Adversarial Distortion for Learned Video Compression

Vijay Veerabadran¹³, Reza Pourreza¹, Amirhossein Habibian², Taco Cohen²
¹ Qualcomm AI Research, Qualcomm Technologies, Inc.
² Qualcomm AI Research, Qualcomm Technologies Netherlands B.V.
³ Department of Cognitive Science, University of California San Diego

Abstract

In this paper, we present a novel adversarial lossy video compression model. At extremely low bit-rates, standard video coding schemes suffer from unpleasant reconstruction artifacts such as blocking, ringing etc. Existing learned neural approaches to video compression have achieved reasonable success on reducing the bit-rate for efficient transmission and reduce the impact of artifacts to an extent. However, they still tend to produce blurred results under extreme compression. In this paper, we present a deep adversarial learned video compression model that minimizes an auxiliary adversarial distortion objective. We find this adversarial objective to correlate better with human perceptual quality judgement relative to traditional quality metrics such as MS-SSIM and PSNR. Our experiments using a state-of-the-art learned video compression system demonstrate a reduction of perceptual artifacts and reconstruction of detail lost especially under extremely high compression.

1. Introduction

As the resolution of digitally recorded and streamed videos keeps growing, there is an increasing demand for video compression algorithms that enable fast transmission of videos without loss in Quality-of-Experience. While current video codecs can encode video at low bitrates, this usually results in unpleasant compression artifacts [34, 24]. The application of deep neural networks to develop learned video compression algorithms as explored in recent art [28, 25, 22, 13, 9, 8, 11] produces promising results at solving this issue of perceptual artifacts. However, due to the use of distortion metrics such as MS-SSIM [31] and MSE, the reconstructions tend to be blurry [6]. Generative Adversarial Networks (GANs) have been shown to be capable of producing highly realistic images and videos from random noise inputs [18, 17, 35, 7, 10]. This suggests that the GAN objective more accurately reflects image/video quality as perceived by humans. Indeed the work of [4] has shown that

GANs can be used for low-rate high-quality image compression, by augmenting the rate/distortion loss with an adversarial loss. However, so far there is little work on the application of adversarial losses to video compression due to scaling issues.

We tackle the scaling issue via factorization of our adversarial discriminator into smaller neural network components and show results that demonstrate our compression system’s relatively improved perceptual quality even under extreme compression (see Fig. 1). Our model is based on the one proposed by [13], which is a 3D autoencoder with discrete latents and a PixelCNN++[29] prior, trained end-to-end to optimize a rate/distortion loss. Our contributions shall be applicable to other learned video compression systems in general. We also present an ablation study resulting from our search across various formulations of GANs in terms of their architecture and loss functions within the context of lossy video compression. The contributions of this paper are: i) we propose adversarial loss to improve the perceptual quality of learned video compression, ii) we study techniques to improve the training stability using adversarial loss, iii) we study a spatial-temporal factorization of discriminator to enable end-to-end training of deep video compression networks.

Figure 1: Demonstration of the effectiveness of training with adversarial loss. (a) uncompressed frame, learned compression [13] via (b) MS-SSIM distortion, (c) adversarial distortion, at similar bitrates. (See Fig. 5 caption for license information)
2. Learned Video Compression

A learned video compression typically consists of an encoder, decoder, entropy estimator and distortion estimator. All the components are trained end-to-end on a collection of videos $x$.

Encoder maps an input sequence $x$ into a latent representation $z = E(x)$. Encoder is a stack of convolutional layers with several down-samplings that reduce the input dimension. In the last layer, encoder employs a quantisation function on the activations to reduce the bit-width of latents $z = Q(z)$. Quantizer maps each element, or group of elements, in activations $\tilde{z}_i$ to a discrete symbol $z_i \in \{0, \ldots, L\}$. Learning discrete representation, as a non-differentiable function, requires adding uniform noise or soft assignment as an approximation. The decoder, which is a stack of convolutional layers with several up-samplings, reconstructs the video given discrete latents $\hat{x} = D_\psi(z)$.

Entropy estimator predicts the average number of bits needed to encode latents using a lossless entropy coding schema such as Huffman or arithmetic coding. The bit rate is measured as the cross-entropy between the true distribution of latents $p(z)$ and a density estimated by $P_\theta(z)$ as in Eq. 1. The density estimator $P_\theta(z)$ is parameterized as a neural network usually with an auto-regressive architecture i.e. PixelCNN++ [29].

$$H(z) = \mathbb{E}_{z \sim p(z)}[−\log(P_\theta(z))]$$

(1)

Distortion loss measures the difference between the input and reconstructed videos $d(x, \hat{x})$ usually by pixel-wise distances, such as $\ell_1$ and $\ell_2$, or by more sophisticated metrics such as MS-SSIM. All the aforementioned components are trained end-to-end by minimizing a rate-distortion trade-off as the loss function:

$$d(x, D_\psi(E_\phi(x))) + \beta H(z)$$

(2)

Table 1: Component functions for a few adversarial losses.

