
Video Prediction with Appearance and Motion Conditions

Yunseok Jang^{1 2} Gunhee Kim² Yale Song³

Abstract

Video prediction aims to generate realistic future frames by learning dynamic visual patterns. One fundamental challenge is to deal with future uncertainty: How should a model behave when there are multiple correct, equally probable future? We propose an Appearance-Motion Conditional GAN to address this challenge. We provide appearance and motion information as conditions that specify how the future may look like, reducing the level of uncertainty. Our model consists of a generator, two discriminators taking charge of appearance and motion pathways, and a perceptual ranking module that encourages videos of similar conditions to look similar. To train our model, we develop a novel conditioning scheme that consists of different combinations of appearance and motion conditions. We evaluate our model using facial expression and human action datasets and report favorable results compared to existing methods.

1. Introduction

Video prediction is concerned with generating high-fidelity future frames given past observations by learning dynamic visual patterns from videos. It is a promising direction for video representation learning because the model will have to learn to disentangle factors of variation based on complex visual patterns, i.e., how objects move and deform over time, how scenes change as the camera moves, how background changes as the foreground objects move, etc. While the recent advances in deep generative models (Kingma & Welling, 2013; Goodfellow et al., 2014) have brought a rapid progress to image generation (Radford et al., 2016; Isola et al., 2017; Zhu et al., 2017a), relatively little progress has been made in video prediction. We believe this is due in

This work was done at Yahoo Research during summer internship, ¹University of Michigan, Ann Arbor ²Seoul National University ³Microsoft AI & Research. Correspondence to: Yunseok Jang <yunseokj@umich.edu>, Gunhee Kim <gunhee@snu.ac.kr>, Yale Song <yalesong@microsoft.com>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

part to future uncertainty (Walker et al., 2016), making the problem somewhat ill-posed and evaluation difficult.

Previous work has addressed the uncertainty issue in several directions. One popular approach is *learning to extrapolate* multiple past frames into the future (Srivastava et al., 2015; Mathieu et al., 2016). This helps reduce uncertainty because input frames act as conditions that constrain the range of options for the future. However, when input frames are not sufficient statistics of the future, which is often the case with just a few frames (e.g., four in (Mathieu et al., 2016)), these methods suffer from blurry output caused by future uncertainty. Recent methods thus leverage *auxiliary information*, e.g., motion category labels and human pose, along with multiple input frames (Finn et al., 2016; Villegas et al., 2017b; Walker et al., 2017). Unfortunately, these methods still suffer from motionless and/or blurry output caused by the lack of clear supervision signals or suboptimal solutions found by training algorithms.

In this work, we propose an Appearance-Motion Conditional Generative Adversarial Network (AMC-GAN). Unlike most existing methods that learn from *multiple* input frames (Srivastava et al., 2015; Mathieu et al., 2016; Finn et al., 2016; Villegas et al., 2017b; Liang et al., 2017), which contain both appearance and motion information, we instead disentangle appearance from motion, and learn from a *single* input frame (appearance) and auxiliary input (motion). This allows our model to learn different factors of variation more precisely. Encoding motion with an auxiliary variable allows our model to *manipulate* how the future would look like; with a simple change of the auxiliary variable, we can make a neutral face happy or frown, or make a neutral body pose perform different gestures.

Training GANs is notoriously difficult (Salimans et al., 2016). We develop a novel conditioning scheme that constructs multiple different combinations of appearance and motion conditions – including even the ones that are *not* part of the training samples – and specify constraints to the learning objective such that videos generated under different conditions all look plausible. This makes the model generate videos under conditions *beyond what is available in the training data* and thus work much harder to satisfy the constraints during training, improving the generalization ability. In addition, we incorporate perceptual triplet ranking into

the learning objective so that videos with similar conditions look more similar to each other than the ones with different conditions. This mixed-objective learning strategy helps our model find the optimal solution effectively.

One useful byproduct of our conditional video prediction setting is that we can design an *objective* evaluation methodology that checks whether generated videos contain the likely content as specified in the input condition. This is in contrast to the traditional video prediction setting where there is no expected output, other than it being plausibly looking (Vondrick et al., 2016b). We design an evaluation technique where we train a video classifier on real data with motion category labels and test it on generated videos. We also perform qualitative analysis to assess the visual quality of the output, and report favorable results on the MUG facial expression dataset (Aifanti et al., 2010) and the NATOPS human action dataset (Song et al., 2011).

To summarize, our contributions include:

- We propose AMC-GAN that can generate multiple different videos from a single image by manipulating input conditions. The code is available at <http://vision.snu.ac.kr/projects/amc-gan>.
- We develop a novel conditioning scheme that helps the training by varying appearance and motion conditions.
- We use perceptual triplet ranking to encourage videos of similar conditions to look similar. To our best knowledge, this has not been explored in video prediction.

2. Related Work

Future Prediction: Early work proposed to use the past observation to predict certain representation of the future, e.g., object trajectory (Walker et al., 2014), optical flow (Walker et al., 2015), dense trajectory features (Walker et al., 2016), visual representation (Vondrick et al., 2016a), and human poses (Chao et al., 2017). Our work is distinct from this line of research as we aim to predict future frames rather than certain representation of the future.

Video Prediction: Ranzato et al. (2014) proposed a recurrent neural network that predicts a target frame composed of image patches (akin to words in language). Srivastava et al. (2015) used a sequence-to-sequence model to predict future frames. Early observations in video prediction have shown that predicted frames tend to be blurry (Mathieu et al., 2016; Finn et al., 2016). One primary reason for this is future uncertainty (Walker et al., 2016; Xue et al., 2016); there could be multiple correct, equally probable next frames given the previous frames. This observation has motivated two research directions: using adversarial training to make the predicted frames look realistic, and using auxiliary information as conditions to constrain what the future may look like. Our work is closely related to both directions as

we perform conditional video prediction with adversarial training. Below we review the most representative work in the two research directions.

Adversarial Training: Recent methods employ adversarial training to encourage predicted frames to look realistic and less blurry. Most work differ by the design of the discriminator: Villegas et al. (2017b) use an appearance discriminator, Mathieu et al. (2016); Villegas et al. (2017a); Vondrick et al. (2016b); Walker et al. (2017) use a motion discriminator, and Liang et al. (2017); Tulyakov et al. (2017) use both. Vondrick et al. (2016b) use a motion discriminator based on a 3D CNN; Walker et al. (2017) adopt the same motion discriminator. Our motion discriminator is similar to theirs, but differ by the use of conditioning variables. Liang et al. (2017) define two discriminators: an appearance discriminator that inspects each frame, and a motion discriminator that inspects an optical flow image predicted from each consecutive frames. Our work also employs dual discriminators, but we do not require optical flow information.

