TGAN

Overall Summary
Lei Zhang
CS 297

Two Generators

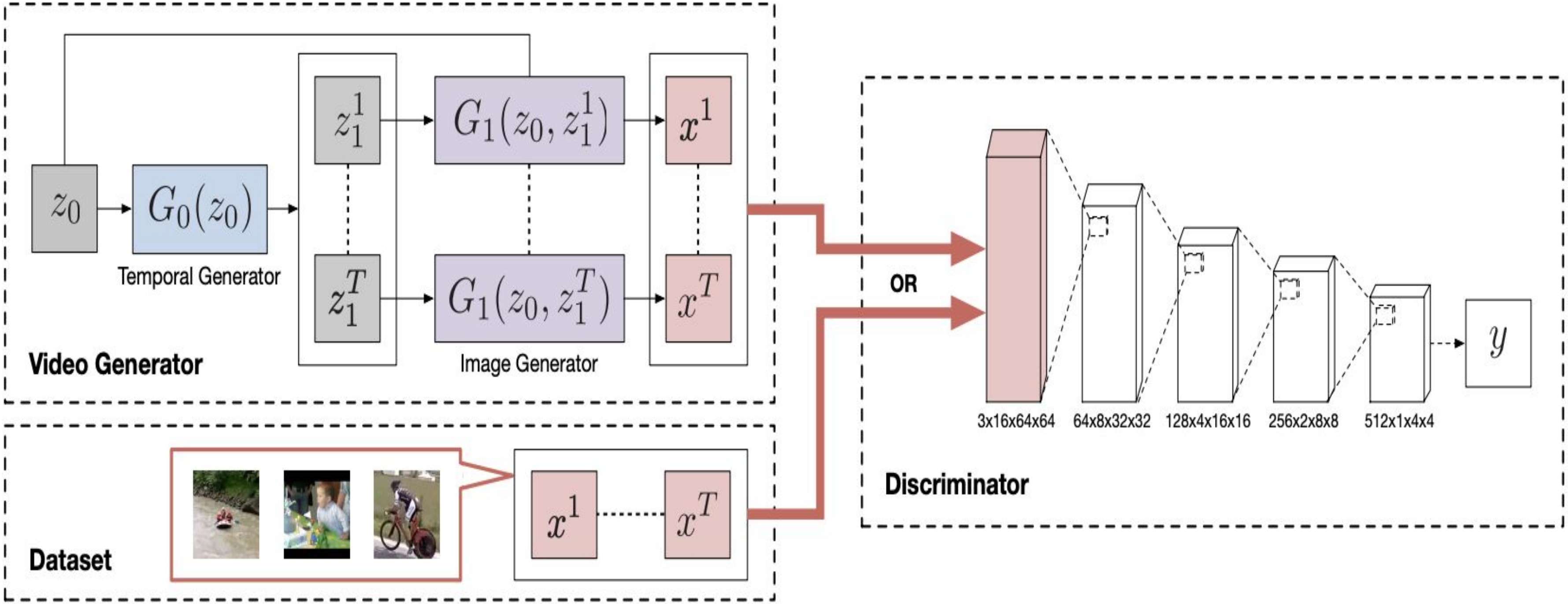
- A temporal generator
  - Input: a single noise vector
  - Output: a set of noise vectors
- An image generator
  - A set of noise vectors
  - A sequence of generated images

Argues of Current Methods

- VideoGAN - cannot generate a scene with dynamic background
- Single 3D CNN layer generator - equally changes for x-t and y-t

Although a simple approach is to use 3D convolutional layers for representing the generating process of a video, it implies that images along x-t plane and y-t plane besides x-y plane are considered equally, where x and y denote the spatial dimensions and t denotes the time dimension. We believe that the nature of time dimension is essentially different from the spatial dimensions in the case of videos so that such approach has difficulty on the video generation problem. The relevance of this assumption has been also discussed in some recent studies [33, 24, 46] that have shown good performance on the video recognition task.