<table>
<thead>
<tr>
<th>Loss Function</th>
<th>$f$</th>
<th>$g$</th>
<th>$h$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimax [12]</td>
<td>$-\log(1 + e^{-y})$</td>
<td>$-y - \log(1 + e^{-y})$</td>
<td>$-y$</td>
</tr>
<tr>
<td>Wasserstein [5]</td>
<td>$-\log(1 + e^{-y})$</td>
<td>$-y - \log(1 + e^{-y})$</td>
<td>$-y$</td>
</tr>
<tr>
<td>Least Squares [23]</td>
<td>$(y-1)^2$</td>
<td>$-(y-1)^2$</td>
<td>$-(y-1)^2$</td>
</tr>
</tbody>
</table>

3. Adversarial Distortion Loss

The distortion, measured by pixel-wise metrics i.e. $\ell_1$, $\ell_2$, and MS-SSIM, are often not perfectly aligned with perceptual quality. Recent work [6] mathematically proves that the distortion and perceptual quality are at odds with each other and minimizing the mean distortion leads to a decrease in perceptual quality. Instead of solely relying on pixel-wise distances, we define the distortion as an adversarial loss between the decoder and a discriminator $C_\omega$. This setting can be interpreted as training a conditional GAN, where the decoder $D_\psi$ learns to generate a video given the encoded latents $z$. The discriminator encourages the decoder to generate videos which reside on the data manifold that improves perceptual quality.

3.1. Stable adversarial training

Adversarial loss Adversarial loss can be defined in various formulations depending on how to specify the component functions $f$, $g$ and $h$:

$$\max_{C_\omega} \mathbb{E}_{x}[f(C_\omega(x))] + \mathbb{E}_{\hat{x}}[g(C_\omega(\hat{x}))] \quad (3)$$

$$\min_{D_\psi} \mathbb{E}_{\hat{x}}[h(C_\omega(\hat{x}))] \quad (4)$$

Table 1 specifies the component functions for several widely used GAN formulations. We investigate the impact of each formulation on training stability as they have different loss landscapes and gradient behavior; finding the best formulation is hence non-trivial. Minimax loss [12] and Wasserstein loss [5] resulted in fairly decent improvements in terms of reconstruction quality, but we noticed the training to be unstable and time consuming. We also experimented with the Least Squares [23] and Relativistic [16] formulations. Both of these formulations resulted in stable adversarial training. Among these two choices, our best formulation was the Least Squares loss that generated higher quality videos (see section 4.2).

Perceptual loss As a way of further stabilizing our model’s adversarial training, we incorporated a semantic loss [21, 30] that minimizes the $\ell_1$ of the difference between framewise VGG-19 representations of $x$ and $\hat{x}$. This semantic loss resulted in faster and more stable training of our adversarial video compression model.

3.2. Factorized spatial-temporal discriminators

Recent work in training GANs to generate videos points out the advantage of scaling up the training, i.e., using larger batch sizes, deeper models etc. [10]. However, due to scalability issues relating to working with video data and the
models size, we faced difficulty in jointly training all components. In this case, our two choices to scale up our training were: (i) finetune the decoder using an adversarial distortion and fixing the prior and encoder, hence loading only the adversarial distortion components in memory, and (ii) factorizing our model into smaller components that enable large-scale training. While analyzing the compression performance of the above two choices, we observed that the latter produced higher quality reconstruction at the same bitrates. In order to scale up joint adversarial training for our complete model, we resorted to factorizing the discriminator into two smaller spatial and spatio-temporal discriminators. Both these discriminators were formulated as LSGAN discriminators. The average loss from these discriminators was used to train the decoder.

Putting all together, we train our model end-to-end by optimizing the rate-distortion trade-off Eq. 2 using the following distortion loss:

\[ d(x, \hat{x}) = \alpha \| x - \hat{x} \|_2^2 + \gamma \| \sigma(x) - \sigma(\hat{x}) \|_1 + \rho (C(\hat{x}) - 1)^2 \]  

(5)

where \( \sigma \) represents the VGG-19 features\(^1\). These design choices are summarized in our architecture in Fig. 2.