Conditional Generation: Most approaches in video prediction use multiple frames as input and predict future frames by *learning to extrapolate* (Ranzato et al., 2014; Srivastava et al., 2015; Mathieu et al., 2016; Villegas et al., 2017a; Liang et al., 2017). We consider these methods related to ours because multiple frames essentially provide appearance and motion conditions. Some of these work, similar to ours, decompose input into appearance and motion pathways and handle them separately (Villegas et al., 2017a; Liang et al., 2017). Our work is, however, distinct from all the previous methods in that we do not “learn to extrapolate”; rather, we learn to predict the future from *a single frame* so the resulting video faithfully contains motion information provided as an auxiliary variable. This latter aspect makes our work unique because, as we show later in the paper, it allows our model to *manipulate* the future depending on motion input.

For predicting future frames containing human motion, some methods estimate body pose from input frames, and decode input frames (appearance) and poses (motion) into a video (Villegas et al., 2017b; Walker et al., 2017); these methods do *video prediction by pose estimation*. Pose information is attractive because they are low-dimensional. Our work also uses a motion condition that is of low-dimensional, but is more flexible because we work with generic keypoint statistics (e.g., location and velocity); we show how we encode motion information in Section 4.

Several approaches provide auxiliary information as conditioning variables. Finn et al. (2016) use action and state information of a robotic arm. Oh et al. (2015) use Atari game actions. Reed et al. (2016) propose text-to-image synthesis; Marwah et al. (2017) propose text-to-video prediction. These methods, similar to ours, can *manipulate* how the output may look like, by changing the auxiliary

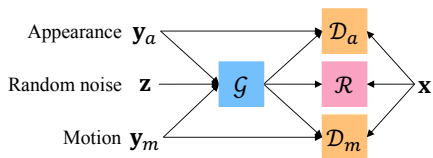


Figure 1. Our AMC-GAN consists of a generator \mathcal{G} , two discriminators each taking charge of appearance \mathcal{D}_a and motion \mathcal{D}_m pathways, and a perceptual ranking module \mathcal{R} .

information. Thus, we empirically compare our method with Finn et al. (2016), Mathieu et al. (2016) and Villegas et al. (2017a) and report improved performance.

Lastly, different from all above mentioned work, we incorporate a perceptual ranking loss (Wang et al., 2014; Gatys et al., 2015) to encourage videos that share the same appearance/motion conditions to look similar than videos that do not. Our work is, to the best of our knowledge, the first to use this constraint in the video prediction setting.

3. Approach

Our goal is to generate a video given an appearance and motion information. We formulate this as learning the conditional distribution $p(\mathbf{x}|\mathbf{y})$ where \mathbf{x} is a video and $\mathbf{y} = [\mathbf{y}_a, \mathbf{y}_m]$ is a set of conditions known to occur. We define two conditioning variables, \mathbf{y}_a and \mathbf{y}_m , that encode appearance and motion information, respectively.

We propose an Appearance-Motion Conditional GAN, shown in Figure 1. The generator \mathcal{G} seeks to produce realistic future frames. We denote a generated video by $\hat{\mathbf{x}}|\mathbf{y} = \mathcal{G}(\mathbf{z}|\mathbf{y})$, where \mathbf{z} is random noise. The two discriminator networks, on the other hand, attempt to distinguish the generated videos from the real ones: \mathcal{D}_a checks if individual frames look realistic given \mathbf{y}_a . \mathcal{D}_m checks if a video contains realistic motion given \mathbf{y}_m . Note that either discriminator alone would be insufficient to achieve our goal: without \mathcal{D}_a a generated video may have inconsistent visual appearance across frames, without \mathcal{D}_m a generated video may not depict the motion we intend to hallucinate.

The generator and the two discriminators form a conditional GAN (Mirza & Osindero, 2014). This alone would be insufficient to learn the role of conditioning variables unless a proper care is taken. If we follow the traditional training method (Mirza & Osindero, 2014), the model may treat them as random noise. To ensure that the conditioning variables have intended influence on the data generation process, we employ a ranking network \mathcal{R} , which takes as input a triplet $(\mathbf{x}|\mathbf{y}, \hat{\mathbf{x}}|\mathbf{y}, \hat{\mathbf{x}}|\mathbf{y}')$ and forces $\mathbf{x}|\mathbf{y}$ and $\hat{\mathbf{x}}|\mathbf{y}$ to look more similar to each other than $\mathbf{x}|\mathbf{y}$ and $\hat{\mathbf{x}}|\mathbf{y}'$, because in the latter pair, the conditions do not match ($\mathbf{y} \neq \mathbf{y}'$).

In addition to the ranking constraint, we propose a novel conditioning scheme to put constraints on the learning objective

with respect to the conditioning variables. We explain our learning strategy and the conditioning scheme in Section 3.3, and discuss model training in Section 3.4.

3.1. Appearance and Motion Conditions

The appearance condition \mathbf{y}_a can be any high-level abstraction that encodes visual appearance; we use a single RGB image $\mathbf{y}_a \in \mathbb{R}^{64 \times 64 \times 3}$ (e.g., the first frame of a video).

The motion condition \mathbf{y}_m can also be any high-level abstraction that encodes motion. We define it as $\mathbf{y}_m = [\mathbf{y}_m^l, \mathbf{y}_m^v]$, where $\mathbf{y}_m^l \in \mathbb{R}^c$ is a motion category label encoded as a one-hot vector, and $\mathbf{y}_m^v \in \mathbb{R}^{(T-1) \times 2k}$ is the velocity of k keypoints in 2D space detected from an image sequence of length T . We explain how we extract keypoints in Section 4. We repeat \mathbf{y}_m^l $T-1$ times to obtain $\mathbf{y}_m \in \mathbb{R}^{(T-1) \times q}$, where $q = (c + 2k)$. We set $T = 32$ in all our experiments.

We assume \mathbf{y}_m^l is known both during training and inference. However, we assume \mathbf{y}_m^v is known *only during training*; during inference, we randomly sample it from those training examples that share the same class \mathbf{y}_m^l as the test example.

3.2. The Model

We describe the four modules of our model (see Figure 2); implementation details are provided in the supplementary.

Generator: This has the encoder-decoder structure with a convLSTM (Shi et al., 2015) in the middle. It takes as input the two conditioning variables \mathbf{y}_a and \mathbf{y}_m , and a random noise vector $\mathbf{z} \in \mathbb{R}^p$ sampled from a normal distribution $\mathcal{N}(0, I)$. The output is a video $\hat{\mathbf{x}}|\mathbf{y}$ generated frame-by-frame by unrolling the convLSTM for $T-1$ times.

