). These models thus allow us to perform smooth interpolations between images in the same category $c \in C$ ($G((1-\alpha)(z_1, c)+\alpha(z_2, c))$), and images from different categories $c_1, c_2 \in C$ ($G((1 - \alpha)(z_1, c_1) + \alpha(z_2, c_2))$). Given an audio recording a, we compute the spectrogram (the spectrum of different audio frequencies) which produces a 2-dimensional array $S(a)$ of size $F \times T$, where F is the number of frequencies per time sample, and T is the number of time segments the audio file has been decomposed into.

Our procedure is summarized in Algorithm 1. In words, it proceeds as follows: **(1)** Compute the Total Variation (TV) distance between the frequency spectrograms of consecutive time slices (line 2). **(2)** Slide through these TV distances and find the inflection points by comparing the current point t with the average TV difference of a window of slice L before and after t (lines 5-6); if the TV distance at point t is at a *peak*, then record this as an $inflectionPoint$ (lines 7-9); the first and last points are also added as inflection points (lines 3, 11). **(3)** Compute alpha values by normalizing the cumulative sum between inflection points (lines 13-17). **(4)** For each inflection point, generate a random category or use the ones provided by the user (lines 18 - 24); concurrently, generate latent codes z for each category. **(5)** Using the generated latent points, categories, and interpolations, generate the frames for the video (lines 25-32). This process is illustrated in Figure 1, and the full code is available at https://github.com/psc-g/ganterpretation.

We present two demonstrations of this method which allow readers to understand its functionality. In the first, we explore creating a video accompaniement for a musical performance, where the categories were selected randomly (https://youtu.be/oQI8zG0WNuI); In the second we demonstrate the ability for visual narrative, where the categories were pre-selected to match the storyline (https://youtu.be/YelauzLHI6E).

The method of GANterpretatinos can be used in different ways, two of which are shown in the videos above. Other variations might include varying the method of converting the 2D spectrogram to a 1D signal (instead of via TV distances), using different algorithms for selecting inflection points can be used, and using different GAN models. Additionally, the way the audio is generated/recorded

can affect the generated video; indeed, the audio storytelling example used explicit silences to force inflection points at desired times. An avenue we are exploring is pre-generating images in order to have real-time response so as to be used during musical performance; in this way, we will have a closed creative loop: the musical performance affects the video generation, which in turn affects the musical performance.

---

noend 1 GANterpretations

1: **Given**: Audio a, rolling length L, inflection threshold δ, pre-selected categories κ (optional)
2: $TV \leftarrow [mean(|S(a)[:, i] - S[:, i+1]|) \quad for \quad i \in [0, T-1]]$
3: $inflectionPoints \leftarrow [0]$
4: **for** $t = L+1, \ldots, T-L$ **do**
5: $\quad prevMean \leftarrow mean(TV[t-1-L : t-1]$ (rolling mean before position t)
6: $\quad nextMean \leftarrow mean(TV[t+1 : t+1+L]$ (rolling mean after position t)
7: $\quad$ **if** $sign(TV[t] - prevMean) == sign(TV[t] - nextMean)$ and
$\qquad |TV[t] - prevMean| > \delta$ and $|TV[t] - nextMean| > \delta$ **then**
8: $\qquad$ Add t to $inflectionPoints$
9: Add $T-1$ to $inflectionPoints$
10: $alphas \leftarrow [0.0]$
11: **for** $i = 1, \ldots, len(inflectionPoints)$ **do**
12: $\quad interpolation \leftarrow cumsum(TV[inflectionPoints[i-1], TV[inflectionPoints[i]])$
13: $\quad interpolation/ = (inflectionPoints[i] - inflectionPoints[i-1])$
14: $\quad$ Append $interpolation$ to $alphas$.
15: $zs \leftarrow []$
16: **for** $i = 0, \ldots, len(inflectionPoints)$ **do**
17: $\quad$ Add a random sample from Z to zs
18: $\quad$ **if** $\kappa[i]$ does not exist **then**
19: $\qquad$ Add a random category from C to κ
20: $vidFrames \leftarrow []$
21: **for** $i = 1, \ldots, len(inflectionPoints)$ **do**
22: $\quad$ **for** $t = inflectionPoints[i-1], \ldots, inflectionPoints[i]$ **do**
23: $\qquad \alpha \leftarrow alphas[t]$
24: $\qquad$ Add $G((1-\alpha)(z[i-1], \kappa[i-1]) + \alpha(z[i], \kappa[i]))$ to $vidFrames$
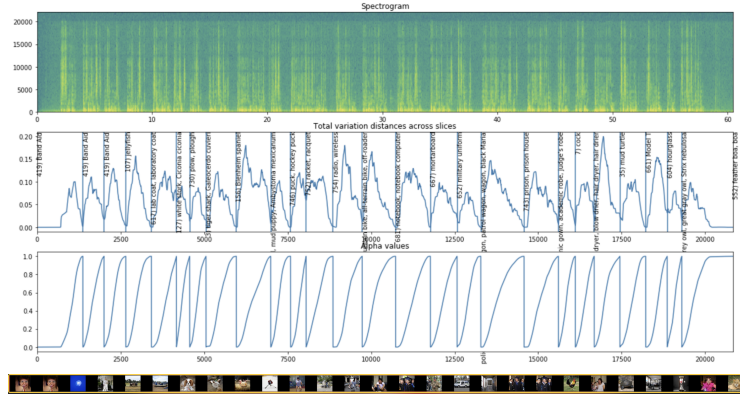25: Return $vidFrames$

Figure 1: From top to bottom: Audio spectrogram; TV distances with computed inflection points and selected categories; resulting α values used for interpolation; Final video clips

Ethical implications

This work is meant to be for purely artistic use, so the same ethical considerations that apply for all forms of art apply equally to this method. Additionally, as GANs can sometimes overfit to its training data, care must be used if displaying the generated images publicly. For our work we have restricted ourselves to BigGAN, which is trained on the well-known and public ImageNet dataset.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=B1xsqj09Fm`.

Nicholas Frosst and Jonathon Kereliuk. Text conditional lyric video generation. In *Machine Learning for Creativity and Design NeurIPS 2019 Workshop*, 2019. URL `https://neurips2019creativity.github.io/doc/Text%20Conditional%20Lyric% 20Video%20Generation.pdf`.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014. URL `http://papers. nips.cc/paper/5423-generative-adversarial-nets.pdf`.

Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, and Zhihao Fang. Towards the automatic anime characters creation with generative adversarial networks, 2017.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.