4. Experiments

4.1. Experimental setup

Network architecture We demonstrate the impact of our adversarial training on the Rate-Distortion Autoencoder [13] (RDAE) model which achieves state-of-the-art video compression performance compared to other learned [22, 33] compression methods in terms of MS-SSIM at various bit-rates. Further details on the architecture of RDAE’s encoder, decoder, entropy estimator and quantizer can be found in [13]. We employed a 2D ResNet-34 [15] (trained on ImageNet) and a 3D ResNet [14] (trained from scratch) as our spatial and spatio-temporal discriminators respectively. We spatially downsize our spatio-temporal discriminator’s input by half in order to save memory, however, we did not temporally subsample our spatial discriminator’s input.

Dataset We created a dataset sourced from Kinetics400 [19] by selecting the first 16 frames from each high-quality video and downsampling them to alleviate the existing compression artifacts resulting in a total of 93750 videos for training and 5687 videos for validation. We used random 160 x 160 crops for training and used full-size frames for validation. We used UVG [2] as the test dataset for comparisons with other methods.

Implementation details We pretrained our video compression model using rate-loss and MSE distortion loss to speed up adversarial training; this step provided a good initialization for adversarially training decoder weights. Our hyperparameter choices for optimizing Eq. 5 are: \( \alpha = 0.005, \gamma = 0.1, \) and \( \rho = 0.0001. \) We used a batch size of 37, for a total of 12 epochs. We trained our network with 4 values of \( \beta \{0.1, 0.3, 0.5, 0.7\} \) to obtain a rate-distortion curve. All models were trained using Adam optimizer [20] with learning rate of \( 10^{-4}, \beta_1 = 0.9 \) and \( \beta_2 = 0.999. \)

4.2. Results

Comparison to state of the art In this section, we compare the performance of our method with a learned video compression method RDAE [13] as well as two non-learned codecs H.265 [27] and H.264 [32]. Fig. 3 shows the comparison of our method with RDAE in terms of Inception Score [26] (IS, ↑) on Kinetics validation set. Incorporation of adversarial training improves IS, and hence the perceptual salience of decoded videos by a great extent.

Due to the small number of videos present in UVG, we are unable to accurately compute IS on this dataset; we report a framewise Learned Perceptual Image Patch Similarity [36] (LPIPS, ↓) score on UVG for this reason. Fig. 4 shows the comparison of our method with RDAE, H.264

\(^1\)We used features from the 4th convolution before the 5th max-pooling layer of an ImageNet-trained VGG-19 network.
and H.265 in terms of LPIPS on UVG dataset. We used ffmpeg [1] implementation of H.265 and H.264. From this analysis, we observe that adversarial training improves the perceptual quality of RDAE, resulting in a lower LPIPS score. Our model, albeit underperforms H.265, closes the gap between the learned methods and H.265 in terms of LPIPS to a high extent. In Fig. 5 we compare the visual quality of our lowest rate model with H.265 and RDAE at a similar rate. We can see that our result is relatively free from the compression artifacts usually present in H.265 and RDAE at low bitrate.

**Ablation study** We trained video compression models separately on each of the 4 GAN loss formulations mentioned in section 3 along with a stabilizing pixel-wise $\ell_1$- or $\ell_2$-norm of the distance between $x$ and $\hat{x}$ (8 GAN models in total, WGAN not reported due to unstable training). We report a summary of these experiments in Table 2 and we decided to use the best performing LSGAN with $\ell_2$ pixel-wise loss for our video compression model.

<table>
<thead>
<tr>
<th>GAN Type</th>
<th>Pixel Loss</th>
<th>MS-SSIM</th>
<th>PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCGAN [12]</td>
<td>L1</td>
<td>0.957</td>
<td>26.655</td>
</tr>
<tr>
<td>DCGAN</td>
<td>L2</td>
<td>0.958</td>
<td>26.862</td>
</tr>
<tr>
<td>RaGAN [16]</td>
<td>L1</td>
<td>0.957</td>
<td>26.554</td>
</tr>
<tr>
<td>RaGAN</td>
<td>L2</td>
<td>0.957</td>
<td>26.625</td>
</tr>
<tr>
<td>LSGAN [23]</td>
<td>L1</td>
<td>0.961</td>
<td>27.032</td>
</tr>
<tr>
<td>LSGAN</td>
<td>L2</td>
<td>0.961</td>
<td>27.032</td>
</tr>
</tbody>
</table>

Table 2: Ablation study on different GAN losses, PSNR and MS-SSIM Comparisons on Kinetics validation set.

**5. Conclusion**

In this paper, we have presented a new deep adversarial lossy video compression algorithm that outperforms state-of-the-art learned video compression systems in terms of visual quality. By employing adversarial training for the decoder, we demonstrate reduction in the perceptual artifacts (especially under very low bit-rates) typically present in the reconstructed output. We have also presented an ablation study of our design choices that resulted in our final adversarial compression model.
References