We use the encoder output to initialize the convLSTM. At each time step t , we provide the t -th slice of $\mathbf{y}_{m,t} \in \mathbb{R}^q$ to the convLSTM, and combine its output with the noise vector \mathbf{z} and the encoder output. This becomes input to the image decoder. The noise vector \mathbf{z} , sampled once per video, introduces a certain degree of randomness to the decoder, helping the generator probe the distribution better (Goodfellow et al., 2014). We add a skip connection to create a direct path from the encoder to the decoder. This helps the model focus on learning changes in movement rather than full appearance and motion. We empirically found this to be crucial in producing high quality output.

Appearance Discriminator: This takes as input four images, an appearance condition \mathbf{y}_a and three frames $\mathbf{x}_{t-1:t+1}$ from either a real or a generated video, and produces a scalar indicating whether the frame is real or fake. Note the conditional formulation with \mathbf{y}_a : This is crucial to ensure the appearance of generated frames is cohesive across time with the first frame, e.g., it should be facial movements that change over time, not its identity.

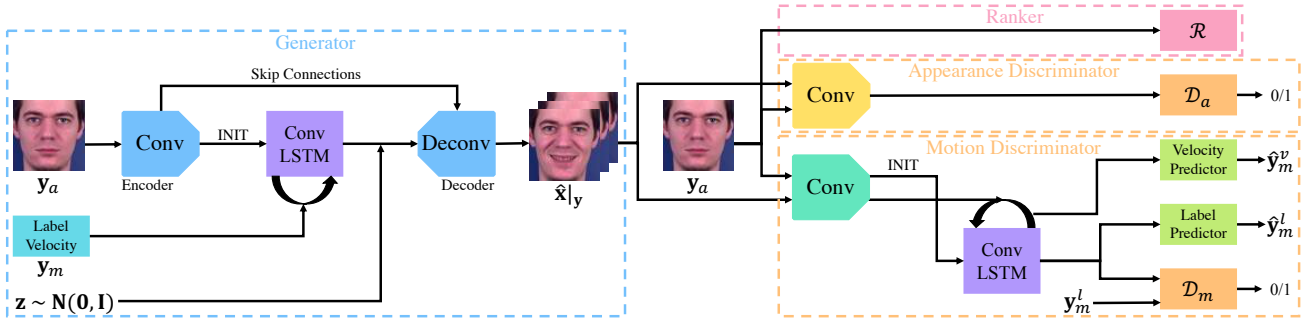


Figure 2. An overview of our AMC-GAN; we provide architecture and implementation details in the supplementary material.

Motion Discriminator: This takes as input a video $\mathbf{x} = [\mathbf{x}_{1:T}]$ and the two conditions \mathbf{y}_a and \mathbf{y}_m^l . It predicts three variables: a scalar indicating whether the video is real or fake, $\hat{y}_m^l \in \mathbb{R}^c$ representing motion categories, and $\hat{y}_m^v \in \mathbb{R}^{2k}$ representing the velocity of k keypoints. The first is the adversarial discrimination task; we provide the motion category label \mathbf{y}_m^l to perform class-conditional discrimination of the videos. The latter two are auxiliary tasks, similar to InfoGAN (Chen et al., 2016) and BicycleGAN (Zhu et al., 2017b), introduced to make our model more robust. We show the importance of the auxiliary tasks in Section 4.

Perceptual Ranking: This takes a triplet $(\mathbf{x}|_y, \hat{\mathbf{x}}|_y, \hat{\mathbf{x}}|_{y'})$ and outputs a scalar indicating the amount of violation for a constraint $d(\mathbf{x}|_y, \hat{\mathbf{x}}|_y) < d(\mathbf{x}|_y, \hat{\mathbf{x}}|_{y'})$, where $d(\cdot, \cdot)$ is a function that computes a perceptual distance between two videos, and $\mathbf{y}' \neq \mathbf{y}$; we call \mathbf{y}' a “mismatched” condition.

To compute the perceptual distance, we adapt the idea of the perceptual loss used in image style transfer (Gatys et al., 2015; Johnson et al., 2016), in which the distance is measured based on the feature representation at each layer of a pretrained CNN (e.g., VGG-16). In this work, we cannot simply use a pretrained CNN because of the conditioning variables; we instead use our own discriminator networks to compute them. Since we have two discriminators, we choose one based on a mismatched condition \mathbf{y}' , i.e., we use \mathcal{D}_a when $\mathbf{y}' = [\mathbf{y}_a', \mathbf{y}_m]$ and \mathcal{D}_m when $\mathbf{y}' = [\mathbf{y}_a, \mathbf{y}_m']$.

There are two ways to compute the perceptual distance: compare filter responses directly or the Gram matrices of the filter responses. The former encourages filter responses of a generated video to replicate, pixel-to-pixel, the ones of a training video. This is too restrictive for our purpose because we want our model to “go beyond” what exists in the training data; we want $\hat{\mathbf{x}}|_{y'}$ to look realistic even if the given (video, condition) pair does not exist in the training set. The latter relaxes this restriction by encouraging filter responses to share similar correlation patterns between two videos. We take this latter approach in our work.

Let $G_j^D(\cdot)$ be the Gram matrix computed at the j -th layer of a discriminator network \mathcal{D} . We define the distance function

at the j -th layer of the network as

$$d_j(\mathbf{x}|_y, \hat{\mathbf{x}}|_y) = \|G_j^D(\mathbf{x}|_y) - G_j^D(\hat{\mathbf{x}}|_y)\|_F \quad (1)$$

where $\|\cdot\|_F$ is the Frobenius norm. To compute the Gram matrix, we reshape the output of the j -th layer from a discriminator network to be the size of $N_j \times M_j$, where $N_j = T_j \times C_j$ (sequence length \times number of channels) and $M_j = H_j \times W_j$ (height \times width). Denoting this reshaped matrix by $\omega_j(\mathbf{x})$, the Gram matrix is

$$G_j^D(\mathbf{x}) = \omega_j(\mathbf{x})^\top \omega_j(\mathbf{x}) / N_j M_j. \quad (2)$$

Finally, we employ triplet ranking (Wang et al., 2014; Schroff et al., 2015) to measure the amount of violation, using $\mathbf{x}|_y$ as an anchor point and $\hat{\mathbf{x}}|_y$ and $\hat{\mathbf{x}}|_{y'}$ as positive and negative samples, respectively. Specifically, we use the hinge loss form to quantify the amount of violation:

$$\mathcal{R}(\mathbf{x}|_y, \hat{\mathbf{x}}|_y, \hat{\mathbf{x}}|_{y'}) = \sum_j \max(0, \rho - d_j^- + d_j^+) \quad (3)$$

where ρ determines the margin between positive and negative pairs (we set ρ as 0.01 for \mathcal{D}_a and 0.001 for \mathcal{D}_m), $d_j^+ = d_j(\mathbf{x}|_y, \hat{\mathbf{x}}|_y)$ and $d_j^- = d_j(\mathbf{x}|_y, \hat{\mathbf{x}}|_{y'})$, and $j = [1, 2]$.