TGAN Architecture



Use Wasserstein GAN

- To improve the GAN stability
- Minimize an Earth Mover's distance (EMD, a.k.a. First Wasserstein distance)

Generator Configuration

Temporal generator	Image generator	
$z_0 \in \mathbb{R}^{1\times100}$	$z_0 \in \mathbb{R}^{1\times100}$	$z_1^t \in \mathbb{R}^{100}$
deconv (1, 512, 0, 1)	linear ($256 \cdot 4^2$)	linear ($256 \cdot 4^2$)
deconv (4, 256, 1, 2)	concat + deconv (4, 256, 1, 2)	
deconv (4, 128, 1, 2)	deconv (4, 128, 1, 2)	
deconv (4, 128, 1, 2)	deconv (4, 64, 1, 2)	
deconv (4, 100, 1, 2)	deconv (4, 32, 1, 2)	
tanh	deconv (3, 3, 1, 1) + tanh	

Conditional TGAN

- Generator takes both label and noise vector (latent variable)
- Concatenate both vector and label for both generators

Inception scores on UCF-101 dataset

Method	Inception score
3D model (Weight clipping)	$4.32 \pm .01$
3D model (SVC)	$4.78 \pm .02$
Video GAN [44] (Normal GAN)	$8.18 \pm .05$
Video GAN (SVC)	$8.31 \pm .09$
TGAN (Normal GAN)	$9.18 \pm .11$
TGAN (Weight clipping)	$11.77 \pm .11$
TGAN (SVC)	$\mathbf{11.85 \pm .07}$
Conditional TGAN (SVC)	$\mathbf{15.83 \pm .18}$
UCF-101 dataset	$34.49 \pm .03$

Table 4. Inception scores for models of UCF-101.

Source Code

- Source
- Chainer

MoCoGAN

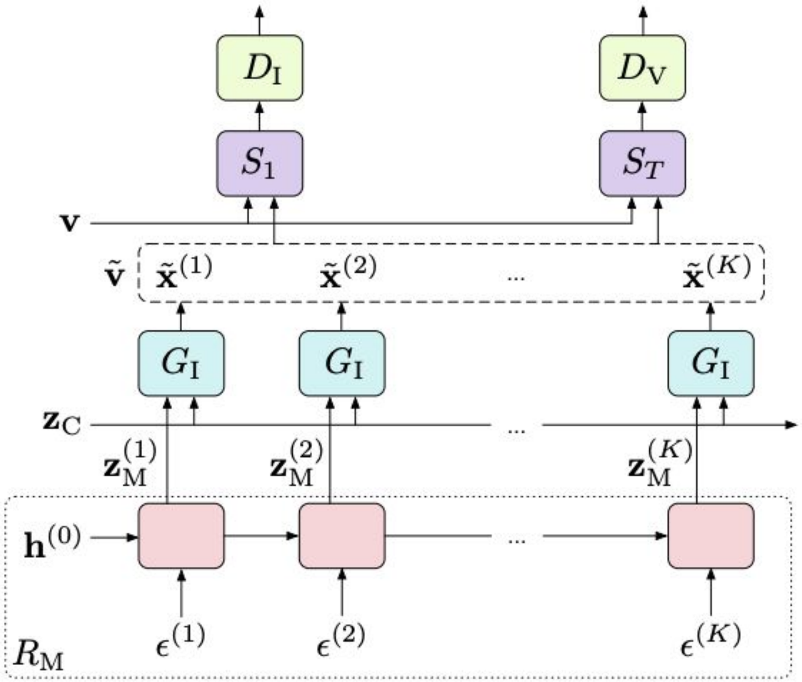


Figure 2: The MoCoGAN framework for video generation. For a video, the content vector, $\mathbf{z}_{\mathrm{C}}$, is sampled once and fixed. Then, a series of random variables $[\epsilon^{(1)}, ..., \epsilon^{(K)}]$ is sampled and mapped to a series of motion codes $[\mathbf{z}_{\mathrm{M}}^{(1)}, ..., \mathbf{z}_{\mathrm{M}}^{(K)}]$ via the recurrent neural network R_{M}. A generator G_{I} produces a frame, $\tilde{\mathbf{x}}^{(k)}$, using the content and the motion vectors $\{\mathbf{z}_{\mathrm{C}}, \mathbf{z}_{\mathrm{M}}^{(k)}\}$. The discriminators, D_{I} and D_{V}, are trained on real and fake images and videos, respectively, sampled from the training set $\mathbf{v}$ and the generated set $\tilde{\mathbf{v}}$. The function S_1 samples a single frame from a video, S_{T} samples T consequtive frames.