3.3. Learning Strategy

We specify three constraints on the behavior of our model to help it learn the data distribution effectively:

- **C1:** If we take one of the training samples $\mathbf{x}|_y$ and pair it with a different condition, i.e., $(\mathbf{x}|_y, \mathbf{y}')$, our discriminators should be able to tell the pair is fake.
- **C2:** Regardless of the input condition, videos produced by the generator should be able to fool the discriminators into believing that $\hat{\mathbf{x}}|_y$ and $\hat{\mathbf{x}}|_{y'}$ are real.
- **C3:** The pair $(\mathbf{x}|_y, \hat{\mathbf{x}}|_y)$ should look more similar to each other than the pair $(\mathbf{x}|_y, \hat{\mathbf{x}}|_{y'})$ because the former shares the same condition (in the latter, $\mathbf{y} \neq \mathbf{y}'$).

Conditioning Scheme: We provide three conditions to the generator, listed in Table 1. The first contains the original condition $(\mathbf{y}_a, \mathbf{y}_m)$ matched with a training video $\mathbf{x}|_{\mathbf{y}_a, \mathbf{y}_m}$. The other two pairs contain mismatched information on

Appearance	Motion	Output
\mathbf{y}_a	\mathbf{y}_m	$\hat{\mathbf{x}} _{\mathbf{y}_a, \mathbf{y}_m} = \mathcal{G}(\mathbf{z} \mathbf{y}_a, \mathbf{y}_m)$
$\mathbf{y}_{a'}$	\mathbf{y}_m	$\hat{\mathbf{x}} _{\mathbf{y}_{a'}, \mathbf{y}_m} = \mathcal{G}(\mathbf{z} \mathbf{y}_{a'}, \mathbf{y}_m)$
\mathbf{y}_a	$\mathbf{y}_{m'}$	$\hat{\mathbf{x}} _{\mathbf{y}_a, \mathbf{y}_{m'}} = \mathcal{G}(\mathbf{z} \mathbf{y}_a, \mathbf{y}_{m'})$

Table 1. Three conditions used in the generator network.

\mathcal{D}_a				\mathcal{D}_m			
\mathbf{x}	\mathbf{y}	\mathcal{G}	\mathcal{D}	\mathbf{x}	\mathbf{y}	\mathcal{G}	\mathcal{D}
$\mathbf{x} _{\mathbf{y}_a, \mathbf{y}_m}$	\mathbf{y}_a	-	✓	$\mathbf{x} _{\mathbf{y}_a, \mathbf{y}_m}$	\mathbf{y}_m	-	✓
$\mathbf{x} _{\mathbf{y}_a, \mathbf{y}_{m'}}$	$\mathbf{y}_{a'}$	-	✗	$\mathbf{x} _{\mathbf{y}_a, \mathbf{y}_m}$	$\mathbf{y}_{m'}$	-	✗
$\hat{\mathbf{x}} _{\mathbf{y}_a, \mathbf{y}_m}$	\mathbf{y}_a	✓	✗	$\hat{\mathbf{x}} _{\mathbf{y}_a, \mathbf{y}_m}$	\mathbf{y}_m	✓	✗
$\hat{\mathbf{x}} _{\mathbf{y}_{a'}, \mathbf{y}_m}$	$\mathbf{y}_{a'}$	✓	✗	$\hat{\mathbf{x}} _{\mathbf{y}_a, \mathbf{y}_{m'}}$	$\mathbf{y}_{m'}$	✓	✗

Table 2. Four conditions used in each discriminator network, with labels for the generator and discriminators: real (✓) or fake (✗).

either variable. We select the mismatched condition by randomly selecting another condition from the training set. Note that we do not feed the pair $(\mathbf{y}_{a'}, \mathbf{y}_{m'})$ to the generator as it is equivalent to one of the other three combinations.

We provide four conditions to each discriminator, listed in Table 2. The first and the third rows are identical to conditional GAN (Mirza & Osindero, 2014). Training our model with just these two conditions may make our model treat the conditioning variables as random noise in the worst case. This is because there is no constraint on the expected behavior of the conditioning variables on the generation process, other than just having the end results look realistic.

We provide $(\mathbf{x}|_{\mathbf{y}}, \mathbf{y}')$ to the discriminators (the second row in Table 2) and have them identify it as fake; this enforces the constraint **C1**. Note that there is no gradient flow back to the generator because it has no control over $\mathbf{x}|_{\mathbf{y}}$. A similar idea was used by (Reed et al., 2016), where they used a mismatched sentence for the text-to-image synthesis task. We provide $(\hat{\mathbf{x}}|_{\mathbf{y}'}, \mathbf{y}')$ to the discriminators (the fourth row) to enforce the constraint **C2**. With this, the generator needs to work harder to fool the discriminators because this condition does not exist in the training set. We do not include $(\hat{\mathbf{x}}|_{\mathbf{y}}, \mathbf{y}')$ and $(\hat{\mathbf{x}}|_{\mathbf{y}'}, \mathbf{y})$ because the conditions used in generator do not match with the conditions provided to the discriminator.

3.4. Model Training

Our learning objective is to solve the min-max game:

$$\min_{\theta_{\mathcal{G}}} \max_{\theta_{\mathcal{D}}} \mathcal{L}_{gan}(\theta_{\mathcal{G}}, \theta_{\mathcal{D}}) + \mathcal{L}_{rank}(\theta_{\mathcal{G}}) + \mathcal{L}_{\mathcal{D}_{aux}}(\theta_{\mathcal{D}}) + \mathcal{L}_{\mathcal{G}_{aux}}(\theta_{\mathcal{G}}) \quad (4)$$

where each $\mathcal{L}_{\{\cdot\}}$ has its own loss weight to balance the influence of it (see supplementary). The first term follows the conditional GAN objective (Mirza & Osindero, 2014):

$$\mathcal{L}_{gan}(\theta_{\mathcal{G}}, \theta_{\mathcal{D}}) = \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [\log \mathcal{D}(\mathbf{x}|\mathbf{y})] + \mathbb{E}_{\mathbf{z} \sim p_z(\mathbf{z})} [\log(1 - \mathcal{D}(\mathcal{G}(\mathbf{z}|\mathbf{y})|\mathbf{y}))] \quad (5)$$

where we collapsed \mathcal{D}_a and \mathcal{D}_m into \mathcal{D} for brevity. We use the cross entropy loss for the real/fake discriminators. The

Algorithm 1 AMC-GAN Training Algorithm

- 1: **Input:** Dataset $\{\mathbf{x}|_{\mathbf{y}}\}$, conditions \mathbf{y} and \mathbf{y}' , step size η
- 2: **for** each step **do**
- 3: $\mathbf{z} \sim \mathcal{N}(0, I)$
- 4: $\hat{\mathbf{x}}|_{\mathbf{y}} \leftarrow \mathcal{G}(\mathbf{z}|\mathbf{y}), \hat{\mathbf{x}}|_{\mathbf{y}'} \leftarrow \mathcal{G}(\mathbf{z}|\mathbf{y}')$
- 5: $(s_r, v_r, l_r) \leftarrow \mathcal{D}(\mathbf{x}|_{\mathbf{y}}, \mathbf{y}), (s_m, v_m, l_m) \leftarrow \mathcal{D}(\mathbf{x}|_{\mathbf{y}}, \mathbf{y}')$
 $(s_f, v_f, l_f) \leftarrow \mathcal{D}(\hat{\mathbf{x}}|_{\mathbf{y}}, \mathbf{y}), (s_{f'}, v_{f'}, l_{f'}) \leftarrow \mathcal{D}(\hat{\mathbf{x}}|_{\mathbf{y}'}, \mathbf{y}')$
- 7: $\mathcal{L}_{\mathcal{D}} \leftarrow \log(s_r) + 0.5[\log(1 - s_m) + 0.5(\log(1 - s_f) + \log(1 - s_{f'}))]$
- 9: $\mathcal{L}_{\mathcal{D}_{aux}} \leftarrow \left[\|y_m^v - v_r\|_2^2 + \|y_m^v - v_f\|_2^2 + \|y_m^v - v_{f'}\|_2^2 - \sum_i y_{m,i}^l [\log y(l_{r,i}) + \log y(l_{f,i}) + \log y(l_{f',i})] \right]$
- 11: $\theta_{\mathcal{D}} \leftarrow \theta_{\mathcal{D}} + \eta \frac{\partial(\mathcal{L}_{\mathcal{D}} - \mathcal{L}_{\mathcal{D}_{aux}})}{\partial \theta_{\mathcal{D}}}$
- 12: $d_j^+ = \|G_j^{\mathcal{D}}(\mathbf{x}|_{\mathbf{y}}) - G_j^{\mathcal{D}}(\hat{\mathbf{x}}|_{\mathbf{y}})\|_F$ for $j = 1, 2$
- 13: $d_j^- = \|G_j^{\mathcal{D}}(\mathbf{x}|_{\mathbf{y}'} - G_j^{\mathcal{D}}(\hat{\mathbf{x}}|_{\mathbf{y}'})\|_F$ for $j = 1, 2$
- 14: $\mathcal{L}_{\mathcal{G}} \leftarrow \log(s_f) + \log(s_{f'})$
- 15: $\mathcal{L}_{\mathcal{G}_{aux}} \leftarrow \sum_{t=2}^T \|\mathbf{x}_t|_{\mathbf{y}} - \hat{\mathbf{x}}_t|_{\mathbf{y}}\|_1 + \sum_{j=1}^2 d_j^+$
- 16: $\mathcal{L}_{rank} \leftarrow \sum_{j=1}^2 \max(0, \rho - d_j^- + d_j^+)$
- 17: $\theta_{\mathcal{G}} \leftarrow \theta_{\mathcal{G}} + \eta \frac{\partial(\mathcal{L}_{\mathcal{G}} - \mathcal{L}_{\mathcal{G}_{aux}} - \mathcal{L}_{rank})}{\partial \theta_{\mathcal{G}}}$
- 18: **end for**

second term is our perceptual ranking loss (see Eqn. (3))

$$\mathcal{L}_{rank}(\theta_{\mathcal{G}}) = \mathcal{R}(\mathbf{x}|_{\mathbf{y}}, \hat{\mathbf{x}}|_{\mathbf{y}}, \hat{\mathbf{x}}|_{\mathbf{y}'}). \quad (6)$$

The two terms play complementary roles during training: The first encourages the solution to satisfy **C1** and **C2**, while the second encourages the solution to satisfy **C3**.

The third term is introduced to increase the power of our motion discriminator:

$$\mathcal{L}_{\mathcal{D}_{aux}} = \mathcal{L}_{CE}(\mathbf{y}_m^l, \hat{\mathbf{y}}_m^l) + \mathcal{L}_{MSE}(\mathbf{y}_m^v, \hat{\mathbf{y}}_m^v) \quad (7)$$

where the first term is the cross entropy loss for predicting motion category labels, and the second is the mean square error loss for predicting the velocity of keypoints.

The fourth term is introduced to increase the power of the generator, and is similar to the reconstruction loss widely used in video prediction (Mathieu et al., 2016),

$$\mathcal{L}_{\mathcal{G}_{aux}} = \|\mathbf{x}|_{\mathbf{y}} - \hat{\mathbf{x}}|_{\mathbf{y}}\|_1 + \sum_j d_j(\mathbf{x}|_{\mathbf{y}}, \hat{\mathbf{x}}|_{\mathbf{y}}). \quad (8)$$

Algorithm 1 summarizes how we train our model. We solve the bi-level optimization problem where we alternate between solving for $\theta_{\mathcal{D}}$ with respect to the optimum of $\theta_{\mathcal{G}}$ and vice versa. We train the discriminator networks based on a mini-batch containing a mix of the four cases listed in Table 2. We put different weights to each of the four cases (Line 7), as suggested by (Reed et al., 2016). The generator is trained on a mini-batch of the three cases listed in Table 1. We use the ADAM optimizer (Kingma & Ba, 2015) with learning rate $2e-4$. For the cross entropy losses, we adopt the label smoothing trick (Salimans et al., 2016) with a weight decay of $1e-5$ per mini-batch (Arjovsky & Bottou, 2017).

Method	MUG	NATOPS
Random	16.67	4.17
CDNA [†] (Finn et al., 2016)	35.38	6.80
Adv+GDL [‡] (Mathieu et al., 2016)	40.47	8.45
MCnet [‡] (Villegas et al., 2017a)	43.22	12.76
AMC-GAN[†] (ours)	99.79	91.12

Table 3. Video classification results (accuracy). Models with [†] learn from a single input frame, while [‡] use four input frames.

4. Experiments

We evaluate our approach on the MUG facial expression dataset (Aifanti et al., 2010) and the NATOPS human action dataset (Song et al., 2011). The MUG dataset contains 931 video clips performing six basic emotions (Ekman, 1992) (anger, disgust, fear, happy, sad, surprise). We preprocess it so that each video has 32 frames with 64×64 pixels (see supplementary for details). We use 11 facial landmark locations (2, 9, 16, 20, 25, 38, 42, 45, 47, 52, 58th) as keypoints for each frame, detected using the OpenFace toolkit (Baltrušaitis et al., 2016). The NATOPS dataset contains 9,600 video clips performing 24 action categories. We crop the video to 180×180 pixels with the chest at the center position and rescale it to 64×64 pixels. We use 9 joint locations (head, chest, naval, L/R-shoulders, L/R-elbows, L/R-wrists) as keypoints for each frame, provided by the dataset.

4.1. Quantitative Evaluation

Methodology: We design a c -way motion classifier using a 3D CNN (Tran et al., 2015) that predicts the motion label y_m^l from a video (see the supplementary for the architecture). To prevent the classifier from predicting the label simply by seeing the input frame(s), we only use the last 28 generated frames as input. We train the classifier on real training data, using roughly 10% for validation, and test it on generated videos from different methods.

We compare our method with recent approaches in video prediction: CDNA (Finn et al., 2016), Adv+GDL with ℓ_1 loss (Mathieu et al., 2016), and MCnet (Villegas et al., 2017a). For CDNA, we provide y_a as input image and y_m as the “action & state” variable. We use 10 masks suggested in their work, and disable teacher forcing for fair comparison with other methods. Following the original implementations for Adv+GDL and MCnet, we provide as input the first *four* consecutive frames, but no y_m . We also perform ablative analyses by eliminating various components of our method; we explain various settings as we discuss the results.

Results: Table 3 shows the results. We notice that the CDNA performs worse than the other methods. This is expected because it predicts future frames by combining multiple frames via masking, each generated by shifting the entire pixels of the previous frame in a certain direction. Our

No y_a	4.75	No \mathcal{D}_a	85.97
No y_m	7.62	No \mathcal{D}_m	79.23
No y, y'	5.52	No $\mathcal{L}_{\mathcal{D}_{aux}}$	81.05
No y'	86.80	No $\mathcal{L}_{\mathcal{G}_{aux}}$	88.29
Ours	91.12	No \mathcal{L}_{rank}	90.83

Table 4. Ablation study results on the NATOPS dataset (accuracy).

datasets contain complex object deformations that cannot be synthesized simply by shifting pixels. Because our network predicts pixel values directly, we achieve better results on more naturalistic videos. Both Adv+GDL and MCnet outperforms CDNA but not ours. We believe this is because both models learn to extrapolate past observations into the future. Therefore, if the input (four consecutive frames) do not provide enough motion information, as is true in our case (most videos start with “neutral” faces and body poses), extrapolation fails to predict future frames. Lastly, our model outperforms all the baselines by significant margins. It is because our model is *optimized* to generate videos that fool the motion discriminator with $\mathcal{L}_{\mathcal{D}_{aux}}$, which guides our model to preserve well the property of the motion condition y_m .

To verify whether our model successfully generates videos with the *correct* motion information provided as input, we run a similar experiment on the MUG dataset with only keypoints extracted from the generated output. For this, we use the OpenFace toolkit (Baltrušaitis et al., 2016) to extract 68 facial landmarks from the predicted output and render them with a Gaussian blur on a 2D grid to produce grayscale images. This is then fed into a c -way 2D CNN classifier (details in the supplementary). The results confirm that our method produces videos with the most accurate keypoint trajectories, with an accuracy of 70.34%, compared to CDNA (23.52%), Adv+GDL (28.81%), MCnet (35.38%).

For an ablation study, we remove input conditions and their corresponding discriminators to measure the relative importance of appearance and motion. Not surprisingly, removing either y_a or y_m significantly drops the performance. Similarly, having no y and y' (i.e., produce videos solely based on random noise z) results in poor performance. Finally, we remove the mismatched conditions y' from our conditioning scheme, i.e., we use only the first row of Table 1 and the first and third rows of Table 2; this is similar to the standard conditional GAN (Mirza & Osindero, 2014). We can see a performance drop. This is because our model ends up treating the conditioning variables alike to random noise; without contradicting conditions, the discriminators have no chance of learning to discriminate different conditions.

Removing \mathcal{D}_m shows significant drop in performance, which is expected because without motion constraints the model is incentivized to produce what appears as a static image (repeat y_a to make the appearance realistic). Removing \mathcal{D}_a also decreases the performance, but not as much as removing \mathcal{D}_m . This is however deceptive: videos look visu-

Preference	MUG	NATOPS
Prefers ours over CDNA	77.2%	97.1%
Prefers ours over Adv+GDL	80.4%	91.4%
Prefers ours over MCnet	72.4%	98.2%
Prefers ours over the ground truth	13.9%	5.3%

Table 5. Human subjective preference results.

ally implausible and appear to be adversarial examples (the faces under “No \mathcal{D}_a ” in Fig. 3). This shows the importance of enforcing constraints on visual appearance: without it, the model “over-optimizes” for the motion constraint.

Finally, we remove loss terms from our objective Eqn. (4). Removing $\mathcal{L}_{\mathcal{D}_{aux}}$ significantly deteriorates our model. This shows its effectiveness in enhancing the power of the motion discriminator; without it, similar to removing \mathcal{D}_m , the model is less constrained to predict realistic motion. Removing $\mathcal{L}_{\mathcal{G}_{aux}}$ decreases the performance moderately. Visual inspection of the generated videos revealed that this has a similar effect to removing \mathcal{D}_a or \mathbf{y}_a ; the model over-optimizes for the motion constraint. This is consistent with the literature (Mathieu et al., 2016; Shrivastava et al., 2017) that shows the effectiveness of the reconstruction constraint. Removing \mathcal{L}_{rank} hurts the performance only marginally; however, we found that the ranking loss improves visual quality and leads to faster model convergence.

4.2. Qualitative Results

Methodology: We adapt the evaluation protocol from (Vondrick et al., 2016b) and ask humans to specify their subjective preference when given a pair of videos generated using different methods under the same condition. We randomly chose 100 videos from the test split of each dataset and created 400 video pairs. We recruited 10 participants for this study; each rated all the pairs from each dataset.

Results: Table 5 shows the results when the participants are provided with motion category information along with videos. This ensures that their decision takes into account both *appearance* and *motion*; without the category label, their decision is purely based on appearance. Our participants significantly preferred ours over the baselines. Notably, for the NATOPS dataset, more than 90% of the participants voted for ours. This is because the dataset is more challenging with more categories (24 actions vs. 6 emotions); a model must generate plausibly looking videos (appearance) with distinct movements across categories (motion), which is more challenging with more categories.

To evaluate the quality of the generated videos in terms of appearance and motion separately, we designed another experiment with two tasks: We give the participants the same preference task but without motion category information. Subsequently, we ask them to identify which of the 7 facial expressions (neutral and 6 emotions) is depicted in each

generated video. These two tasks focus on appearance and motion, respectively. Our participants preferred ours over CDNA (80.8%), Adv+GDL (86.4%), MCnet (55%), and the ground truth (5%). The MCnet approach was a close match, showing that videos generated by ours and MCnet have a similar quality in terms of appearance. However, results from the second task showed that none of the three baselines successfully produced videos with distinct motion patterns: The human classification accuracy was: Ours 66%, CDNA 7%, Adv+GDL 3%, MCnet 7%, GT: 77%. This suggests that MCnet, while producing visually plausible output, fails to produce videos with intended motion.

Figure 3 shows generated videos. Our method produces noticeably sharper frames and manifests more distinct/correct motion patterns than the baselines. Most importantly, the results show that we can *manipulate* the future frames by changing the motion condition; notice how the same input frame \mathbf{y}_a turns into different videos. The results also show the importance of appearance and motion discriminators. Removing \mathcal{D}_a deteriorates the visual realism in the output: While the results still manifest the intended motion (“happy” in the first set of examples), the generated frames look visually implausible (the face identity changes over time). Removing \mathcal{D}_m produces what appears as a static video.

The CDNA produces blurry frames without clear motion, despite the fact that it receives the same \mathbf{y}_a and \mathbf{y}_m as our model. MCnet and Adv+GDL receive four-frame input frames, which provide appearance and motion information. While the results are sharper than the CDNA, we see motion patterns are not as distinct/correct as ours (they look almost stationary), due to future uncertainty caused by too little motion information exist in the input. This suggests that the “learning to extrapolate” approaches do not successfully address the ill-conditioning issue in video prediction.

The results from our quantitative and qualitative experiments highlight the advantage of our approach: Disentangling appearance from motion in the input space and learning dynamic visual representation using our method produces higher-fidelity videos than the compared methods, which suggests that our method learns video representation more precisely than the baselines.

5. Conclusion

We presented an AMC-GAN to address the future uncertainty issue in video prediction. The decomposition of appearance and motion conditions enabled us to design a novel conditioning scheme, which puts constraints on the behavior of videos generated under different conditions. We empirically demonstrated that our method produces sharp videos with the content expected by input conditions better than alternative solutions.

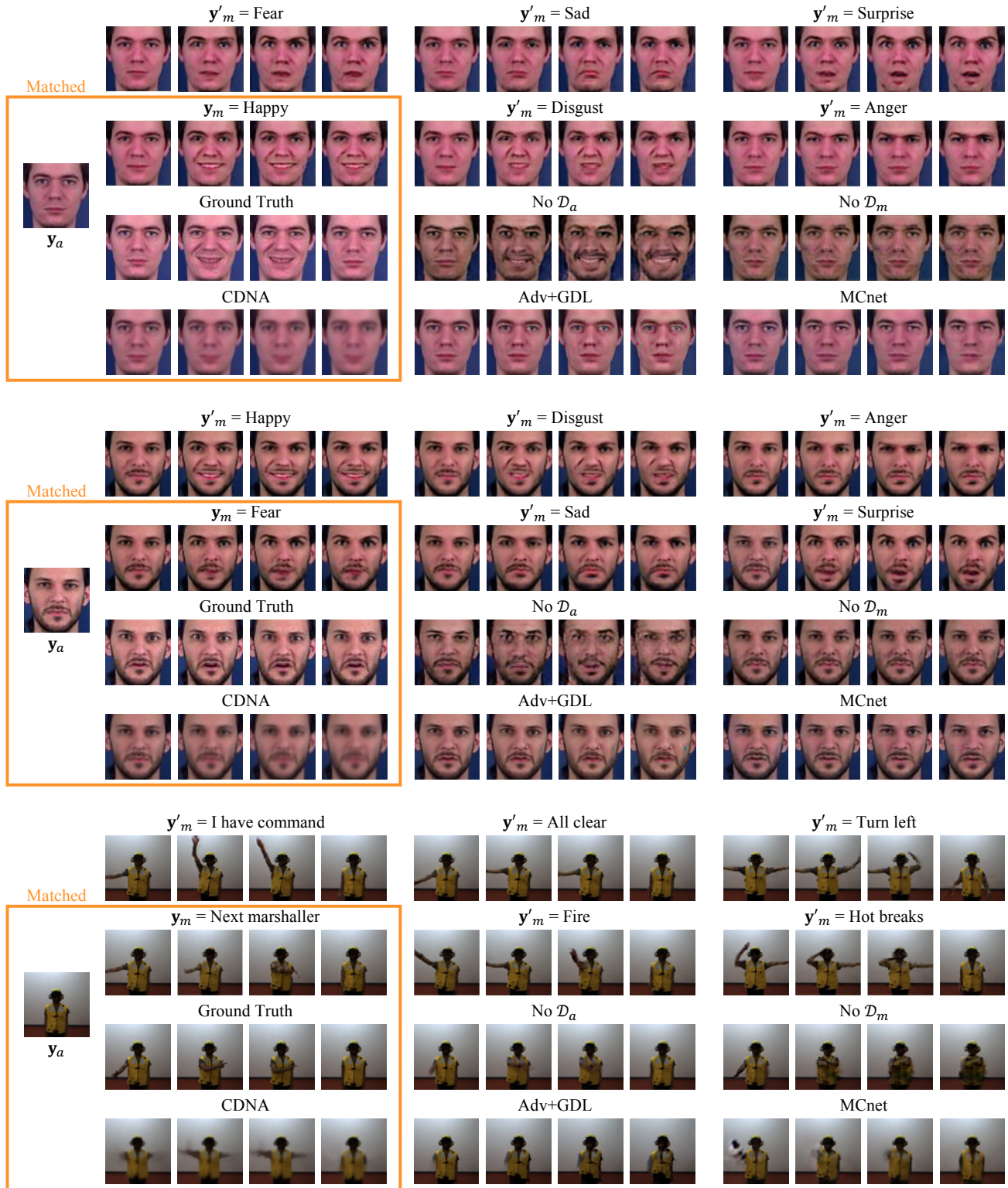


Figure 3. Video prediction results; y_a is the appearance condition (input frame), y_m is the motion condition, four frames under each method are generated results (8/16/24/32th frames). We show our approach generating different videos using matched (y_m) and mismatched (y'_m) motion conditions. We also show our ablation results (No $\mathcal{D}_a/\mathcal{D}_m$) and the baseline results. See the text for discussion. To see more results in video format, we invite the readers to visit our project page at <http://vision.snu.ac.kr/projects/amc-gan>.

Acknowledgements

We thank Kang In Kim for helpful comments about building a human evaluation page. We also appreciate Youngjin Kim, Youngjae Yu, Juyoung Kim, Insu Jeon and Jongwook Choi for helpful discussions related to the design of our model. This work is partially supported by Korea-U.K. FP Programme through the National Research Foundation of Korea (NRF-2017K1A3A1A16067245).

References

- Aifanti, N., Papachristou, C., and Delopoulos, A. The MUG Facial Expression Database. In *WIAMIS*, 2010.
- Arjovsky, M. and Bottou, L. Towards Principled Methods for Training Generative Adversarial Networks. In *ICLR*, 2017.
- Baltrušaitis, T., Robinson, P., and Morency, L.-P. OpenFace: An Open Source Facial Behavior Analysis Toolkit. In *WACV*, 2016.
- Chao, Y., Yang, J., Price, B. L., Cohen, S., and Deng, J. Forecasting Human Dynamics from Static Images. In *CVPR*, 2017.
- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., and Abbeel, P. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *NIPS*, 2016.
- Ekman, P. An Argument for Basic Emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.
- Finn, C., Goodfellow, I. J., and Levine, S. Unsupervised Learning for Physical Interaction through Video Prediction. In *NIPS*, 2016.
- Gatys, L., Ecker, A. S., and Bethge, M. Texture Synthesis Using Convolutional Neural Networks. In *NIPS*, 2015.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. In *NIPS*, 2014.
- Isola, P., Zhu, J., Zhou, T., and Efros, A. A. Image-to-Image Translation with Conditional Adversarial Networks. In *CVPR*, 2017.
- Johnson, J., Alahi, A., and Li, F. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *ECCV*, 2016.
- Kingma, D. P. and Ba, J. L. ADAM: A Method For Stochastic Optimization. In *ICLR*, 2015.
- Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114*, 2013.
- Liang, X., Lee, L., Dai, W., and Xing, E. P. Dual Motion GAN for Future-Flow Embedded Video Prediction. In *ICCV*, 2017.
- Marwah, T., Mittal, G., and Balasubramanian, V. N. Attentive Semantic Video Generation using Captions. In *ICCV*, 2017.
- Mathieu, M., Couprie, C., and LeCun, Y. Deep Multi-Scale Video Prediction Beyond Mean Square Error. In *ICLR*, 2016.
- Mirza, M. and Osindero, S. Conditional Generative Adversarial Nets. *arXiv:1411.1784*, 2014.
- Oh, J., Guo, X., Lee, H., Lewis, R. L., and Singh, S. Action-Conditional Video Prediction using Deep Networks in Atari Games. In *NIPS*, 2015.
- Radford, A., Metz, L., and Chintala, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *ICLR*, 2016.
- Ranzato, M., Szlam, A., Bruna, J., Mathieu, M., Collobert, R., and Chopra, S. Video (Language) Modeling: A Baseline for Generative Models of Natural Videos. *arXiv:1412.6604*, 2014.
- Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. Generative Adversarial Text to Image Synthesis. In *ICML*, 2016.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved Techniques for Training GANs. In *NIPS*, 2016.
- Schroff, F., Kalenichenko, D., and Philbin, J. Facenet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015.
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and chun Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *NIPS*, 2015.
- Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., and Webb, R. Learning from Simulated and Unsupervised Images through Adversarial Training. In *CVPR*, 2017.
- Song, Y., Demirdjian, D., and Davis, R. Tracking Body and Hands for Gesture Recognition: Natops Aircraft Handling Signals Database. In *FG*, 2011.
- Srivastava, N., Mansimov, E., and Salakhutdinov, R. Unsupervised Learning of Video Representations using LSTMs. In *ICML*, 2015.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. Learning Spatiotemporal Features with 3D Convolutional Networks. In *ICCV*, 2015.

- Tulyakov, S., Liu, M., Yang, X., and Kautz, J. MoCoGAN: Decomposing Motion and Content for Video Generation. *arXiv:1707.04993*, 2017.
- Villegas, R., Yang, J., Hong, S., Lin, X., and Lee, H. Decomposing Motion and Content for Natural Video Sequence Prediction. In *ICLR*, 2017a.
- Villegas, R., Yang, J., Zou, Y., Sohn, S., Lin, X., and Lee, H. Learning to Generate Long-term Future via Hierarchical Prediction. In *ICML*, 2017b.
- Vondrick, C., Pirsiavash, H., and Torralba, A. Anticipating the Future by Watching Unlabeled Video. In *CVPR*, 2016a.
- Vondrick, C., Pirsiavash, H., and Torralba, A. Generating Videos with Scene Dynamics. In *NIPS*, 2016b.
- Walker, J., Gupta, A., and Hebert, M. Patch to the Future: Unsupervised Visual Prediction. In *CVPR*, 2014.
- Walker, J., Gupta, A., and Hebert, M. Dense Optical Flow Prediction from a Static Image. In *ICCV*, 2015.
- Walker, J., Doersch, C., Gupta, A., and Hebert, M. An Uncertain Future: Forecasting from Static Images using Variational Autoencoders. In *ECCV*, 2016.
- Walker, J., Marino, K., Gupta, A., and Hebert, M. The Pose Knows: Video Forecasting by Generating Pose Futures. In *ICCV*, 2017.
- Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., and Wu, Y. Learning Fine-grained Image Similarity with Deep Ranking. In *CVPR*, 2014.
- Xue, T., Wu, J., Bouman, K. L., and Freeman, W. T. Visual Dynamics: Probabilistic Future Frame Synthesis via Cross Convolutional Networks. In *NIPS*, 2016.
- Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *ICCV*, 2017a.
- Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A. A., Wang, O., and Shechtman, E. Toward Multimodal Image-to-Image Translation. In *NIPS*, 2017b.